National
Semiconductor
Corporation

400068

Microcontroller
Databook

A Corporate Dedication to Quality and Reliability

National Semiconductor is an industry leader in the manufacture of high quality, high reliability integrated circuits. We have been the leading proponent of driving down IC defects and extending product lifetimes. From raw material through product design, manufacturing and shipping, our quality and reliability is second to none.

We are proud of our success . . . it sets a standard for others to achieve. Yet, our quest for perfection is ongoing so that you, our customer, can continue to rely on National Semiconductor Corporation to produce high quality products for your design systems.

Charles E. Sporck
President, Chief Executive Officer
National Semiconductor Corporation

Wir fühlen uns zu Qualität und Zuverlässigkeit verpflichtet

National Semiconductor Corporation ist führend bei der Herstellung von integrierten Schaltungen hoher Qualität und hoher Zuverlässigkeit. National Semiconductor war schon immer Vorreiter, wenn es galt, die Zahl von IC Ausfällen zu verringern und die Lebensdauern von Produkten zu verbessern. Vom Rohmaterial über Entwurf und Herstellung bis zur Auslieferung, die Qualität und die Zuverlässigkeit der Produkte von National Semiconductor sind unübertroffen.

Wir sind stolz auf unseren Erfolg, der Standards setzt, die für andere erstrebenswert sind. Auch ihre Ansprüche steigen ständig. Sie als unser Kunde können sich auch weiterhin auf National Semiconductor verlassen.

La Qualité et La Fiabilité:

Une Vocation Commune Chez National Semiconductor Corporation

National Semiconductor Corporation est un des leaders industriels qui fabrique des circuits intégrés d'une très grande qualité et d'une fiabilité exceptionelle. National a été le premier à vouloir faire chuter le nombre de circuits intégrés défectueux et a augmenter la durée de vie des produits. Depuis les matières premières, en passant par la conception du produit sa fabrication et son expédition, partout la qualité et la fiabilité chez National sont sans équivalents.

Nous sommes fiers de notre succès et le standard ainsi défini devrait devenir l'objectif à atteindre par les autres sociétés. Et nous continuons à vouloir faire progresser notre recherche de la perfection; il en résulte que vous, qui êtes notre client, pouvez toujours faire confiance à National Semiconductor Corporation, en produisant des systèmes d'une très grande qualité standard.

Un Impegno Societario di Qualità e Affidabilità

National Semiconductor Corporation è un'industria al vertice nella costruzione di circuiti integrati di altà qualità ed affidabilità. National è stata il principale promotore per l'abbattimento della difettosità dei circuiti integrati e per l'allungamento della vita dei prodotti. Dal materiale grezzo attraverso tutte le fasi di progettazione, costruzione e spedizione, la qualità e affidabilità National non è seconda a nessuno.

Noi siamo orgogliosi del nostro successo che fissa per gli altri un traguardo da raggiungere. Il nostro desiderio di perfezione è d'altra parte illimitato e pertanto tu, nostro cliente, puoi continuare ad affidarti a National Semiconductor Corporation per la produzione dei tuoi sistemi con elevati livelli di qualità.

Charles E. Sporck
President, Chief Executive Officer
National Semiconductor Corporation

MICROCONTROLLER
DATABOOK

TRADEMARKS

Following is the most current list of National Semiconductor Corporation's trademarks and registered trademarks.

AbuseableTM
AnadigTM
ANS-R-TRANTM
APPSTM
Auto-Chem DeflasherTM
BI-FETTM
BI-FET IITM
BI-LINETM
BIPLANTM
BLCTM
BLXTM
Brite-LiteTM
BTLTM
CIMTM
CIMBUSTM
Clock✓ChekTM
COMBOTM
COMBO ITM
COMBO IITM
COPSTM microcontrollers
DATACHECKER®
DENSPAKTM
DIBTM
Digitalker®
DISCERNTM
DISTILLTM
DNRTM
DPVMTM
ELSTARTM
E-Z-LINKTM
GENIXTM
GNXTM
HEX 3000TM
HPCTM
INFOCHEXTM
Integral ISETM
IntelisplayTM
ISETM
ISE/06TM
ISE/08TM
ISE/16TM
ISE32TM
LMCMOSTM
MacrobusTM
MacrocomponentTM
Meat✓ChekTM
MicrobusTM data bus (adjective)
MICRO-DACTM
μtalkerTM
MicrotalkerTM
MICROWIRETM
MICROWIRE/PLUSTM
MOLETM
MSTTM
Naked-8TM
National®
NAX 800TM
Nitride PlusTM
Nitride Plus OxideTM
NMLTM
NOBUSTM
NSC800TM
NSX-16TM
NS-XC-16TM
NURAMTM
OXISSTM
Perfect WatchTM
Pharma✓ChekTM
PLANTM
PolycraftTM
POSitalkerTM
QUAD3000TM
RATTM
RTX16TM
Script✓ChekTM
SCXTM
Shelf-ChekTM
SERIES/800TM
Series 32000®
SPIRETM
STARTM
StarlinkTM
STARPLEXTM
STARPLEX IITM
SuperChipTM
SYS32TM
TAPE-PAKTM
TDSTM
TeleGateTM
The National Anthem®
Time✓ChekTM
TLCTM
TrapezoidalTM
TRI-CODETM
TRI-POLYTM
TRI-SAFETM
TRI-STATE®
TURBOTRANSCEIVERTM
VIPTM
VR32TM
XMOSTM
XPUTM
Z STARTM
883B/RETSTM
883S/RETSTM

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation 2900 Semiconductor Drive, P.O. Box 58090, Santa Clara, California 95052-8090 (408) 721-5000
TWX (910) 339-9240

Microcontrollers for Today and Tomorrow

ANSWERING CUSTOMER NEEDS:

In todays applications, system designers are using more and more microcontrollers to lower system cost while providing more functions and reducing system size. By using advanced technologies and designs, National's families of microcontrollers offer a complete range of performance and features from the lowest cost to the highest performance in the industry. Now a microcontroller can be chosen that is "just right for the application."

These microcontrollers have become the standard in a multitude of products from many companies in many different industries. Designers can now choose from a wide range of devices to match the performance and cost objective of a particular product. Since all the latest products are designed to fit in National's standard cell program they can be customized for a specific application.

THE FAMILY APPROACH:

National offers families of products that satisfy a wide variety of application needs. The COPS Family of devices provide the most flexible, cost effective system solution in applications requiring timing, counting or control functions. This family offers products that sell for under 50¢ in volume and provide a very high cost/performance efficiency. The COPS Family includes over sixty different microcontrollers. They are all compatible so if you design one into a system, you can easily migrate to others in the family.

National's COPS Family is being expanded with the COP800 Family. The COP800 Family offers an excellent cost/performance efficiency. The family is designed with the core/modularity concept to make it easy to add additional family members. The chip has been designed to make it easy to add peripheral functions: such as EERAM, PLL's, A/D's, display drivers, etc.

National's 8050 Family includes the 8048, 8049, and 8050. The 8050U provides MICROWIRE PLUS for serial communication. In addition National offers the Display/Terminal Management Processor (TMP) which make designing a display controller easy.

Because future designs require more processing power, National's High Performance Controller (HPC) Family provides high performance. Utilizing microCMOS advanced low power technology, HPC is designed for code efficiency and is capable of greater than 240 ns register instruction cycle time. Designed utilizing the core/modularity concept on board peripheral functions can easily be added—A/D's, UARTS, HDLC controller, Timers/counters and gate array to name a few.

microCMOS MICROCONTROLLERS:

National's advanced microCMOS technology combines NMOS speeds and bipolar ruggedness with the inherent benefits of CMOS. This gives a definite edge in performance:

- Wide performance range
- Power saving operation
- Extended temperature range

These families of microcontrollers enables the design of reliable, high performance CMOS systems at a competitive cost.

SUPPORT FROM DEVELOPMENT TO PRODUCTION:

The MOLE (Microcontroller on Line Emulator) development system makes for easy program development and software debugging. The MOLE was developed with the microcontroller design engineer in mind. This low cost and powerful development system consists of three components—a Brain Board, a Personality Board and a personal computer.

In addition to the development systems there are Piggyback and ROMless Microcontrollers for low volume production. You can market test your product, add features and have the flexibility of tailoring the program to meet the needs of your market place.

QUALITY AND RELIABILITY:

Being second to none in quality and reliability is a corporate objective at National. It has led to reduced I.C. defects and extended product lifetimes. With a committment from Nationals top management and the implementation of a company-wide Quality Program National's quality continues towards the goal of Zero Defects.

In addition to quality, National has invested heavily in reliability. Two programs provide for product dependability: Reliability Fast Reaction Program and Long-term Reliability Audit Program.

Reliability performance at National has improved due to procedures such as:

- Designed to Withstand High Temperature Testing
- Electro Static Discharge (ESD) Voltages
- Copper Frames

National's investment in producing quality products has earned many awards from customers for outstanding quality achievements. Nationals's goal is to provide consistent levels of quality and service that are second to none.

CONTINUED INNOVATIONS:

National's heavy investment in R&D will continue to provide the most effective microcontrollers for your designs.

National is committed to stay in the forefront with innovations in cost effective microcontrollers; and by providing/developing the latest in High Peformance Controllers; and supported by a wide variety of development aids: MOLE, Piggyback devices, ROMless.

National works with you from initial development through production, for optimal cost performance solutions to your design needs. National offers Field Application Engineers along with Factory Application Engineers to provide both hardware and software design support. In addition there are workshops with hands on training, and Dial-a-helper to help with code development and debug to enable you to quickly get to market with the most competitive product available.

Table of Contents

Table of Contents (Continued)

Table of Contents (Continued)

Alpha-Numeric Index

Alpha-Numeric Index (Continued)

Alpha-Numeric Index (Continued)

Alpha-Numeric Index (Continued)

Alpha-Numeric Index (Continued)

Section 1

COP400 Family

Section 1 Contents

COPS™

The COPS family of microcontrollers provides a flexible, cost-effective system solution in applications requiring timing, counting or other control functions. COPS can be used to replace discrete logic in high-volume consumer products and low-volume industrial products allowing you to add features, miniaturize and reduce component count.

- 0.5k ROM and 32 x 4 RAM to 2k ROM and 160 x 4 RAM
- ROM-efficient instruction set
- 2–16 μs instruction cycle time plus HALT
- Common pin-out
- Wide temperature ranges: −55°C to +125°C
- MICROWIRE™ serial communication
- Low-cost MOLE™ development system

COPS wide acceptance comes from innovative products. National has built on this established family with continued and enhanced devices.

- The first under-a-dollar microcontroller led to a broader range of automotive and consumer applications.
- The first high-speed, low-power CMOS microcontrollers with 0.5k ROM provides design flexibility at low cost.
- The first microcontroller implementing MICROWIRE/PLUS™ allowing two-way communication across only three lines.
- The first under $.50 microcontroller providing excellent cost/performance benefits for applications impossible before.

National's microCMOS controllers offer a definite edge in performance. These advanced microcontrollers combine NMOS speed and bipolar ruggedness with the inherent benefits of CMOS. All the benefits of microCMOS are found in the high performance microcontroller family which now boasts more than 22 compatible device types.

As system designs become more complex, and as low power becomes more important in meeting design objectives, greater performance is required from microcontrollers. The COPS family provides the system solution.

COPS Family of Microcontrollers

Commercial Temp Version 0°C to +70°C	Industrial Temp Version −40°C to +85°C	Military Temp Version −55°C to +125°C	Technology	Description		Features									Development Tools		
				Memory		I/O											
				ROM (Bytes)	RAM (Digits)	I/O Pins	Serial I/O	Interrupt	Stack	Time Base Counter	Micro Bus	Typ. 5V Operat. Power	Max Standby at 3.3V	Size (Pins)	ROMless Device	Piggyback	Data Sheet Page
COP413L	COP313L		NMOS Low Power	0.5k	32	16	Yes	No	2 Level	No	No	15 mW	7.5 mW	20	COP401L-X13/ R13		1-67
COP414L	COP314L		NMOS Low Power	0.5k	32	16	Yes	No	2 Level	No	No	15 mW	7.5 mW	20	COP401LN		1-94
COP410L	COP310L		NMOS Low Power	0.5k	32	19	Yes	No	2 Level	No	No	15 mW	7.5 mW	20	COP401LN		1-49
COP411L	COP311L		NMOS Low Power	0.5k	32	16	Yes	No	2 Level	No	No	15 mW	7.5 mW	20	COP401LN		1-49
COP413C	COP313C		CMOS Low Power	0.5k	32	16	Yes	No	2 Level	No	No	1 mW	0.1 mW	20	COP404CN	COP444CP	1-80
COP413CH	COP313CH		CMOS Hi Speed	0.5k	32	16	Yes	No	2 Level	No	No	1 mW	0.1 mW	20	COP404CN	COP444CP	1-80
COP410C	COP310C	COP210C (Note 1)	CMOS Hi Speed	0.5k	32	19	Yes	No	2 Level	No	No	1 mW	0.1 mW	24	COP404CN	COP444CP	1-34
COP411C	COP311C	COP211C (Note 1)	CMOS Hi Speed	0.5k	32	16	Yes	No	2 Level	No	No	1 mW	0.1 mW	20	COP404CN	COP444CP	1-34
COP420	COP320		NMOS Hi Speed	1.0k	64	23	Yes	1 Source	3 Level	Yes	Yes	100 mW	N/A mW	28	COP402N	COP420P	1-111
COP421	COP321		NMOS Hi Speed	1.0k	64	19	Yes	No	3 Level	Yes	No	100 mW	N/A mW	24	COP402N	COP420P	1-111
COP422	COP322		NMOS Hi Speed	1.0k	64	16	Yes	No	3 Level	Yes	No	100 mW	N/A mW	20	COP402N	COP420P	1-111
COP424C	COP324C	COP224C (Note 2)	CMOS Hi Speed	1.0k	64	23	Yes	1 Source	3 Level	Yes	Yes	1 mW	0.1 mW	28	COP404CN	COP444CP	1-182
COP425C	COP325C	COP225C (Note 2)	CMOS Hi Speed	1.0k	64	19	Yes	No	3 Level	Yes	No	1 mW	0.1 mW	24	COP404CN	COP444CP	1-182
COP426C	COP326C	COP226C (Note 2)	CMOS Hi Speed	1.0k	64	16	Yes	No	3 Level	Yes	No	1 mW	0.1 mW	20	COP404CN	COP444CP	1-182
COP420L	COP320L		NMOS Low Power	1.0k	64	23	Yes	1 Source	3 Level	Yes	Yes	45 mW	9.9 mW	28	COP404LSN-5	COP444LP	1-133
COP421L	COP321L		NMOS Low Power	1.0k	64	19	Yes	No	3 Level	Yes	No	45 mW	9.9 mW	24	COP404LSN-5	COP444LP	1-133
COP422L	COP322L		NMOS Low Power	1.0k	64	16	Yes	No	3 Level	Yes	No	45 mW	9.9 mW	20	COP404LSN-5	COP444LP	1-133
COP420H	COP320H		NMOS Low Power	1.0k	64	23	Yes	1 Source	3 Level	Yes	Yes	40 mW	9.9 mW	28	COP404LSN-5	COP444LP	1-159
COP421H	COP321H		NMOS Low Power	1.0k	64	19	Yes	No	3 Level	Yes	No	40 mW	9.9 mW	24	COP404LSN-5	COP444LP	1-159
COP422H	COP322H		NMOS Low Power	1.0k	64	16	Yes	No	3 Level	Yes	No	30 mW	9.9 mW	20	COP404LSN-5	COP444LP	1-159
COP440	COP340		NMOS Hi Speed	2.0k	160	35	Yes	4 Sources	4 Level	Yes	Yes	205 mW	9.9 mW	40	COP404N	COP440R	1-224
COP441	COP341		NMOS Hi Speed	2.0k	160	23	Yes	4 Sources	4 Level	Yes	Yes	205 mW	9.9 mW	28	COP404N	COP440R	1-224
COP442	COP342		NMOS Hi Speed	2.0k	160	19	Yes	2 Sources	4 Level	Yes	No	205 mW	9.9 mW	24	COP404N	COP440R	1-224
COP444C	COP344C	COP244C (Note 2)	CMOS Hi Speed	2.0k	128	23	Yes	1 Source	3 Level	Yes	Yes	1 mW	0.1 mW	28	COP404CN	COP444CP	1-182
COP445C	COP345C	COP245C (Note 2)	CMOS Hi Speed	2.0k	128	19	Yes	No	3 Level	Yes	No	1 mW	0.1 mW	24	COP404CN	COP444CP	1-182
COP444L	COP344L		NMOS Low Power	2.0k	128	23	Yes	1 Source	3 Level	Yes	No	65 mW	9.9 mW	28	COP404LSN-5	COP444LP	1-247
COP445L	COP345L		NMOS Low Power	2.0k	128	19	Yes	No	3 Level	Yes	No	65 mW	9.9 mW	24	COP404LSN-5	COP444LP	1-247

Note 1: Datasheet found on page 1-6.

Note 2: Datasheet found on page 1-18.

COPS Family Development Tools

Commercial Temp Version 0°C to +70°C			Technology	Description		Features									Supplementary Description	Data Sheet Page
				Memory		I/O		Interrupt	Stack	Time Base Counter	Micro Bus	Typ. 5V Operat. Power	Max Standby at 3.3V	Size (Pins)		
				ROM (Bytes)	RAM (Digits)	I/O Pins	Serial I/O									
ROMless																
COP401L-X13			NMOS Low Power	0.5k	32	16	Yes	No	2 Level	No	No	100 mW	7.5 mW	40	Has XTAL Oscillator Option	1-284
COP401L-R13			NMOS Low Power	0.5k	32	16	Yes	No	2 Level	No	No	100 mW	7.5 mW	40	Has RC Oscillator Option	1-284
COP401L			NMOS Low Power	0.5k	32	16	Yes	No	2 Level	No	No	100 mW	7.5 mW	40	ROMless Version of COP410L	1-270
COP402			NMOS Hi Speed	1.0k	63	20	Yes	1 Source	3 Level	Yes	No	50 mW	N/A mW	40	Has Interrupt, No Microbus	1-297
COP402M			NMOS Hi Speed	1.0k	63	16	Yes	No	3 Level	Yes	Yes	125 mW	N/A mW	40	No Interrupt, Has Microbus	1-297
COP404LSN-5			NMOS Low Power	1.0k	128	20	Yes	1 Source	3 Level	Yes	No	125 mW	N/A mW	40	W/Push-Pull Mem Interface	1-339
COP404			NMOS Hi Speed	2.0k	160	23	Yes	4 Source	4 Level	Yes	Yes	35 mW	15 mW	48	ROMless Version of COP440	1-315
COP404C			CMOS Hi Speed	2.0k	128	23	Yes	1 Source	3 Level	Yes	Yes	1 mW	0.1 mW	48	CMOS ROMless Device	1-322
Piggyback																
COP420P			NMOS Hi Speed	1.0k	64	23	Yes	3 Sources	3 Level	Yes	No	50 mW	N/A mW	28	Includes: CPU, RAM, I/O	1-181
COP444LP			NMOS Low Power	2.0k	128	23	Yes	3 Sources	3 Level	Yes	No	125 mW	N/A mW	28	and EPROM Socket	1-181
COP444CP			CMOS Hi Speed	2.0k	128	23	Yes	1 Source	1 Level	Yes	Yes	1 mW	1 mW	28	Will Accept Standard EPROM	1-181

National Semiconductor Corporation

PRELIMINARY

COP210C/COP211C Single-Chip CMOS Microcontrollers

General Description

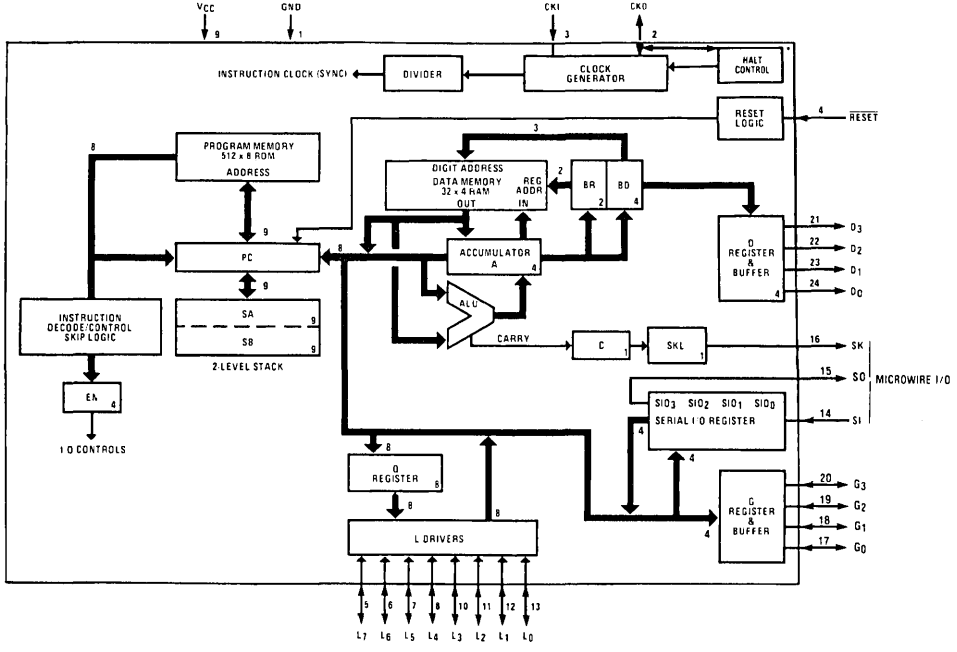
The COP210C and COP211C fully static, single-chip CMOS microcontrollers are members of the COPS™ family, fabricated using double-poly, silicon-gate CMOS technology. These controller-oriented processors are complete microcomputers containing all system timing, internal logic, ROM, RAM, and I/O necessary to implement dedicated control functions in a variety of applications. Features include single supply operation, a variety of output configuration options, with an instruction set, internal architecture, and I/O scheme designed to facilitate keyboard input, display output, and BCD data manipulation. The COP211C is identical to the COP210C but with 16 I/O lines instead of 20. They are an appropriate choice for use in numerous human interface control environments. Standard test procedures and reliable high-density fabrication techniques provide the medium to large volume customers with a customized controller-oriented processor at a low end-product cost.

The COP404C should be used for exact emulation.

Features

- Lowest power dissipation (500 μW typical)
- Low cost
- Power-saving HALT mode with Continue function
- Powerful instruction set
- 512 x 8 ROM, 32 x 4 RAM
- 20 I/O lines (COP210C)
- Two-level subroutine stack
- DC to 4.4 μs instruction time
- Single supply operation (4.5V to 5.5V)
- General purpose and TRI-STATE® outputs
- Internal binary counter register with MICROWIRE™ compatible serial I/O
- LSTTL/CMOS compatible in and out
- Software/hardware compatible with other members of the COP400 family
- Military temperature (−55°C to +125°C) devices

Block Diagram



TL/DD/8444-1

FIGURE 1. COP210C

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Maximum Allowable Voltage	V_{CC} = 6V
Voltage at Any Pin	−0.3V to V_{CC} + 0.3V
Total Allowable Source Current	25 mA
Total Allowable Sink Current	25 mA
Maximum Allowable Power Consumption	150 mW
Operating Temperature Range	−55°C to +125°C
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $-55°C \le T_A \le +125°C$ unless otherwise specified

Parameter	Conditions	Min	Max	Units
Operating Voltage		4.5	5.5	V
Supply Current (Note 1)	V_{CC} = 5.0V, t_c = Min (t_c is instruction cycle time)		4000	μA
Power Supply Ripple (Notes 3, 4)	Peak to Peak		0.25	V
HALT Mode Current (Note 2)	V_{CC} = 5.0V, F_{IN} = 0 kHz		120	μA
Input Voltage Levels				
$\overline{RESET}$, CKI				
Logic High		0.9 V_{CC}		V
Logic Low			0.1 V_{CC}	V
All Other Inputs				
Logic High		0.7 V_{CC}		V
Logic Low			0.2 V_{CC}	V
Hi-Z Input Leakage		−10	+10	μA
Input Capacitance (Note 4)			7	pF
Output Voltage Levels	Standard Outputs (except CKO)			
LSTTL Operation	V_{CC} = 5.0V ±10%			
Logic High	I_{OH} = −100 μA	2.7		V
Logic Low	I_{OL} = 400 μA		0.6	V
CMOS Operation				
Logic High	I_{OH} = −10 μA	V_{CC}−0.2		V
Logic Low	I_{OL} = 10 μA		0.2	V
Allowable Sink/Source Current per Pin (Note 5)			5	mA
CKO Current Levels (As Clock Out)				
Sink ÷4	CKI = V_{CC}, V_{OUT} = V_{CC}	0.2		mA
÷8		0.4		mA
÷16		0.8		mA
Source ÷4	CKI = 0V, V_{OUT} = 0V	−0.2		mA
÷8		−0.4		mA
÷16		−0.8		mA
Allowable Loading on CKO (as HALT I/O pin)			50	pF
Current Needed to Override HALT (Note 6)				
To Continue	V_{IN} = 0.2 V_{CC}		2.0	mA
To Halt	V_{IN} = 0.7 V_{CC}		3.0	mA
TRI-STATE or Open Drain Leakage Current		−10	+10	μA

Note 1: Supply Current is measured after running for 2000 cycle times with a square-wave clock on CKI, CKO open, and all other pins pulled up to V_{CC} with 5k resistors. See current drain equation.

Note 2: The HALT mode will stop CKI from oscillating in the RC and crystal configurations. Test conditions: all inputs tied to V_{CC}. L lines in TRI-STATE mode and tied to ground, all other outputs low and tied to ground.

Note 3: Voltage change must be less than 0.25V in a 1 ms period.

Note 4: This parameter is only sampled and not 100% tested. Variation due to the device included.

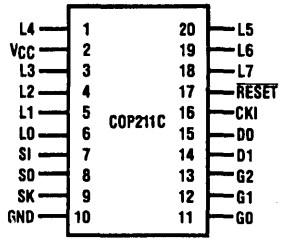
Note 5: SO Output sink current must be limited to keep V_{OL} less than 0.2 V_{CC}.

Note 6: When forcing HALT, current is only needed for a short time (approximatey 200 ns) to flip the HALT flip-flop.

AC Electrical Characteristics $-55°C \le T_A \le +125°C$ unless otherwise specified

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time (t_c)		4.4	DC	μs
Operating CKI Frequency ÷4 mode		DC	0.9	MHz
÷8 mode		DC	1.8	MHz
÷16 mode		DC	3.6	MHz
Instruction Cycle Time RC Oscillator (Note 4)	R = 30k ±5%, C = 82 pF ±5% (÷4 Mode)	6	18	μs
Inputs (See *Figure 3*)				
t_{SETUP} (Note 4) G Inputs	$V_{CC} \ge 4.5V$	tc/4 + 0.8		μs
t_{SETUP} (Note 4) SI Input	$V_{CC} \ge 4.5V$	0.33		μs
t_{SETUP} (Note 4) All Others	$V_{CC} \ge 4.5V$	1.9		μs
t_{HOLD}		0.40		μs
Output Propagation Delay t_{PD1}, t_{PD0}	V_{OUT} = 1.5V, C_L = 100 pF, R_L = 5k		1.4	μs

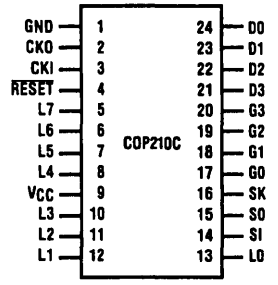
Connection Diagrams



TL/DD/8444-2

Order Number COP211C-XXX/D, See NS Package D20A

Order Number COP211C-XXX/N, See NS Package N20A



TL/DD/8444-3

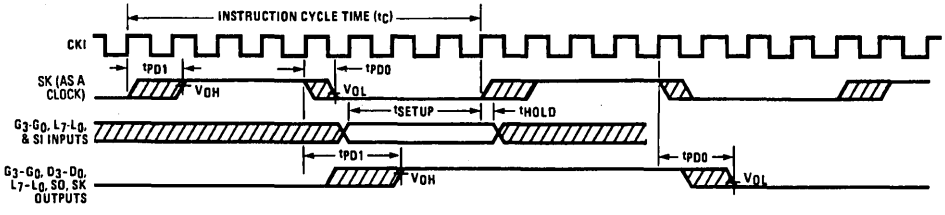
Order Number COP210C-XXX/D, See NS Package D24C

Order Number COP210C-XXX/N, See NS Package N24A

Pin Descriptions

Pin	Description	Pin	Description
L_7–L_0	8-bit bidirectional I/O port with TRI-STATE	SK	Logic-controlled clock (or general purpose output)
G_3–G_0	4-bit bidirectional I/O port (G_2–G_0 for 20-pin package)	CKI	System oscillator input
D_3–D_0	4-bit general purpose output port (D_1–D_0 for 20-pin package)	CKO	Crystal oscillator output, or HALT mode I/O port (24-pin package only)
SI	Serial input (or counter input)	RESET	System reset input
SO	Serial output (or general purpose output)	V_{CC}	System power supply
		GND	System Ground

FIGURE 2



TL/DD/8444-4

FIGURE 3. Input/Output Timing Diagrams (Divide-by-8 Mode)

Functional Description

A block diagram of the COP210C is given in *Figure 1.* Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1"; when a bit is reset, it is a logic "0".

PROGRAM MEMORY

Program memory consists of a 512-byte ROM. As can be seen by an examination of the COP210C/211C instruction set, these words may be program instructions, program data, or ROM addressing data. Because of the special characteristics associated with the JP, JSRP, JID, and LQID instructions, ROM must often be thought of as being organized into 8 pages of 64 words (bytes) each.

ROM ADDRESSING

ROM addressing is accomplished by a 9-bit PC register. Its binary value selects one of the 512 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 9-bit binary count value. Two levels of subroutine nesting are implemented by two 9-bit subroutine save registers, SA and SB.

ROM instruction words are fetched, decoded, and executed by the instruction decode, control and skip logic circuitry.

DATA MEMORY

Data Memory consists of a 128-bit RAM, organized as four data registers of 8 x 4-bit digits. RAM addressing is implemented by a 6-bit B register whose upper two bits (Br) selects one of four data registers and lower three bits of the 4-bit Bd select one of eight 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) are usually loaded into or from, or exchanged with, the A register (accumulator), they may also be loaded into the Q latches or loaded from the L ports. RAM addressing may also be performed directly by the XAD 3, 15 instruction. The Bd register also serves as a source register for 4-bit data sent directly to the D outputs.

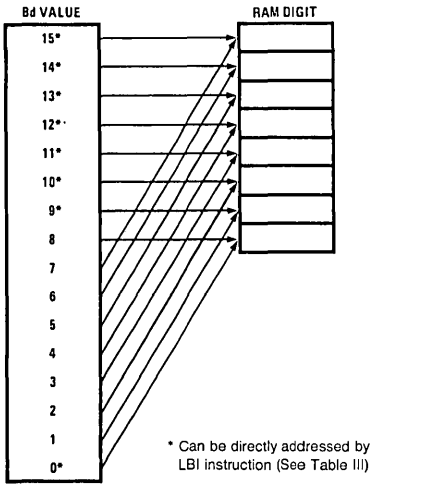
The most significant bit of Bd is not used to select a RAM digit. Hence, each physical digit of RAM may be selected by two different values of Bd as shown in *Figure 4.* The skip condition for XIS and XDS instructions will be true if Bd changes between 0 to 15, but *not* between 7 and 8 (see Table III).

INTERNAL LOGIC

The internal logic of the COP210C/211C is designed to ensure fully static operation of the device.

The 4-bit A register (accumulator) is the source and destination register for most I/O, arithmetic, logic and data memory access operations. It can also be used to load the Bd portion of the B register, to load four bits of the 8-bit Q latch data and to perform data exchanges with the SIO register.

The 4-bit adder performs the arithmetic and logic functions of the COP210C/211C, storing its results in A. It also outputs the carry information to a 1-bit carry register, most often employed to indicate arithmetic overflow. The C register, in conjunction with the XAS instruction and the EN register, also serves to control the SK output. C can be outputted directly to SK or can enable SK to be a sync clock each instruction cycle time. (See XAS instruction and EN register description below.)



TL/DD/8444–5

FIGURE 4. RAM Digit Address to Physical RAM Digit Mapping

The G register contents are outputs to four general purpose bidirectional I/O ports.

The Q register is an internal, latched, 8-bit register, used to hold data loaded from RAM and A, as well as 8-bit data from ROM. Its contents are output to the L I/O ports when the L drivers are enabled under program control. (See LEI instruction.)

The eight L drivers, when enabled, output the contents of latched Q data to the L I/O ports. Also, the contents of L may be read directly into A and RAM.

The SIO register functions as a 4-bit serial-in/serial-out shift register or as a binary counter, depending upon the contents of the EN register. (See EN register description below.) Its contents can be exchanged with A, allowing it to input or output a continuous serial data stream. With SIO functioning as a serial-in/serial-out shift register and SK as a sync clock, the COP210C/211C is MICROWIRE compatible.

The D register provides four general purpose outputs and is used as the destination register for the 4-bit contents of Bd.

The XAS instruction copies C into the SKL latch. In the counter mode, SK is the output of SKL; in the shift register mode, SK is a sync clock, inhibited when SKL is a logic "0".

The EN register is an internal 4-bit register loaded under program control by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register (EN3–EN0).

1. The least significant bit of the enable register, EN0, selects the SIO register as either a 4-bit shift register or as a 4-bit binary counter. With EN0 set, SIO is an asynchronous binary counter, *decrementing* its value by one upon each low-going pulse ("1" to "0") occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output is equal to the value of EN3. With EN0 reset, SIO is a serial shift register, shifting left each instruction cycle time. The data present at SI is shifted into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each instruction cycle time. (See 4, below.) The SK output becomes a logic-controlled clock.

Functional Description (Continued)

TABLE I. Enable Register Modes — Bits EN0 and EN3

EN0	EN3	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL = 1, SK = clock If SKL = 0, SK = 0
0	1	Shift Register	Input to Shift Register	Serial out	If SKL = 1, SK = clock If SKL = 0, SK = 0
1	0	Binary Counter	Input to Counter	0	SK = SKL
1	1	Binary Counter	Input to Counter	1	SK = SKL

2. EN1 is not used, it has *no* effect on the COP210C/211C.
3. With EN2 set, the L drivers are enabled to output the data in Q to the L I/O ports. Resetting EN2 disables the L drivers, placing the L I/O ports in a high impedance input state.
4. EN3, in conjunction with EN0, affects the SO output. With EN0 set (binary counter option selected), SO will output the value loaded into EN3. With EN0 reset (serial shift register option selected), setting EN3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN3 with the serial shift register option selected, disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains reset to "0".

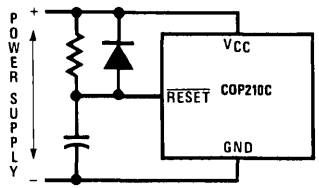
INITIALIZATION

The internal reset logic will initialize the device upon power-up if the power supply rise time is less than 1 ms and if the operating frequency at CKI is greater than 32 kHz, otherwise the external RC network shown in *Figure 5* must be connected to the $\overline{\text{RESET}}$ pin. The $\overline{\text{RESET}}$ pin is configured as a Schmitt trigger input. If not used, it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the $\overline{\text{RESET}}$ input, providing it stays low for at least three instruction cycle times.

When V_{CC} power is applied, the internal reset logic will keep the chip in initialization mode for up to 2500 instruction cycles. If the CKI clock is running at a low frequency, this could take a long time, therefore, the internal logic should be disabled by a mask option with initialization controlled solely by $\overline{\text{RESET}}$ pin.

Note: If CKI clock is less than 32 kHz, the internal reset logic (Option 25 = 1) *must* be disabled and the external RC network *must* be present.

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN, and G registers are cleared. The SK output is enabled as a SYNC output, providing a pulse each instruction cycle time. Data memory (RAM) is not cleared upon initialization. The first instruction at address 0 must be a CLRA (clear A register).



TL/DD/8444-6

RC > 5 × Power Supply Rise Time and RC > 100 × CKI Period

FIGURE 5. Power-Up Clear Circuit

COP211C

If the COP210C is bonded as a 20-pin package, it becomes the COP211C, illustrated in *Figure 2,* COP210C/211C Connection Diagrams. Note that the COP211C does not contain D2, D3, G3, or CKO. Use of this option, of course, precludes use of D2, D3, G3, and CKO options. All other options are available for the COP211C.

HALT MODE

The COP210C/211C is a *fully static* circuit; therefore, the user may stop the system oscillator at any time to halt the chip. The chip also may be halted by the HALT instruction or by forcing CKO high when it is used as a HALT I/O port. Once in the HALT mode, the internal circuitry does not receive any clock signal, and is therefore frozen in the exact state it was in when halted. All information is retained until continuing. The HALT mode is the minimum power dissipation state.

The HALT mode has slight differences depending upon the type of oscillator used.

a. 1-pin oscillator—RC or external

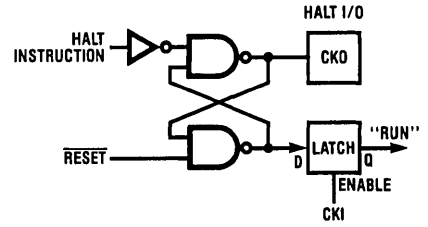
The HALT mode may be entered into by either program control (HALT instruction) or by forcing CKO to a logic "1" state.

The circuit may be awakened by one of two different methods:

1) Continue function. By forcing CKO to a logic "0", the system clock is re-enabled and the circuit continues to operate from the point where it was stopped.
2) Restart. Forcing the $\overline{\text{RESET}}$ pin to a logic "0" will restart the chip regardless of HALT or CKO (see initialization).

b. 2-pin oscillator—crystal

The HALT mode may be entered into by program control (HALT instruction) which forces CKO to a logic "1" state. The circuit can be awakened only by the $\overline{\text{RESET}}$ function.



TL/DD/8444-7

Halt I/O Port

CKO PIN OPTIONS

In a crystal-controlled oscillator system, CKO is used as an output to the crystal network. CKO will be forced high during the execution of a HALT instruction, thus inhibiting the crystal network. If a 1-pin oscillator system is chosen (RC or

Functional Description (Continued)

external), CKO will be selected as HALT and is an I/O flip-flop which is an indicator of the HALT status. An external signal can override this pin to start and stop the chip. By forcing a high level to CKO, the chip will stop as soon as CKI is high and the CKO output will go high to keep the chip stopped. By forcing a low level to CKO, the chip will continue and CKO output will go low.

All features associated with the CKO I/O pin are available with the 24-pin package only.

OSCILLATOR OPTIONS

There are three options available that define the use of CKI and CKO.

a. Crystal-Controlled Oscillator. CKI and CKO are connected to an external crystal. The instruction cycle time equals the crystal frequency divided by 16 (optionally by 8 or 4).

b. External Oscillator. CKI is configured as LSTTL-compatible input accepting an external clock signal. The external frequency is divided by 16 (optionally by 8 or 4) to give the instruction cycle time. CKO is the HALT I/O port.

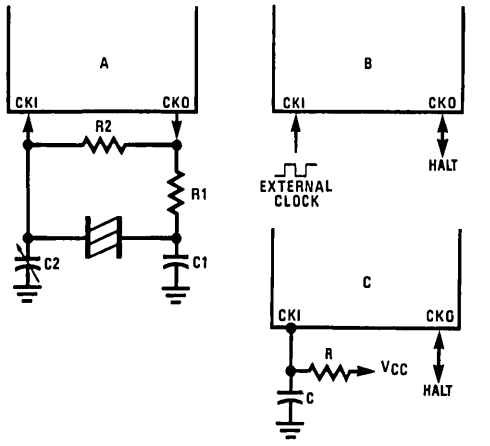
c. RC-Controlled Oscillator. CKI is configured as a single pin RC-controlled Schmitt trigger oscillator. The instruction cycle equals the oscillation frequency divided by 4. CKO is the HALT I/O port.

The RC oscillator is not recommended in systems that require accurate timing or low current. The RC oscillator draws more current than an external oscillator (typically an additional 100 μA at 5V). However, when the part halts, it stops with CKI high and the halt current is at the minimum.

COP210C/COP211C Instruction Set

Table II is a symbol table providing internal architecture, instruction operand and operational symbols used in the instruction set table.

Table III provides the mnemonic, operand, machine code, data flow, skip conditions and description associated with each instruction in the COP210C/211C instruction set.



TL/DD/8444-8

FIGURE 6. COP210C Oscillator

Crystal or Resonator					RC-Controller Oscillator		
Crystal Value	Component Values R1	R2	C1pF	C2pF	R	C	Cycle Time
32 kHz	220k	20M	30	5–36	47k	100 pF	17–25 μs
455 kHz	5k	10M	80	40	30k	82 pF	6–18 μs
3.58 MHz	1k	1M	30	6–36	Note: 15k≤R≤150k, 50 pF≤C≤150 pF		

TABLE II. COP210C/211C Instruction Set Table Symbols

Symbol	Definition
INTERNAL ARCHITECTURE SYMBOLS	
A	4-bit Accumulator
B	6-bit RAM Address Register
Br	Upper 2 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry Register
D	4-bit Data Output Port
EN	4-bit Enable Register
G	4-bit Register to latch data for G I/O Port
L	8-bit TRI-STATE I/O Port
M	4-bit contents of RAM Memory pointed to by B Register
PC	9-bit ROM Address Register (program counter)
Q	8-bit Register to latch data for L I/O Port
SA	9-bit Subroutine Save Register A
SB	9-bit Subroutine Save Register B
SIO	4-bit Shift Register and Counter
SK	Logic-Controlled Clock Output

Symbol	Definition
INSTRUCTION OPERAND SYMBOLS	
d	4-bit Operand Field, 0-15 binary (RAM Digit Select)
r	2-bit Operand Field, 0-3 binary (RAM Register Select)
a	9-bit Operand Field, 0-511 binary (ROM Address)
y	4-bit Operand Field, 0-15 binary (Immediate Data)
RAM(s)	Contents of RAM location addressed by s
ROM(t)	Contents of ROM location addressed by t
OPERATIONAL SYMBOLS	
+	Plus
−	Minus
→	Replaces
⟷	Is exchanged with
=	Is equal to
$\bar{A}$	The one's complement of A
⊕	Exclusive-OR
:	Range of values

Instruction Set (Continued)

TABLE III. COP210C/211C Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	0011 | 0000	A + C + RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	0011 | 0001	A + RAM(B) → A	None	Add RAM to A
AISC	y	5–	0101 | y	A + y → A	Carry	Add immediate, Skip on Carry (y ≠ 0)
CLRA		00	0000 | 0000	0 → A	None	Clear A
COMP		40	0100 | 0000	$\overline{A}$ → A	None	One's complement of A to A
NOP		44	0100 | 0100	None	None	No Operation
RC		32	0011 | 0010	"0" → C	None	Reset C
SC		22	0010 | 0010	"1" → C	None	Set C
XOR		02	0000 | 0010	A ⊕ RAM(B) → A	None	Exclusive-OR RAM with A
TRANSFER OF CONTROL INSTRUCTIONS						
JID		FF	1111 | 1111	ROM (PC_8, A,M) → $PC_{7:0}$	None	Jump Indirect (Note 2)
JMP	a	6– –	0110 | 000 | a_8 $a_{7:0}$	a → PC	None	Jump
JP	a	–	1 | $a_{6:0}$ (pages 2,3 only) or	a → $PC_{6:0}$	None	Jump within Page (Note 1)
		–	11 | $a_{5:0}$ (all other pages)	a → $PC_{5:0}$		
JSRP	a	–	10 | $a_{5:0}$	PC + 1 → SA → SB 010 → $PC_{8:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 2)
JSR	a	6– –	0110 | 100 | a_8 $a_{7:0}$	PC + 1 → SA → SB a → PC	None	Jump to Subroutine
RET		48	0100 | 1000	SB → SA → PC	None	Return from Subroutine
RETSK		49	0100 | 1001	SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
HALT		33 38	0011 | 0011 0011 | 1000		None	Halt processor

Instruction Set (Continued)

TABLE III. COP210C/211C Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
MEMORY REFERENCE INSTRUCTIONS						
CAMQ		33 3C	0011 | 0011 0011 | 1100	$A \rightarrow Q_{7:4}$ $RAM(B) \rightarrow Q_{3:0}$	None	Copy A, RAM to Q
CQMA		33 2C	0011 | 0011 0010 | 1100	$Q_{7:4} \rightarrow RAM(B)$ $Q_{3:0} \rightarrow A$	None	Copy Q to RAM, A
LD	r	−5	00 | r | 0101	$RAM(B) \rightarrow A$ $Br \oplus r \rightarrow Br$	None	Load RAM into A Exclusive-OR Br with r
LQID		BF	1011 | 1111	$ROM(PC_8,A,M) \rightarrow Q$ $SA \rightarrow SB$	None	Load Q Indirect
RMB	0 1 2 3	4C 45 42 43	0100 | 1100 0100 | 0101 0100 | 0010 0100 | 0011	$0 \rightarrow RAM(B)_0$ $0 \rightarrow RAM(B)_1$ $0 \rightarrow RAM(B)_2$ $0 \rightarrow RAM(B)_3$	None	Reset RAM Bit
SMB	0 1 2 3	4D 47 46 4B	0100 | 1101 0100 | 0111 0100 | 0110 0100 | 1011	$1 \rightarrow RAM(B)_0$ $1 \rightarrow RAM(B)_1$ $1 \rightarrow RAM(B)_2$ $1 \rightarrow RAM(B)_3$	None	Set RAM Bit
STII	y	7−	0111 | y	$y \rightarrow RAM(B)$ $Bd + 1 \rightarrow Bd$	None	Store Memory Immediate and Increment Bd
X	r	−6	00 | r | 0110	$RAM(B) \longleftrightarrow A$ $Br \oplus r \rightarrow Br$	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	3,15	23 BF	0010 | 0011 1011 | 1111	$RAM(3,15) \longleftrightarrow A$	None	Exchange A with RAM (3,15)
XDS	r	−7	00 | r | 0111	$RAM(B) \longleftrightarrow A$ $Bd - 1 \rightarrow Bd$ $Br \oplus r \rightarrow Br$	Bd decrements past 0	Exchange RAM with A and Decrement Bd Exclusive-OR Br with r
XIS	r	−4	00 | r | 0100	$RAM(B) \longleftrightarrow A$ $Bd + 1 \rightarrow Bd$ $Br \oplus r \rightarrow Br$	Bd increments past 15	Exchange RAM with A and Increment Bd Exclusive-OR Br with r
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	0101 | 0000	$A \rightarrow Bd$	None	Copy A to Bd
CBA		4E	0100 | 1110	$Bd \rightarrow A$	None	Copy Bd to A
LBI	r,d	−	00 | r | (d - 1) (d = 0,9:15)	$r,d \rightarrow B$	Skip until not a LBI	Load B Immediate with r,d
LEI	y	33 6−	0011 | 0011 0110 | y	$y \rightarrow EN$	None	Load EN Immediate

Instruction Set (Continued)

TABLE III. COP210C/211C Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
TEST INSTRUCTIONS						
SKC		20	0010 | 0000		C = "1"	Skip if C is True
SKE		21	0010 | 0001		A = RAM(B)	Skip if A Equals RAM
SKGZ		33	0011 | 0011		$G_{3:0} = 0$	Skip if G is Zero
		21	0010 | 0001			(all 4 bits)
SKGBZ		33	0011 | 0011	1st byte		Skip if G Bit is Zero
	0	01	0000 | 0001	2nd byte	$G_0 = 0$	
	1	11	0001 | 0001		$G_1 = 0$	
	2	03	0000 | 0011		$G_2 = 0$	
	3	13	0010 | 0011		$G_3 = 0$	
SKMBZ	0	01	0000 | 0001		$RAM(B)_0 = 0$	Skip if RAM Bit is Zero
	1	11	0001 | 0001		$RAM(B)_1 = 0$	
	2	03	0000 | 0011		$RAM(B)_2 = 0$	
	3	13	0001 | 0011		$RAM(B)_3 = 0$	
INPUT/OUTPUT INSTRUCTIONS						
ING		33	0011 | 0011	G → A	None	Input G Ports to A
		2A	0010 | 1010			
INL		33	0011 | 0011	$L_{7:4}$ → RAM(B)	None	Input L Ports to RAM, A
		2E	0010 | 1110	$L_{3:0}$ → A		
OBD		33	0011 | 0011	Bd → D	None	Output Bd to D Outputs
		3E	0011 | 1110			
OMG		33	0011 | 0011	RAM(B) → G	None	Output RAM to G Ports
		3A	0011 | 1010			
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO

Note 1: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 2: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Description of Selected Instructions

The following information is provided to assist the user in understanding the operation of several unique instructions and to provide notes useful to programmers in writing COP210C/211C programs.

XAS INSTRUCTION

XAS (Exchange A with SIO) exchanges the 4-bit contents of the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. An XAS instruction will also affect the SK output. (See Functional Description, EN Register). If SIO is selected as a shift register, an XAS instruction must be performed once every four instruction cycle times to effect a continuous data stream.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower eight bits of the ROM address register PC with the contents of ROM addressed by the 9-bit word, PC_8, A, M. PC_8 is not affected by this instruction.

Note: JID uses two instruction cycles if executed, one if skipped.

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 9-bit word PC_8, A, M. LQID can be used for table look-up or code conversion such as BCD to 7-segment. The LQID instruction "pushes" the stack (PC + 1 → SA → SB) and replaces the least significant eight bits of the PC as follows: A → $PC_{7:4}$, RAM(B) → $PC_{3:0}$, leaving PC_8 unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SB → SA → PC), restoring the saved value of the PC to continue sequential program execution. Since LQID pushes SA → SB, the previous contents of SB are lost.

Note: LQID uses two instruction cycles if executed, one if skipped.

Description of Selected Instructions (Continued)

INSTRUCTION SET NOTES

a. The first word of a COP210C/211C program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, one instruction cycle time is devoted to skipping each byte of the skipped instruction. Thus all program paths take the same number of cycle times whether instructions are skipped or executed (except JID and LQID).

c. The ROM is organized into eight pages of 64 words each. The program counter is a 9-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID, or LQID instruction is located in the last word of a page, the instruction operates as if it were in the next page. For example: A JP located in the last word of a page will jump to a location in the next page. Also, a LQID or JID located in the last word in page 3 or 7 will access data in the next group of four pages.

POWER DISSIPATION

The lowest power drain is when the clock is stopped. As the frequency increases so does current. Current is also lower at lower operating voltages. Therefore, to minimize power consumption, the user should run at the lowest speed and voltage that his application will allow. The user should take care that all pins swing to full supply levels to ensure that outputs are not loaded down and that inputs are not at some intermediate level which may draw current. Any input with a slow rise or fall time will draw additional current. A crystal- or resonator-generated clock will draw additional current. An RC oscillator will draw even more current since the input is a slow rising signal.

If using an external squarewave oscillator, the following equation can be used to calculate the COP210C current drain.

$Ic = Iq + (V \times 35 \times Fi) + (V \times 2195 \times Fi/Dv)$

where Ic = chip current drain in microamps

Iq = quiescent leakage current (from curve)

Fi = CKI frequency in megahertz

V = chip V_{CC} in volts

Dv = divide by option selected

For example, at 5V V_{CC} and 400 kHz (divide by 4),

$Ic = 10 + (5 \times 35 \times 0.4) + (5 \times 2195 \times 0.4/4)$

$Ic = 10 + 50 + 1097.5 = 1157.5\ \mu A$

I/O OPTIONS

COP210C/211C outputs have the following optional configurations, illustrated in *Figure 7*:

a. Standard. A CMOS push-pull buffer with an N-channel device to ground in conjunction with a P-channel device to V_{CC}, compatible with CMOS and LSTTL.

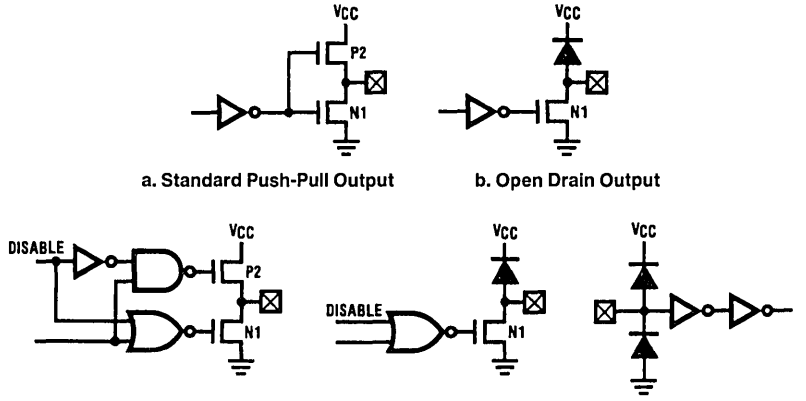
b. Open Drain. An N-channel device to ground only, allowing external pull-up as required by the user's application.

c. Standard TRI-STATE L Output. A CMOS output buffer similar to (a) which may be disabled by program control.

d. Open-Drain TRI-STATE L Output. This has the N-channel device to ground only.

The SI and $\overline{\text{RESET}}$ inputs are Hi-Z inputs (*Figure 7e*).

When using either the G or L I/O ports as inputs, an external pull-up device is necessary.



a. Standard Push-Pull Output

b. Open Drain Output

c. Standard TRI-STATE "L" Output

d. Open Drain TRI-STATE "L" Output

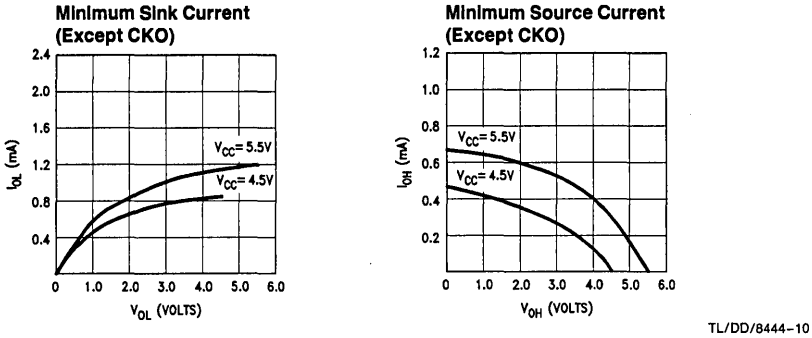
e. Hi-Z Input

TL/DD/8444-9

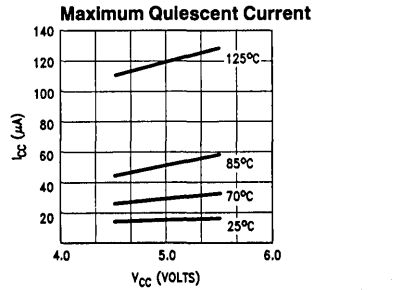
FIGURE 7. I/O Configurations

All output drivers uses one or two common devices numbered 1 to 2. Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figure 8* for each of these devices to allow the designer to effectively use these I/O configurations.

Typical Performance Characteristics



TL/DD/8444–10



TL/DD/8444–11

FIGURE 8

Option List

The COP210C/211C mask-programmable options are assigned numbers which correspond with the COP210C pins.

The following is a list of COP210C options. When specifying a COP211 chip, options 20, 21, and 22 must be set to 0. The options are programmed at the same time as the ROM pattern to provide the user with the hardware flexibility to interface to various I/O components using little or no external circuitry.

Option 1: 0 = Ground Pin. No options available.

Option 2: CKO I/O Port Determined by Option 3. = 0 no option (a. is crystal oscillator output for two pin oscillator b. is HALT I/O for one pin oscillator)

Option 3: CKI Input.
- = 0: Crystal-controlled oscillator input (÷ 4).
- = 1: Single-pin RC-controlled oscillator (÷ 4).
- = 2: External oscillator input (÷ 4).
- = 3: Crystal oscillator input (÷ 8).
- = 4: External oscillator input (÷ 8).
- = 5: Crystal oscillator input (÷ 16).
- = 6: External oscillator input (÷ 16).

Option 4: $\overline{\text{RESET}}$ Input = 1: Hi-Z input. No option available.

Option 5: L_7 Driver
- = 0: Standard TRI-STATE push-pull output.
- = 2: Open-drain TRI-STATE output.

Option 6: L_6 Driver. (Same as Option 5.)

Option 7: L_5 Driver. (Same as Option 5.)

Option 8: L_4 Driver. (Same as Option 5.)

Option 9: V_{CC} Pin = 0 no option.

Option 10: L_3 Driver. (Same as Option 5.)

Option 11: L_2 Driver. (Same as Option 5.)

Option 12: L_1 Driver. (Same as Option 5.)

Option 13: L_0 Driver. (Same as Option 5.)

Option 14: SI Input.
- No option available.
- = 1: Hi-Z input.

Option 15: SO Output.
- = 0: Standard push-pull output.
- = 2: Open-drain output.

Option 16: SK Driver. (Same as Option 15.)

Option 17: G_0 I/O Port. (Same as Option 15.)

Option 18: G_1 I/O Port. (Same as Option 15.)

Option 19: G_2 I/O Port. (Same as Option 15.)

Option 20: G_3 I/O Port. (Same as Option 15.)

Option 21: D_3 Output. (Same as Option 15.)

Option 22: D_2 Output. (Same as Option 15.)

Option 23: D_1 Output. (Same as Option 15.)

Option 24: D_0 Output. (Same as Option 15.)

Option 25: Internal Initialization Logic.
- = 0: Normal operation.
- = 1: No internal initialization logic.

Option 26: No option available.

Option 27: COP Bonding
- = 0: COP210C (24-pin device).
- = 1: COP211C (20-pin device). See Note.
- = 2: COP210C and COP211C. See Note.

Note: If option 27 = 1 or 2 then option 20 must = 0.

Option Table

Please fill out a photocopy of the Option Table and send along with your EPROM.

Option Table

Option 1 Value =	0	is: Ground Pin	
Option 2 Value =	0	is: CKO Pin	
Option 3 Value =		is: CKI Input	
Option 4 Value =	1	is: $\overline{\text{RESET}}$ Input	
Option 5 Value =		is: L_7 Driver	
Option 6 Value =		is: L_6 Driver	
Option 7 Value =		is: L_5 Driver	
Option 8 Value =		is: L_4 Driver	
Option 9 Value =	0	is: V_{CC} Pin	
Option 10 Value =		is: L_3 Driver	
Option 11 Value =		is: L_2 Driver	
Option 12 Value =		is: L_1 Driver	
Option 13 Value =		is: L_0 Driver	
Option 14 Value =	1	is: SI Input	
Option 15 Value =		is: SO Output	
Option 16 Value =		is: SK Driver	
Option 17 Value =		is: G_0 I/O Port	
Option 18 Value =		is: G_1 I/O Port	
Option 19 Value =		is: G_2 I/O Port	
Option 20 Value =		is: G_3 I/O Port	
Option 21 Value =		is: D_3 Output	
Option 22 Value =		is: D_2 Output	
Option 23 Value =		is: D_1 Output	
Option 24 Value =		is: D_0 Output	
Option 25 Value =		is: Internal Initialization Logic	
Option 26 Value =	0	is: No Option	
Option 27 Value =		is: COPS Bonding	

National Semiconductor Corporation

PRELIMINARY

COP224C/COP225C/COP226C/COP244C/COP245C Single-Chip 1k and 2k CMOS Microcontrollers

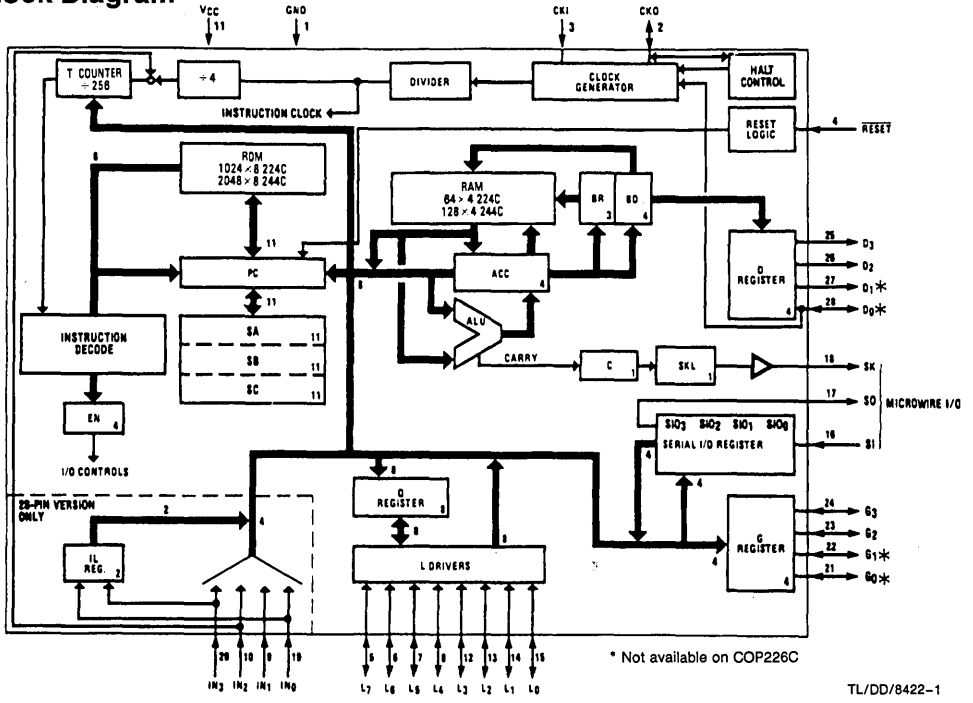
General Description

The COP224C, COP225C, COP226C, COP244C and COP245C fully static, Single-Chip CMOS Microcontrollers are members of the COPS™ family, fabricated using double-poly, silicon gate microCMOS technology. These Controller Oriented Processors are complete microcomputers containing all system timing, internal logic, ROM, RAM, and I/O necessary to implement dedicated control functions in a variety of applications. Features include single supply operation, a variety of output configuration options, with an instruction set, internal architecture and I/O scheme designed to facilitate keyboard input, display output and BCD data manipulation. The COP224C and COP244C are 28 pin chips. The COP225C and COP245C are 24-pin versions (4 inputs removed) and COP226C is 20-pin version with 15 I/O lines. Standard test procedures and reliable high-density techniques provide the medium to large volume customers with a customized microcontroller at a low end-product cost. These microcontrollers are appropriate choices in many demanding control environments especially those with human interface.

Features

- Lowest power dissipation (600 μW typical)
- Fully static (can turn off the clock)
- Power saving IDLE state and HALT mode
- 4.4 μs instruction time
- 2k x 8 ROM, 128 x 4 RAM (COP244C/COP245C)
- 1k x 8 ROM, 64 x 4 RAM (COP224C/COP225C/COP226C)
- 23 I/O lines (COP244C and COP224C)
- True vectored interrupt, plus restart
- Three-level subroutine stack
- Single supply operation (4.5V to 5.5V)
- Programmable read/write 8-bit timer/event counter
- Internal binary counter register with MICROWIRE™ serial I/O capability
- General purpose and TRI-STATE® outputs
- LSTTL/CMOS output compatible
- Software/hardware compatible with COP400 family
- Military temperature (−55°C to +125°C) operation

Block Diagram



* Not available on COP226C

TL/DD/8422–1

FIGURE 1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage (V_{CC})	6V
Voltage at any Pin	$-0.3V$ to $V_{CC}+0.3V$
Total Allowable Source Current	25 mA
Total Allowable Sink Current	25 mA
Total Allowable Power Dissipation	150 mW
Operating Temperature Range	$-55°C$ to $+125°C$
Storage Temperature Range	$-65°C$ to $+150°C$
Lead Temperature (soldering, 10 seconds)	300°C

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $-55°C \le T_A \le +125°C$, $+4.5V \le V_{CC} \le +5.5V$ unless otherwise specified

Parameter	Conditions	Min	Max	Units
Operating Voltage		4.5	5.5	V
Power Supply Ripple (Note 5)	Peak to Peak		0.25 V_{CC}	V
Supply Current (Note 1)	$V_{CC}=5.0V$, tc = 4.4 μs (tc is instruction cycle time)		5000	μA
HALT Mode Current (Note 2)	$V_{CC}=5.0V$, $F_{IN}=0$ kHz		120	μA
Input Voltage Levels				
$\overline{RESET}$, CKI, D_0 (clock input)				
Logic High		0.9 V_{CC}		V
Logic Low			0.1 V_{CC}	V
All Other Inputs				
Logic High		0.7 V_{CC}		V
Logic Low			0.2 V_{CC}	V
Hi-Z Input Leakage		−10	+10	μA
Input Capacitance (Note 4)			7	pF
Output Voltage Levels (except CKO)	Standard Outputs			
LSTTL Operation	$V_{CC}=5.0V \pm 10\%$			
Logic High	$I_{OH}=-100$ μA	2.7		V
Logic Low	$I_{OL}=400$ μA		0.6	V
CMOS Operation				
Logic High	$I_{OH}=-10$ μA	$V_{CC}-0.2$		V
Logic Low	$I_{OL}=10$ μA		0.2	V
CKO Current Levels (As Clock Out)				
Sink ÷4	CKI = V_{CC}, $V_{OUT}=V_{CC}$	0.2		mA
÷8		0.4		mA
÷16		0.8		mA
Source ÷4	CKI = 0V, $V_{OUT}=0V$	0.2		mA
÷8		0.4		mA
÷16		0.8		mA
Allowable Sink/Source Current per Pin (Note 6)			5	mA
Allowable Loading on CKO (as HALT)			50	pF
Current Needed to Over-Ride HALT (Note 3)				
To Continue	$V_{IN}=0.2$ V_{CC}		2.0	mA
To Halt	$V_{IN}=0.7$ V_{CC}		3.0	mA
TRI-STATE or Open Drain Leakage Current		−10	+10	μA

AC Electrical Characteristics $-55°C \leq T_A \leq +125°C$, $+4.5V \leq V_{CC} \leq +5.5V$ unless otherwise specified.

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time (tc)		4.4	DC	μs
Operating CKI ÷4 mode		DC	0.9	MHz
Frequency ÷8 mode		DC	1.8	MHz
÷16 mode		DC	3.6	MHz
Duty Cycle (Note 4)	$f_1 = 3.6$ MHz	40	60	%
Rise Time (Note 4)	$f_1 = 3.6$ MHz External Clock		60	ns
Fall Time (Note 4)	$f_1 = 3.6$ MHz External Clock		40	ns
Instruction Cycle Time	R = 30k ±5%			
RC Oscillator (Note 4)	C = 82 pF ±5% (÷4 Mode)	6	18	μs
Inputs: (See *Figure 3*) (Note 4)				
t_{SETUP}	G Inputs	tc/4 + 0.8		μs
	SI Input	0.33		μs
	All Others	1.9		μs
t_{HOLD}		0.4		μs
Output Propagation Delay	$V_{OUT} = 1.5V$, $C_L = 100$ pF, $R_L = 5k$			
t_{PD1}, t_{PD0}			1.4	μs

Note 1: Supply current is measured after running for 2000 cycle times with a square-wave clock on CKI, CKO open, and all other pins pulled up to V_{CC} with 5k resistors. See current drain equation on page 13.

Note 2: The HALT mode will stop CKI from oscillating in the RC and crystal configurations. Test conditions: all inputs tied to V_{CC}, L lines in TRI-STATE mode and tied to ground, all outputs low and tied to ground.

Note 3: When forcing HALT, current is only needed for a short time (approx. 200 ns) to flip the HALT flip-flop.

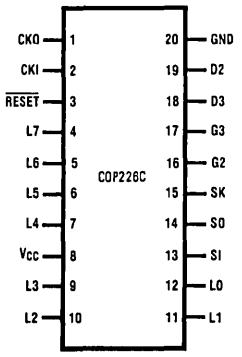
Note 4: This parameter is not tested but guaranteed by design. Variation due to the device included.

Note 5: Voltage change must be less than 0.25 volts in a 1 ms period.

Note 6: SO output sink current must be limited to keep V_{OL} less than 0.2 V_{CC} when part is running in order to prevent entering test mode.

Connection Diagrams

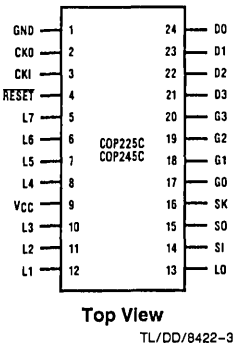
Dual-In-Line Package



Top View

TL/DD/8422-2

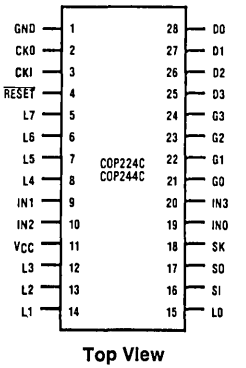
Dual-In-Line Package



Top View

TL/DD/8422-3

Dual-In-Line Package



Top View

TL/DD/8422-4

Pin	Description
L7–L0	8-bit bidirectional port with TRI-STATE
G3–G0	4-bit bidirectional I/O port
D3–D0	4-bit output port
IN3–IN0	4-bit input port (28 pin package only)
SI	Serial input or counter input
SO	Serial or general purpose output
SK	Logic controlled clock output
CKI	Chip oscillator input
CKO	Oscillator output, HALT I/O port or general purpose input
$\overline{\text{RESET}}$	Reset input
V_{CC}	Most positive power supply
GND	Ground

FIGURE 2

Order Number COP226C-XXX/D, COP226C-XXX/N
See NS Package D20A or N20A

Order Number COP225C-XXX/D, COP225C-XXX/N, or COP245C-XXX/D, COP245C-XXX/N
See NS Package D24C or N24A

Order Number COP224C-XXX/D, COP224C-XXX/N, or COP244C-XXX/D, COP244C-XXX/N
See NS Package D28C or N28B

Functional Description

The internal architecture is shown in *Figure 1*. Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1", when a bit is reset, it is a logic "0".

Caution:

The output options available on the COP224C/225C/226C and COP244C/245C are not the same as those available on the COP324C/325C/326C, COP344C/345C, COP424C/425C/426C and COP444C/445C. Options not available on the COP224C/225C/226C and COP244C/245C are: Option 2 value 2; Option 4 value 0; Option 5 value 1; Option 9 value 0; Option 17 value 1; Option 30, Dual Clock, all values; Option 32, Microbus™, all values; Option 33 values 2, 4, and 6; Option 34 all values; and Option 35 all values.

PROGRAM MEMORY

Program Memory consists of ROM, 1024 bytes for the COP224C/225C/226C and 2048 bytes for the COP244C/245C. These bytes of ROM may be program instructions, constants or ROM addressing data.

ROM addressing is accomplished by an 11-bit PC register which selects one of the 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 11-bit binary count value.

Three levels of subroutine nesting are implemented by a three level deep stack. Each subroutine call or interrupt pushes the next PC address into the stack. Each return pops the stack back into the PC register.

DATA MEMORY

Data memory consists of a 512-bit RAM for the COP244C/245C, organized as 8 data registers of 16 × 4-bit digits. RAM addressing is implemented by a 7-bit B register whose upper 3 bits (Br) select 1 of 8 data registers and lower 4 bits (Bd) select 1 of 16 4-bit digits in the selected data register.

Data memory consists of a 256-bit RAM for the COP224C/225C/226C, organized as 4 data registers of 16 × 4-bits digits. The B register is 6 bits long. Upper 2 bits (Br) select 1 of 4 data registers and lower 4 bits (Bd) select 1 of 16 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) are usually loaded into or from, or exchanged with, the A register (accumulator), it may also be loaded into or from the Q latches or T counter or loaded from the L ports. RAM addressing may also be performed directly by the LDD and XAD instructions based upon the immediate operand field of these instructions.

The Bd register also serves as a source register for 4-bit data sent directly to the D outputs.

INTERNAL LOGIC

The processor contains its own 4-bit A register (accumulator) which is the source and destination register for most I/O, arithmetic, logic, and data memory access operations. It can also be used to load the Br and Bd portions of the B register, to load and input 4 bits of the 8-bit Q latch or T counter, to input 4 bits of L I/O ports data, to input 4-bit G, or IN ports, and to perform data exchanges with the SIO register.

A 4-bit adder performs the arithmetic and logic functions, storing the results in A. It also outputs a carry bit to the 1-bit C register, most often employed to indicate arithmetic overflow. The C register in conjunction with the XAS instruction and the EN register, also serves to control the SK output.

The 8-bit T counter is a binary up counter which can be loaded to and from M and A using CAMT and CTMA instructions. This counter may be operated in two modes depending on a mask-programmable option: as a timer or as an external event counter. When the T counter overflows, an

Functional Description (Continued)

overflow flag will be set (see SKT and IT instructions below). The T counter is cleared on reset. A functional block diagram of the timer/counter is illustrated in *Figure 7*.

Four general-purpose inputs, IN3–IN0, are provided.

The D register provides 4 general-purpose outputs and is used as the destination register for the 4-bit contents of Bd.

The G register contents are outputs to a 4-bit general-purpose bidirectional I/O port.

The Q register is an internal, latched, 8-bit register, used to hold data loaded to or from M and A, as well as 8-bit data from ROM. Its contents are outputted to the L I/O ports when the L drivers are enabled under program control.

The 8 L drivers, when enabled, output the contents of latched Q data to the L I/O port. Also, the contents of L may be read directly into A and M.

The SIO register functions as a 4-bit serial-in/serial-out shift register for MICROWIRE I/O and COPS peripherals, or as a binary counter (depending on the contents of the EN register). Its contents can be exchanged with A.

The XAS instruction copies C into the SKL latch. In the counter mode, SK is the output of SKL; in the shift register mode, SK outputs SKL ANDed with the clock.

EN is an internal 4-bit register loaded by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register:

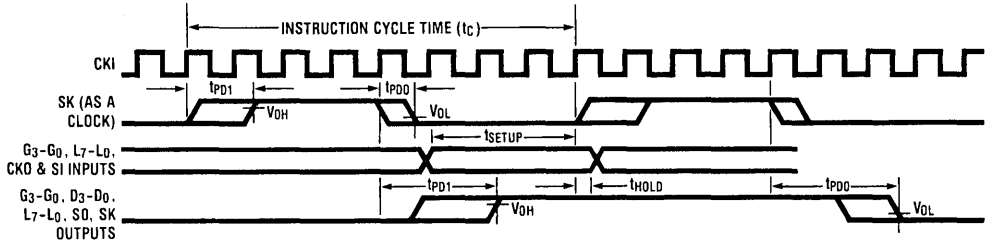
0. The least significant bit of the enable register, EN0, selects the SIO register as either a 4-bit shift register or a 4-bit binary counter. With EN0 set, SIO is an asynchronous binary counter, decrementing its value by one upon each low-going pulse ("1" to "0") occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output equals the value of EN3. With EN0 reset, SIO is a serial shift register left shifting 1 bit each instruction cycle time. The data present at SI goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. The SK outputs SKL ANDed with the instruction cycle clock.
1. With EN1 set, interrupt is enabled. Immediately following an interrupt, EN1 is reset to disable further interrupts.
2. With EN2 set, the L drivers are enabled to output the data in Q to the L I/O port. Resetting EN2 disables the L drivers, placing the L I/O port in a high-impedance input state.
3. EN3, in conjunction with EN0, affects the SO output. With EN0 set (binary counter option selected) SO will output the value loaded into EN3. With EN0 reset (serial shift register option selected), setting EN3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN3 with the serial shift register option selected disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains set to "0".

INTERRUPT

The following features are associated with interrupt procedure and protocol and must be considered by the programmer when utilizing interrupts.

a. The interrupt, once recognized as explained below, pushes the next sequential program counter address (PC+1) onto the stack. Any previous contents at the bottom of the stack are lost. The program counter is set to hex address 0FF (the last word of page 3) and EN1 is reset.

b. An interrupt will be recognized only on the following conditions:
   1. EN1 has been set.
   2. A low-going pulse ("1" to "0") at least two instruction cycles wide has occurred on the IN_1 input.
   3. A currently executing instruction has been completed.



TL/DD/8422–5

FIGURE 3. Input/Output Timing Diagrams (divide by 8 mode)

TABLE I. Enable Register Modes — Bits EN0 and EN3

EN0	EN3	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL=1,SK=clock If SKL=0,SK=0
0	1	Shift Register	Input to Shift Register	Serial out	If SKL=1,SK=clock If SKL=0,SK=0
1	0	Binary Counter	Input to Counter	0	SK=SKL
1	1	Binary Counter	Input to Counter	1	SK=SKL

Functional Description (Continued)

4. All successive transfer of control instructions and successive LBIs have been completed (e.g. if the main program is executing a JP instruction which transfers program control to another JP instruction, the interrupt will not be acknowledged until the second JP instruction has been executed).

c. Upon acknowledgement of an interrupt, the skip logic status is saved and later restored upon popping of the stack. For example, if an interrupt occurs during the execution of ASC (Add with Carry, Skip on Carry) instruction which results in carry, the skip logic status is saved and program control is transferred to the interrupt servicing routine at hex address 0FF. At the end of the interrupt routine, a RET instruction is executed to pop the stack and return program control to the instruction following the original ASC. At this time, the skip logic is enabled and skips this instruction because of the previous ASC carry. Subroutines should not be nested within the interrupt service routine, since their popping of the stack will enable any previously saved main program skips, interfering with the orderly execution of the interrupt routine.

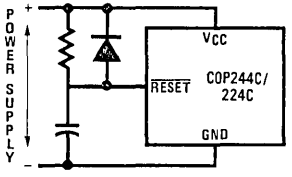
d. The instruction at hex address 0FF must be a NOP.

e. An LEI instruction may be put immediately before the RET instruction to re-enable interrupts.

INITIALIZATION

The internal reset logic will initialize the device upon power-up if the power supply rise time is less than 1 ms and if the operating frequency at CKI is greater than 32 kHz, otherwise the external RC network shown in *Figure 4* must be connected to the $\overline{\text{RESET}}$ pin (the conditions in *Figure 4* must be met). The $\overline{\text{RESET}}$ pin is configured as a Schmitt trigger input. If not used, it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the $\overline{\text{RESET}}$ input, providing it stays low for at least three instruction cycle times.

Note: If CKI clock is less than 32 kHz, the internal reset logic (option #29 = 1) MUST be disabled and the external RC circuit must be used.



TL/DD/8422-6

FIGURE 4. Power-Up Circuit

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN, IL, T and G registers are cleared. The SKL latch is set, thus enabling SK as a clock output. Data Memory (RAM) is not cleared upon initialization. The first instruction at address 0 must be a CLRA (clear A register).

TIMER

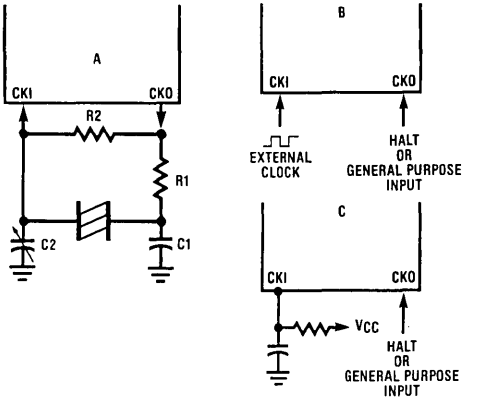
There are two modes selected by mask option:

a. Time-base counter. In this mode, the instruction cycle frequency generated from CKI passes through a 2-bit divide-by-4 prescaler. The output of this prescaler increments the 8-bit T counter thus providing a 10-bit timer. The prescaler is cleared during execution of a CAMT instruction and on reset.

For example, using a 3.58 MHz crystal with a divide-by-16 option, the instruction cycle frequency of 223.70 kHz increments the 10-bit timer every 4.47 μs. By presetting the counter and detecting overflow, accurate timeouts between 17.88 μs (4 counts) and 4.577 ms (1024 counts) are possible. Longer timeouts can be achieved by accumulating, under software control, multiple overflows.

b. External event counter. In this mode, a low-going pulse ("1" to "0") at least 2 instruction cycles wide on the IN2 input will increment the 8-bit T counter.

Note: The IT instruction is not allowed in this mode.



TL/DD/8422-7

Crystal or Resonator

Crystal Value	Component Values			
	R1	R2	C1(pF)	C2(pF)
32 kHz	220k	20M	30	6-36
455 kHz	5k	10M	80	40
2.096 MHz	2k	1M	30	6-36
3.6 MHz	1k	1M	30	6-36

RC Controlled Oscillator

R	C	Cycle Time	V_{CC}
30k	82 pF	6-18 μs	≥4.5V

Note: 15k≤R≤150k
50 pF≤C≤150 pF

FIGURE 5. Oscillator Component Values

Functional Description (Continued)

HALT MODE

The COP244C/245C/224C/225C/226C is a FULLY STATIC circuit; therefore, the user may stop the system oscillator at any time to halt the chip. The chip may also be halted by the HALT instruction or by forcing CKO high when it is mask-programmed as a HALT I/O port. Once in the HALT mode, the internal circuitry does not receive any clock signal and is therefore frozen in the exact state it was in when halted. All information is retained until continuing. The chip may be awakened by one of two different methods:

- Continue function: by forcing CKO low, if it mask-programmed as a HALT I/O port, the system clock is re-enabled and the circuit continues to operate from the point where it was stopped.
- Restart: by forcing the $\overline{\text{RESET}}$ pin low (see Initialization).

The HALT mode is the minimum power dissipation state.

CKO PIN OPTIONS

a. Two-pin oscillator—(Crystal). See *Figure 6a*.

In a crystal controlled oscillator system, CKO is used as an output to the crystal network. The HALT mode may be entered by program control (HALT instruction) which forces CKO high, thus inhibiting the crystal network. The circuit can be awakened only by forcing the $\overline{\text{RESET}}$ pin to a logic "0" (restart).

b. One-pin oscillator—(RC or external). See *Figure 6b*.

If a one-pin oscillator system is chosen, two options are available for CKO:

- CKO can be selected as the HALT I/O port. In that case, it is an I/O flip-flop which is an indicator of the HALT status. An external signal can over-ride this pin to start and stop the chip. By forcing a high level to CKO, the chip will stop as soon as CKI is high and CKO output will stay high to keep the chip stopped if the external driver returns to high impedance state. By forcing a low level to CKO, the chip will continue and CKO will stay low.
- As another option, CKO can be a general purpose input, read into bit 2 of A (accumulator) upon execution of an INIL instruction.

OSCILLATOR OPTIONS

There are three basic clock oscillator configurations available as shown by *Figure 5*.

a. Crystal Controlled Oscillator. CKI and CKO are connected to an external crystal. The instruction cycle time equals the crystal frequency optionally divided by 4, 8 or 16.

b. External Oscillator. The external frequency is optionally divided by 4, 8 or 16 to give the instruction cycle time. CKO is the HALT I/O port or a general purpose input.

c. RC Controlled Oscillator. CKI is configured as a single pin RC controlled Schmitt trigger oscillator. The instruction cycle equals the oscillation frequency divided by 4. CKO is the HALT I/O port or a general purpose input.

Figure 7 shows the clock and timer diagram.

COP245C AND COP225C 24-PIN PACKAGE OPTION

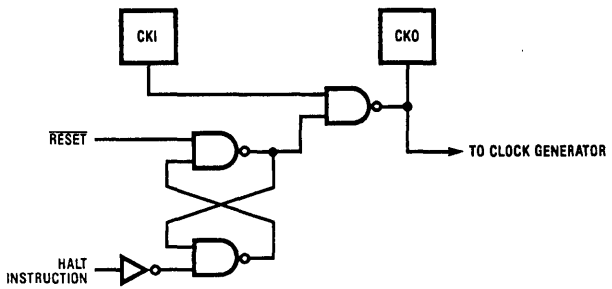
If the COP244C/224C is bonded in a 24-pin package, it becomes the COP245C/225C, illustrated in *Figure 2*, Connection diagrams. Note that the COP245C/225C does not contain the four general purpose IN inputs (IN3–IN0). Use of this option precludes, of course, use of the IN options, interrupt feature, external event counter feature.

Note: If user selects the 24-pin package, options 9, 10, 19 and 20 must be selected as a "2". See option list.

COP226C 20-PIN PACKAGE OPTION

If the COP225C is bonded as 20-pin device it becomes the COP226C. Note that the COP226C contains all the COP225C pins except D_0, D_1, G_0, and G_1.

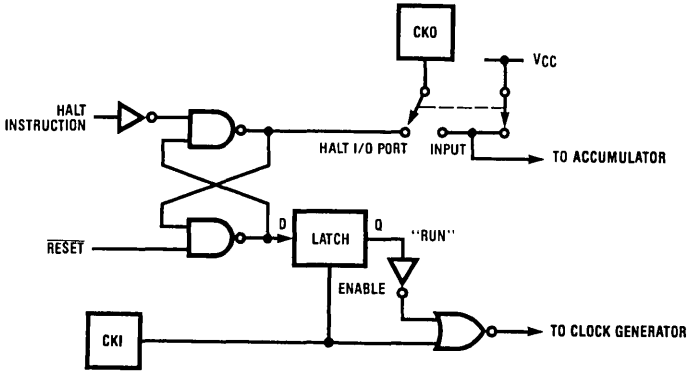
Block Diagram (Continued)



TL/DD/8422–8

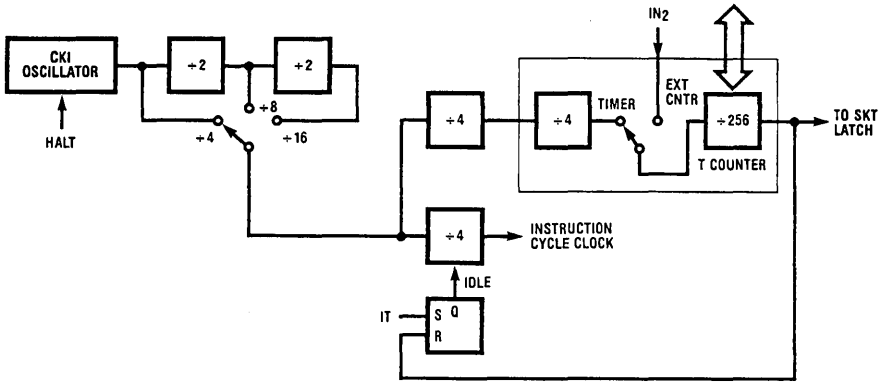
FIGURE 6a. Halt Mode—Two-Pin Oscillator

Block Diagrams (Continued)



TL/DD/8422-9

FIGURE 6b. Halt Mode—One-Pin Oscillator



TL/DD/8422-10

FIGURE 7. Clock and Timer

Instruction Set

Table II is a symbol table providing internal architecture, instruction operand and operation symbols used in the instruction set table.

TABLE II. Instruction Set Table Symbols

Symbol	Definition
Internal Architecture Symbols	
A	4-bit accumulator
B	7-bit RAM address register (6-bit for COP224C)
Br	Upper 3 bits of B (register address) (2-bit for COP224C)
Bd	Lower 4 bits of B (digit address)
C	1-bit carry register
D	4-bit data output port
EN	4-bit enable register
G	4-bit general purpose I/O port
IL	two 1-bit (IN0 and IN3) latches
IN	4-bit input port
L	8-bit TRI-STATE I/O port
M	4-bit contents of RAM addressed by B
PC	11-bit ROM address program counter
Q	8-bit latch for L port
SA,SB,SC	11-bit 3-level subroutine stack
SIO	4-bit shift register and counter
SK	Logic-controlled clock output
SKL	1-bit latch for SK output
T	8-bit timer

Table III provides the mnemonic, operand, machine code data flow, skip conditions and description of each instruction.

Instruction Operand Symbols	
d	4-bit operand field, 0–15 binary (RAM digit select)
r	3(2)-bit operand field, 0–7(3) binary (RAM register select)
a	11-bit operand field, 0–2047 (1023)
y	4-bit operand field, 0–15 (immediate data)
RAM(x)	RAM addressed by variable x
ROM(x)	ROM addressed by variable x

Operational Symbols	
+	Plus
−	Minus
→	Replaces
⟷	Is exchanged with
=	Is equal to
$\overline{A}$	One's complement of A
⊕	Exclusive-or
:	Range of values

TABLE III. COP244C/245C Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	0011 | 0000	A+C+RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	0011 | 0001	A+RAM(B) → A	None	Add RAM to A
ADT		4A	0100 | 1010	A+10_{10} → A	None	Add Ten to A
AISC	y	5–	0101 | y	A+y → A	Carry	Add Immediate. Skip on Carry (y ≠ 0)
CASC		10	0001 | 0000	$\overline{A}$+RAM(B)+C → A Carry → C	Carry	Complement and Add with Carry, Skip on Carry
CLRA		00	0000 | 0000	0 → A	None	Clear A
COMP		40	0100 | 0000	$\overline{A}$ → A	None	Ones complement of A to A
NOP		44	0100 | 0100	None	None	No Operation
RC		32	0011 | 0010	"0" → C	None	Reset C
SC		22	0010 | 0010	"1" → C	None	Set C
XOR		02	0000 | 0010	A⊕RAM(B) → A	None	Exclusive-OR RAM with A

Instruction Set (Continued)

TABLE III. COP244C/245C Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
TRANSFER CONTROL INSTRUCTIONS						
JID		FF	1111 | 1111	ROM ($PC_{10:8}$ A,M) → $PC_{7:0}$	None	Jump Indirect (Notes 1, 3)
JMP	a	6–	0110 | 0 | $a_{10:8}$	a → PC	None	Jump
		––	$a_{7:0}$			
JP	a	––	1 | $a_{6:0}$ (pages 2, 3 only)	a → $PC_{6:0}$	None	Jump within Page (Note 4)
			or			
		––	11 | $a_{5:0}$ (all other pages)	a → $PC_{5:0}$		
JSRP	a	––	10 | $a_{5:0}$	PC+1 → SA → SB → SC 00010 → $PC_{10:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 5)
JSR	a	6–	0110 | 1 | $a_{10:8}$	PC+1 → SA → SB → SC	None	Jump to Subroutine
		––	$a_{7:0}$	a → PC		
RET		48	0100 | 1000	SC → SB → SA → PC	None	Return from Subroutine
RETSK		49	0100 | 1001	SC → SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
HALT		33	0011 | 0011		None	HALT Processor
		38	0011 | 1000			
IT		33	0011 | 0011			IDLE till Timer
		39	0011 | 1001		None	Overflows then Continues
MEMORY REFERENCE INSTRUCTIONS						
CAMT		33	0011 | 0011	A → $T_{7:4}$		
		3F	0011 | 1111	RAM(B) → $T_{3:0}$	None	Copy A, RAM to T
CTMA		33	0011 | 0011	$T_{7:4}$ → RAM(B)		
		2F	0010 | 1111	$T_{3:0}$ → A	None	Copy T to RAM, A
CAMQ		33	0011 | 0011	A → $Q_{7:4}$	None	Copy A, RAM to Q
		3C	0011 | 1100	RAM(B) → $Q_{3:0}$		
CQMA		33	0011 | 0011	$Q_{7:4}$ → RAM(B)	None	Copy Q to RAM, A
		2C	0010 | 1100	$Q_{3:0}$ → A		
LD	r	–5	00 | r | 0101 (r=0:3)	RAM(B) → A Br⊕r → Br	None	Load RAM into A, Exclusive-OR Br with r
LDD	r,d	23	0010 | 0011	RAM(r,d) → A	None	Load A with RAM pointed to directly by r,d
		––	0 | r | d			
LQID		BF	1011 | 1111	ROM($PC_{10:8}$,A,M) → Q SB → SC	None	Load Q Indirect (Note 3)
RMB	0	4C	0100 | 1100	0 → $RAM(B)_0$	None	Reset RAM Bit
	1	45	0100 | 0101	0 → $RAM(B)_1$		
	2	42	0100 | 0010	0 → $RAM(B)_2$		
	3	43	0100 | 0011	0 → $RAM(B)_3$		
SMB	0	4D	0100 | 1101	1 → $RAM(B)_0$	None	Set RAM Bit
	1	47	0100 | 0111	1 → $RAM(B)_1$		
	2	46	0100 | 0110	1 → $RAM(B)_2$		
	3	4B	0100 | 1011	1 → $RAM(B)_3$		

Instruction Set (Continued)

TABLE III. COP244C/245C Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
MEMORY REFERENCE INSTRUCTIONS (Continued)						
STII	y	7–	0111 | y	y → RAM(B) Bd + 1 → Bd	None	Store Memory Immediate 1 and Increment Bd
X	r	–6	00 | r | 0110 (r=0:3)	RAM(B) ⟷ A Br ⊕ r → Br	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	r,d	23 – –	0010 | 0011 1 | r | d	RAM(r,d) ⟷ A	None	Exchange A with RAM Pointed to Directly by r,d
XDS	r	–7	00 | r | 0111 (r=0:3)	RAM(B) ⟷ A Bd–1 → Bd Br ⊕ r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd. Exclusive-OR Br with r
XIS	r	–4	00 | r | 0100 (r=0:3)	RAM(B) ⟷ A Bd+1 → Bd Br ⊕ r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd, Exclusive-OR Br with r
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	0101 | 0000	A → Bd	None	Copy A to Bd
CBA		4E	0100 | 1110	Bd → A	None	Copy Bd to A
LBI	r,d	– –	00 | r | (d–1) (r=0:3: d=0,9:15) or 0011 | 0011 1 | r | d (any r, any d)	r,d → B	Skip until not a LBI	Load B Immediate with r,d (Note 6)
LEI	y	33 6–	0011 | 0011 0110 | y	y → EN	None	Load EN Immediate (Note 7)
XABR		12	0001 | 0010	A ⟷ Br	None	Exchange A with Br (Note 8)
TEST INSTRUCTIONS						
SKC		20	0010 | 0000		C="1"	Skip if C is True
SKE		21	0010 | 0001		A=RAM(B)	Skip if A Equals RAM
SKGZ		33 21	0011 | 0011 0010 | 0001		$G_{3:0}=0$	Skip if G is Zero (all 4 bits)
SKGBZ		33	0011 | 0011	1st byte		Skip if G Bit is Zero
	0	01	0000 | 0001	2nd byte	$G_0=0$	
	1	11	0001 | 0001		$G_1=0$	
	2	03	0000 | 0011		$G_2=0$	
	3	13	0001 | 0011		$G_3=0$	
SKMBZ	0	01	0000 | 0001		$RAM(B)_0=0$	Skip if RAM Bit is Zero
	1	11	0001 | 0001		$RAM(B)_1=0$	
	2	03	0000 | 0011		$RAM(B)_2=0$	
	3	13	0001 | 0011		$RAM(B)_3=0$	
SKT		41	0100 | 0001		A time-base counter carry has occurred since last test	Skip on Timer (Note 3)

Instruction Set (Continued)

TABLE III. COP244C/245C Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
INPUT/OUTPUT INSTRUCTIONS						
ING		33	0011 | 0011	G → A	None	Input G Ports to A
		2A	0010 | 1010			
ININ		33	0011 | 0011	IN → A	None	Input IN Inputs to A (Note 2)
		28	0010 | 1000			
INIL		33	0011 | 0011	IL_3, CKO,"0", IL_0 → A	None	Input IL Latches to A (Note 3)
		29	0010 | 1001			
INL		33	0011 | 0011	$L_{7:4}$ → RAM(B) $L_{3:0}$ → A	None	Input L Ports to RAM,A
		2E	0010 | 1110			
OBD		33	0011 | 0011	Bd → D	None	Output Bd to D Outputs
		3E	0011 | 1110			
OGI	y	33	0011 | 0011	y → G	None	Output to G Ports Immediate
		5–	0101 | y			
OMG		33	0011 | 0011	RAM(B) → G	None	Output RAM to G Ports
		3A	0011 | 1010			
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO (Note 3)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicitly defined (e.g., Br and Bd are explicitly defined). Bits are numbered 0 to N where 0 signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit A register.

Note 2: The ININ instruction is not available on the 24-pin packages since these devices do not contain the IN inputs.

Note 3: For additional information on the operation of the XAS, JID, LQID, INIL, and SKT instructions, see below.

Note 4: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 5: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 6: LBI is a single-byte instruction if d = 0, 9, 10, 11, 12, 13, 14, or 15. The machine code for the lower 4 bits equals the binary value of the "d" data *minus 1*, e.g., to load the lower four bits of B(Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Note 7: Machine code for operand field y for LEI instruction should equal the binary value to be latched into EN, where a "1" or "0" in each bit of EN corresponds with the selection or deselection of a particular function associated with each bit. (See Functional Description, EN Register.)

Note 8: For 2K ROM devices, A ⟷ Br (0 → A3). For 1K ROM devices, A ⟷ Br (0,0 → A3, A2).

Description of Selected Instructions

XAS INSTRUCTION

XAS (Exchange A with SIO) copies C to the SKL latch and exchanges the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. If SIO is selected as a shift register, an XAS instruction can be performed once every 4 instruction cycles to effect a continuous data stream.

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 11-bit word PC10:PC8,A,M. LQID can be used for table lookup or code conversion such as BCD to seven-segment. The LQID instruction "pushes" the stack (PC+1 → SA → SB → SC) and replaces the least significant 8 bits of the PC as follows: A → PC7:4, RAM(B) → PC3:0, leaving PC10, PC9 and PC8 unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SC → SB → SA → PC), restoring the saved value of PC to continue sequential program execution. Since LQID pushes SB → SC, the previous contents of SC are lost.

Note: LQID uses 2 instruction cycles if executed, one if skipped.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower 8 bits of the ROM address register PC with the contents of ROM addressed by the 11-bit word, PC10:8,A,M. PC10,PC9 and PC8 are not affected by JID.

Note: JID uses 2 instruction cycles if executed, one if skipped.

SKT INSTRUCTION

The SKT (Skip On Timer) instruction tests the state of the T counter overflow latch (see internal logic, above), executing the next program instruction if the latch is not set. If the latch has been set since the previous test, the next program instruction is skipped and the latch is reset. The features associated with this instruction allow the processor to generate its own time-base for real-time processing, rather than relying on an external input signal.

Note: If the most significant bit of the T counter is a 1 when a CAMT instruction loads the counter, the overflow flag will be set. The following sample of codes should be used when loading the counter:

```
CAMT   ; load T counter
SKT    ; skip if overflow flag is set and reset it
NOP
```

IT INSTRUCTION

The IT (idle till timer) instruction halts the processor and puts it in an idle state until the time-base counter overflows. This idle state reduces current drain since all logic (except the oscillator and time base counter) is stopped. IT instruction is not allowed if the T counter is mask-programmed as an external event counter (option #31 = 1).

INIL INSTRUCTION

INIL (Input IL Latches to A) inputs 2 latches, IL3 and IL0, CKO and 0 into A. The IL3 and IL0 latches are set if a low-going pulse ("1" to "0") has occurred on the IN3 and IN0 inputs since the last INIL instruction, provided the input pulse stays low for at least two instruction cycles. Execution of an INIL inputs IL3 and IL0 into A3 and A0 respectively, and resets these latches to allow them to respond to subsequent low-going pulses on the IN3 and IN0 lines. If CKO is mask programmed as a general purpose input, an INIL will input the state of CKO into A2. If CKO has not been so programmed, a "1" will be placed in A2. A0 is input into A1. IL latches are cleared on reset. IL latches are not available on the COP245C/225C, and COP226C.

INSTRUCTION SET NOTES

a. The first word of a program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, they are still fetched from the program memory. Thus program paths take the same number of cycles whether instructions are skipped or executed except for JID, and LQID.

c. The ROM is organized into pages of 64 words each. The Program Counter is a 11-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID, or LQID is the last word of a page, it operates as if it were in the next page. For example: a JP located in the last word of a page will jump to a location in the next page. Also, a JID or LQID located in the last word of every fourth page (i.e. hex address 0FF, 1FF, 2FF, 3FF, 4FF, etc.) will access data in the next group of four pages.

Note: The COP224C/225C/226C needs only 10 bits to address its ROM. Therefore, the eleventh bit (P10) is ignored.

Power Dissipation

The lowest power drain is when the clock is stopped. As the frequency increases so does current. Current is also lower at lower operating voltages. Therefore, the user should run at the lowest speed and voltage that his application will allow. The user should take care that all pins swing to full supply levels to insure that outputs are not loaded down and that inputs are not at some intermediate level which may draw current. Any input with a slow rise or fall time will draw additional current. A crystal or resonator generated clock input will draw additional current. For example, a 500 kHz crystal input will typically draw 100 μA more than a squarewave input. An R/C oscillator will draw even more current since the input is a slow rising signal.

If using an external squarewave oscillator, the following equation can be used to calculate operating current drain.

$I_{CO} = I_Q + V \times 70 \times Fi + V \times 2400 \times Fi/Dv$ where:

I_{CO} = chip operating current drain in microamps

I_Q = quiescent leakage current (from curve)

Fi = CKI frequency in MegaHertz

V = chip V_{CC} in volts

Dv = divide by option selected

For example at 5 volts V_{CC} and 400 kHz (divide by 4)

$I_{CO} = 120 + 5 \times 70 \times 0.4 + 5 \times 2400 \times 0.4/4$

$I_{CO} = 120 + 140 = 800\ \mu A$

Power Dissipation (Continued)

If an IT instruction is executed, the chip goes into the IDLE mode until the timer overflows. In IDLE mode, the current drain can be calculated from the following equation:

$Ici = I_Q + V \times 70 \times Fi$

For example, at 5 volts V_{CC} and 400 kHz

$Ici = 120 + 5 \times 70 \times 0.4 = 260\ \mu A$

The total average current will then be the weighted average of the operating current and the idle current:

$$Ita = I_{CO} \times \frac{To}{To+Ti} + Ici \times \frac{Ti}{To+Ti}$$

where: Ita = total average current
I_{CO} = operating current
Ici = idle current
To = operating time
Ti = idle time

I/O OPTIONS

Outputs have the following optional configurations, illustrated in *Figure 8*:

a. Standard — A CMOS push-pull buffer with an N-channel device to ground in conjunction with a P-channel device to V_{CC}, compatible with CMOS and LSTTL.

b. Open Drain — An N-channel device to ground only, allowing external pull-up as required by the user's application.

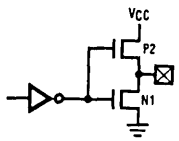
c. Standard TRI-STATE L Output — A CMOS output buffer similar to a. which may be disabled by program control.

d. Open-Drain TRI-STATE L Output — This has the N-channel device to ground only.

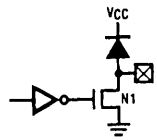
All inputs have the following option:

e. Hi-Z input which must be driven by the users logic.

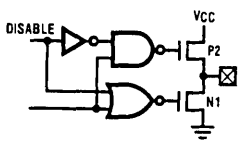
All output drivers use two common devices numbered 1 to 2. Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figure 9* for each of these devices to allow the designer to effectively use these I/O configurations.



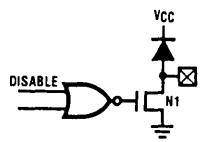
a. Standard Push-Pull Output



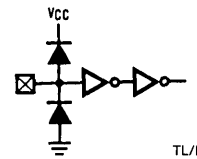
b. Open-Drain Output



c. Standard TRI-STATE "L" Output



d. Open Drain TRI-STATE "L" Output

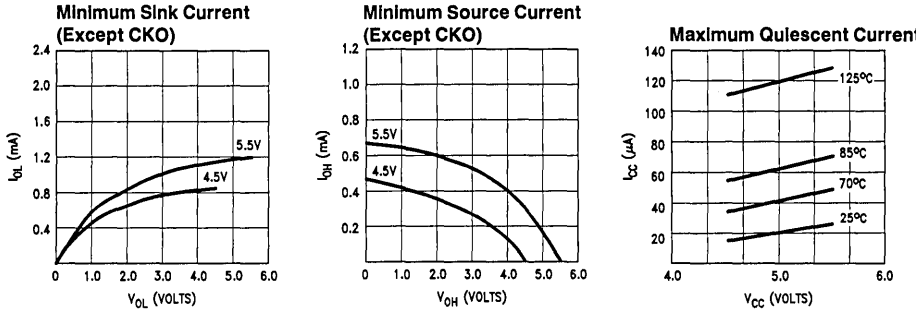


TL/DD/8422–11

e. Hi-Z Input

FIGURE 8. Input/Output Configurations

Power Dissipation (Continued)



TL/DD/8422-12

FIGURE 9. Input/Output Characteristics

Option List

The COP244C/245C/224C/225C/COP226C mask-programmable options are assigned numbers which correspond with the COP244C/224C pins.

The following is a list of options. The options are programmed at the same time as the ROM pattern to provide the user with the hardware flexibility to interface to various I/O components using little or no external circuitry.

Caution:

The output options available on the COP224C/225C/226C and COP244C/245C are not the same as those available on the COP324C/325C/326C, COP344C/345C, COP424C/425C/426C and COP444C/445C. Options not available on the COP224C/225C/226C and COP244C/245C are: Option 2 value 2; Option 4 value 0; Option 5 value 1; Option 9 value 0; Option 17 value 1; Option 30, Dual Clock, all values; Option 32, Microbus, all values; Option 33 values 2 4, and 6; Option 34 all values; and Option 35 all values.

PLEASE FILL OUT THE OPTION TABLE on the next page. Photocopy the option data and send it in with your disk or EPROM.

Option 1 = 0: Ground Pin — no options available

Option 2: CKO Pin
- = 0: clock generator output to crystal/resonator
- = 1: HALT I/O port
- = 3: general purpose input, high-Z

Option 3: CKI input
- = 0: Crystal controlled oscillator input divide by 4
- = 1: Crystal controlled oscillator input divide by 8
- = 2: Crystal controlled oscillator input divide by 16
- = 4: Single-pin RC controlled oscillator (divide by 4)
- = 5: External oscillator input divide by 4
- = 6: External oscillator input divide by 8
- = 7: External oscillator input divide by 16

Option 4: $\overline{\text{RESET}}$ input
- = 1: Hi-Z input

Option 5: L7 Driver
- = 0: Standard TRI-STATE push-pull output
- = 2: Open-drain TRI-STATE output

Option 6: L6 Driver — (same as option 5)

Option 7: L5 Driver — (same as option 5)

Option 8: L4 Driver — (same as option 5)

Option 9: IN1 input
- = 1: Hi-Z input, mandatory for 28 Pin Package
- = 2: Mandatory for 20 and 24 Pin Packages

Option 10: IN2 input — (same as option 9)

Option 11 = 0: V_{CC} Pin — no option available

Option 12: L3 Driver — (same as option 5)

Option 13: L2 Driver — (same as option 5)

Option 14: L1 Driver — (same as option 5)

Option 15: L0 Driver — (same as option 5)

Option 16: SI input — (same as option 4)

Option 17: SO Driver
- = 0: Standard push-pull output
- = 2: Open-drain output

Option 18: SK Driver — (same as option 17)

Option 19: IN0 Input — (same as option 9)

Option 20: IN3 Input — (same as option 9)

Option 21: G0 I/O Port — (same as option 17)

Option 22: G1 I/O Port — (same as option 17)

Option 23: G2 I/O Port — (same as option 17)

Option 24: G3 I/O Port — (same as option 17)

Option 25: D3 Output — (same as option 17)

Option 26: D2 Output — (same as option 17)

Option 27: D1 Output — (same as option 17)

Option List (Continued)

Option 28: D0 Output — (same as option 17)

Option 29: Internal Initialization Logic

=0: Normal operation

=1: No internal initialization logic

Option 30=0: No Option Available

Option 31: Timer

=0: Time-base counter

=1: External event counter

Option 32=0: No Option Available

Option 33: COP bonding. See note.

(1k and 2k Microcontroller)

=0: 28-pin package

=1: 24-pin package

(1k Microcontroller only)

=3: 20-pin package

=5: 24- and 20-pin package

Note:—If opt. #33=0 then opt. #9, 10, 19, and 20 must=1.

If opt. #33=1 then opt. #9, 10, 19 and 20 must=2, and option #31 must=0.

If opt. #33=3 or 5 then opt. #9, 10, 19, 20 must=2 and opt. #21, 22, 31 must=0.

Option 34=0: No Option Available

Option 35=0: No Option Available

Option Table

The following option information is to be sent to National along with the EPROM.

	OPTION DATA			OPTION DATA	
OPTION 1 VALUE =	0	IS: GROUND PIN	OPTION 19 VALUE =		IS: IN0 INPUT
OPTION 2 VALUE =		IS: CKO PIN	OPTION 20 VALUE =		IS: IN3 INPUT
OPTION 3 VALUE =		IS: CKI INPUT	OPTION 21 VALUE =		IS: G0 I/O PORT
OPTION 4 VALUE =	1	IS: $\overline{\text{RESET}}$ INPUT	OPTION 22 VALUE =		IS: G1 I/O PORT
OPTION 5 VALUE =		IS: L7 DRIVER	OPTION 23 VALUE =		IS: G2 I/O PORT
OPTION 6 VALUE =		IS: L6 DRIVER	OPTION 24 VALUE =		IS: G3 I/O PORT
OPTION 7 VALUE =		IS: L5 DRIVER	OPTION 25 VALUE =		IS: D3 OUTPUT
OPTION 8 VALUE =		IS: L4 DRIVER	OPTION 26 VALUE =		IS: D2 OUTPUT
OPTION 9 VALUE =		IS: IN1 INPUT	OPTION 27 VALUE =		IS: D1 OUTPUT
OPTION 10 VALUE =		IS: IN2 INPUT	OPTION 28 VALUE =		IS: D0 OUTPUT
OPTION 11 VALUE =	0	IS: VCC PIN	OPTION 29 VALUE =		IS: INT INIT LOGIC
OPTION 12 VALUE =		IS: L3 DRIVER	OPTION 30 VALUE =	0	IS: N/A
OPTION 13 VALUE =		IS: L2 DRIVER	OPTION 31 VALUE =		IS: TIMER
OPTION 14 VALUE =		IS: L1 DRIVER	OPTION 32 VALUE =	0	IS: N/A
OPTION 15 VALUE =		IS: L0 DRIVER	OPTION 33 VALUE =		IS: COP BONDING
OPTION 16 VALUE =	1	IS: SI INPUT	OPTION 34 VALUE =	0	IS: N/A
OPTION 17 VALUE =		IS: SO DRIVER	OPTION 35 VALUE =	0	IS: N/A
OPTION 18 VALUE =		IS: SK DRIVER			

National Semiconductor Corporation

microCMOS

COP410C/COP411C/COP310C/COP311C Single-Chip CMOS Microcontrollers

General Description

The COP410C, COP411C, COP310C, and COP311C fully static, single-chip CMOS microcontrollers are members of the COPS™ family, fabricated using double-poly, silicon-gate CMOS technology. These controller-oriented processors are complete microcomputers containing all system timing, internal logic, ROM, RAM, and I/O necessary to implement dedicated control functions in a variety of applications. Features include single supply operation, a variety of output configuration options, with an instruction set, internal architecture, and I/O scheme designed to facilitate keyboard input, display output, and BCD data manipulation. The COP411C is identical to the COP410C but with 16 I/O lines instead of 20. They are an appropriate choice for use in numerous human interface control environments. Standard test procedures and reliable high-density fabrication techniques provide the medium to large volume customers with a customized controller-oriented processor at a low end-product cost.

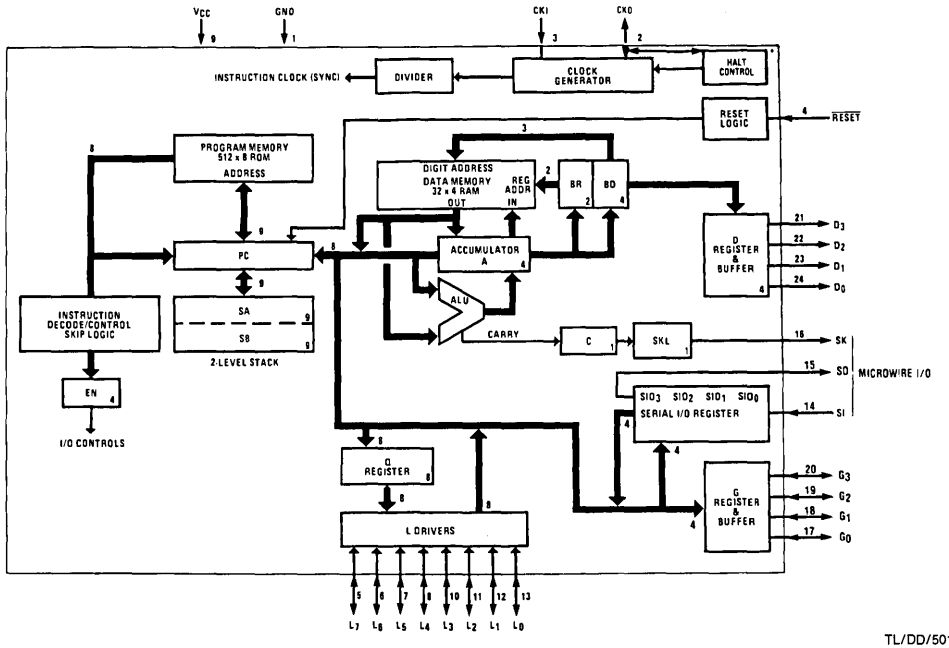
The COP310C/COP311C is the extended temperature range version of the COP410C/COP411C.

The COP404C should be used for exact emulation.

Features

- Lowest power dissipation (40 μW typical)
- Low cost
- Power-saving HALT Mode with Continue function
- Powerful instruction set
- 512 x 8 ROM, 32 x 4 RAM
- 20 I/O lines (COP410C)
- Two-level subroutine stack
- DC to 4 μs instruction time
- Single supply operation (2.4V to 5.5V)
- General purpose and TRI-STATE® outputs
- Internal binary counter register with MICROWIRE™ compatible serial I/O
- LSTTL/CMOS compatible in and out
- Software/hardware compatible with other members of the COP400 family
- Extended temperature (−40°C to +85°C) devices available
- The military temperature range devices (−55°C to +125°C) are specified on COP210C/211C data sheet.

Block Diagram



TL/DD/5015-1

FIGURE 1. COP410C

COP410C/COP411C

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage	6V
Voltage at Any Pin	−0.3V to V_{CC}+0.3V
Total Allowable Source Current	25 mA
Total Allowable Sink Current	25 mA
Operating Temperature Range	0°C to +70°C
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $0°C \le T_A \le 70°C$ unless otherwise specified

Parameter	Conditions	Min	Max	Units
Operating Voltage		2.4	5.5	V
Power Supply Ripple[5]			0.1 V_{CC}	V
Supply Current	V_{CC} = 2.4V, t_c = 125 μs		80	μA
	V_{CC} = 5.0V, t_c = 16 μs		500	μA
	V_{CC} = 5.0V, t_c = 4 μs		2000	μA
	(t_c is instruction cycle time)			
HALT Mode Current[2]	V_{CC} = 5.0V, F_{IN} = 0 kHz		20	μA
	V_{CC} = 2.4V, F_{IN} = 0 kHz		7	μA
Input Voltage Levels				
RESET, CKI				
Logic High		0.9 V_{CC}		V
Logic Low			0.1 V_{CC}	V
All Other Inputs				
Logic High		0.7 V_{CC}		V
Logic Low			0.2 V_{CC}	V
Hi-Z Input Leakage		−1	+1	μA
Input Capacitance			7	pF
Output Voltage Levels	Standard Outputs			
LSTTL Operation	V_{CC} = 5.0V ±10%			
Logic High	I_{OH} = −25 μA	2.7		V
Logic Low	I_{OL} = 400 μA		0.4	V
CMOS Operation				
Logic High	I_{OH} = −10 μA	V_{CC}−0.2		V
Logic Low	I_{OL} = 10 μA		0.2	V
Output Current Levels[4] (Except CKO)				
Sink	V_{CC} = 4.5V, V_{OUT} = V_{CC}	1.2		mA
	V_{CC} = 2.4V, V_{OUT} = V_{CC}	0.2		mA
Source (Standard Option)	V_{CC} = 4.5V, V_{OUT} = 0V	−0.5		mA
	V_{CC} = 2.4V, V_{OUT} = 0V	−0.1		mA
Source (Low Current Option)	V_{CC} = 4.5V, V_{OUT} = 0V	−30	−330	μA
	V_{CC} = 2.4V, V_{OUT} = 0V	−6	−80	μA
CKO Current Levels (As Clock Out)				
Sink ÷4	V_{CC} = 4.5V, CKI = V_{CC}, V_{OUT} = V_{CC}	0.3		mA
÷8		0.6		mA
÷16		1.2		mA
Source ÷4	V_{CC} = 4.5V, CKI = 0V, V_{OUT} = 0V	−0.3		mA
÷8		−0.6		mA
÷16		−1.2		mA
Allowable Sink/Source Current Per Pin[4]			5	mA

1

COP410C/COP411C

DC Electrical Characteristics (Continued)

Parameter	Conditions	Min	Max	Units
Allowable Loading on CKO (as HALT I/O pin)			100	pF
Current Needed to Override HALT[3]				
To Continue	V_{CC} = 4.5V, V_{IN} = 0.2 V_{CC}		0.6	mA
To Halt	V_{CC} = 4.5V, V_{IN} = 0.7 V_{CC}		1.6	mA
TRI-STATE or Open Drain Leakage Current		−2	+2	μA

Note 1: Supply current is measured after running for 2000 cycle times with a square-wave clock on CKI, CKO open, and all other pins pulled up to V_{CC} with 5k resistors. See current drain equation on page 13.

Note 2: The Halt mode will stop CKI from oscillating in the RC and crystal configurations.

Note 3: When forcing HALT, current is only needed for a short time (approximately 200 ns) to flip the HALT flip-flop.

Note 4: SO output sink current must be limited to keep V_{OL} less than 0.2 V_{CC} when part is running in order to prevent entering test mode.

Note 5: Voltage change must be less than 0.5V in a 1 ms period.

Note 6: This parameter is only sampled and not 100% tested.

Note 7: Variation due to the device included.

COP410C/COP411C

AC Electrical Characteristics $0°C \le T_A \le 70°C$ unless otherwise specified

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time (t_c)	$V_{CC} \ge 4.5V$	4	DC	μs
	4.5V > $V_{CC} \ge$ 2.4V	16	DC	μs
Operating CKI Frequency ÷4 mode	$V_{CC} \ge 4.5V$	DC	1.0	MHz
÷8 mode	$V_{CC} \ge 4.5V$	DC	2.0	MHz
÷16 mode	$V_{CC} \ge 4.5V$	DC	4.0	MHz
÷4 mode	4.5V > $V_{CC} \ge$ 2.4V	DC	250	kHz
÷8 mode	4.5V > $V_{CC} \ge$ 2.4V	DC	500	kHz
÷16 mode	4.5V > $V_{CC} \ge$ 2.4V	DC	1.0	MHz
Instruction Cycle Time RC Oscillator[7]	R = 30k ± 5%, V_{CC} = 5V C = 82 pF ± 5% (÷4 Mode)	8	16	μs
Duty Cycle[6]	f_I = 4 MHz	40	60	%
Rise Time[6]	f_I = 4 MHz External Clock		60	ns
Fall Time[6]	f_I = 4 MHz External Clock		40	ns
Inputs (See *Figure 3*)				
t_{SETUP} G Inputs	$V_{CC} \ge 4.5V$	tc/4+0.7		μs
t_{SETUP} SI Input	$V_{CC} \ge 4.5V$	0.3		μs
t_{SETUP} All Others	$V_{CC} \ge 4.5V$	1.7		μs
t_{HOLD}	$V_{CC} \ge 4.5V$	0.25		μs
	$V_{CC} \ge$ 2.4V	1.0		μs
Output Propagation Delay	V_{OUT} = 1.5V, C_L = 100 pF, R_L = 5k			
t_{PD1}, t_{PD0}	$V_{CC} \le 4.5V$		1.0	μs
t_{PD1}, t_{PD0}	$V_{CC} \le$ 2.4V		4.0	μs

COP310C/COP311C

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage	6V
Voltage at Any Pin	−0.3V to V_{CC} + 0.3V
Total Allowable Source Current	25 mA
Total Allowable Sink Current	25 mA
Operating Temperature Range	−40°C to +85°C
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $-40°C \le T_A \le +85°C$ unless otherwise specified

Parameter	Conditions	Min	Max	Units
Operating Voltage		3.0	5.5V	V
Power Supply Ripple[5]			0.1 V_{CC}	V
Supply Current	V_{CC} = 3.0V, t_c = 125 μs V_{CC} = 5.0V, t_c = 16 μs V_{CC} = 5.0V, t_c = 4 μs (t_c is instruction cycle time)		100 600 2500	μA μA μA
HALT Mode Current[2]	V_{CC} = 5.0V, F_{IN} = 0 kHz V_{CC} = 3.0V, F_{IN} = 0 kHz		20 10	μA μA
Input Voltage Levels RESET, CKI Logic High Logic Low All Other Inputs Logic High Logic Low		 0.9 V_{CC} 0.7 V_{CC}	 0.1 V_{CC} 0.2 V_{CC}	 V V V V
Hi-Z Input Leakage		−2	+2	μA
Input Capacitance			7	pF
Output Voltage Levels LSTTL Operation Logic High Logic Low CMOS Operation Logic High Logic Low	Standard Outputs V_{CC} = 5.0V ±10% I_{OH} = −25 μA I_{OL} = 400 μA I_{OH} = −10 μA I_{OL} = 10 μA	 2.7 V_{CC}−0.2	 0.4 0.2	 V V V V
Output Current Levels[4] (Except CKO) Sink Source (Standard Option) Source (Low Current Option) 	 V_{CC} = 4.5V, V_{OUT} = V_{CC} V_{CC} = 3.0V, V_{OUT} = V_{CC} V_{CC} = 4.5V, V_{OUT} = 0V V_{CC} = 3.0V, V_{OUT} = 0V V_{CC} = 4.5V, V_{OUT} = 0V V_{CC} = 3.0V, V_{OUT} = 0V	 1.2 0.2 −0.5 −0.1 −30 −8	 −440 −200	 mA mA mA mA μA μA
CKO Current Levels (As Clock Out) Sink ÷4 ÷8 ÷16 Source ÷4 ÷8 ÷16	 V_{CC} = 4.5V, CKI = V_{CC}, V_{OUT} = V_{CC} V_{CC} = 4.5V, CKI = 0V, V_{OUT} = 0V 	 0.3 0.6 1.2 −0.3 −0.6 −1.2	 	 mA mA mA mA mA mA
Allowable Sink/Source Current Per Pin[4]			5	mA

COP310C/COP311C

DC Electrical Characteristics (Continued)

Parameter	Conditions	Min	Max	Units
Allowable Loading on CKO (as HALT I/O pin)			100	pF
Current Needed to Override HALT[3]				
To Continue	V_{CC} = 4.5V, V_{IN} = 0.2 V_{CC}		0.8	mA
To Halt	V_{CC} = 4.5V, V_{IN} = 0.7 V_{CC}		2.0	mA
TRI-STATE or Open Drain Leakage Current		−4	+4	μA

Note 1: Supply current is measured after running for 2000 cycle times with a square-wave clock on CKI, CKO open, and all other pins pulled up to V_{CC} with 5k resistors. See current drain equation on page 13.

Note 2: The Halt mode will stop CKI from oscillating in the RC and crystal configurations.

Note 3: When forcing HALT, current is only needed for a short time (approximately 200 ns) to flip the HALT flip-flop.

Note 4: SO output sink current must be limited to keep V_{OL} less than 0.2 V_{CC} when part is running in order to prevent entering test mode.

Note 5: Voltage change must be less than 0.5V in a 1 ms period.

Note 6: This parameter is only sampled and not 100% tested.

Note 7: Variation due to the device included.

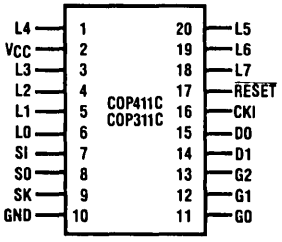
COP310C/COP311C

AC Electrical Characteristics $-40°C \le T_A \le +85°C$ unless otherwise specified

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time (t_c)	$V_{CC} \ge 4.5V$	4	DC	μs
	4.5V > $V_{CC} \ge$ 3.0V	16	DC	μs
Operating CKI ÷4 mode	$V_{CC} \ge 4.5V$	DC	1.0	MHz
Frequency ÷8 mode	$V_{CC} \ge 4.5V$	DC	2.0	MHz
÷16 mode	$V_{CC} \ge 4.5V$	DC	4.0	MHz
÷4 mode	4.5V > $V_{CC} \ge$ 3.0V	DC	250	kHz
÷8 mode	4.5V > $V_{CC} \ge$ 3.0V	DC	500	kHz
÷16 mode	4.5V > $V_{CC} \ge$ 3.0V	DC	1.0	MHz
Instruction Cycle Time RC Oscillator[7]	R = 30k ±5%, V_{CC} = 5V C = 82 pF ±5% (÷4 Mode)	8	16	μs
Duty Cycle[6]	f_I = 4 MHz	40	60	%
Rise Time[6]	f_I = 4 MHz External Clock		60	ns
Fall Time[6]	f_I = 4 MHz External Clock		40	ns
Inputs (See *Figure 3*)				
t_{SETUP} G Inputs	$V_{CC} \ge 4.5V$	tc/4+0.7		μs
t_{SETUP} SI Input	$V_{CC} \ge 4.5V$	0.3		μs
t_{SETUP} All Others	$V_{CC} \ge 4.5V$	1.7		μs
t_{HOLD}	$V_{CC} \ge 4.5V$	0.25		μs
	$V_{CC} \ge 3.0V$	1.0		μs
Output Propagation Delay	V_{OUT} = 1.5V, C_L = 100 pF, R_L = 5k			
t_{PD1}, t_{PD0}	$V_{CC} \le 4.5V$		1.0	μs
t_{PD1}, t_{PD0}	$V_{CC} \le 3.0V$		4.0	μs

Connection Diagrams

Dual-In-Line Package

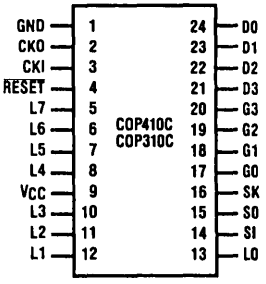


TL/DD/5015-2

Top View

Order Number COP311C-XXX/D, COP411C-XXX/D, COP311C-XXX/N or COP411C-XXX/N, COP311C-XXX/N or COP411C-XXX/N
See NS Package Number D20A, M20B or N20A

Dual-In-Line Package



TL/DD/5015-3

Top View

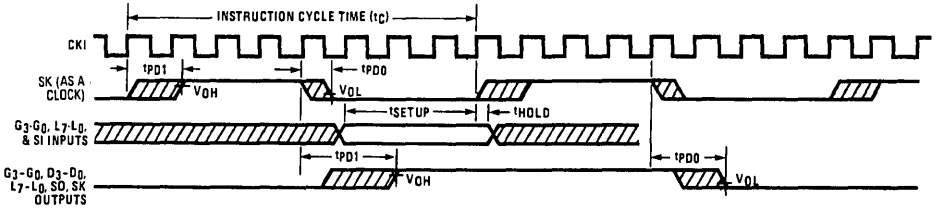
Order Number COP310C-XXX/D, COP410C-XXX/D, COP310C-XXX/M or COP410C-XXX/M, COP310C-XXX/N or COP410C-XXX/N
See NS Package Number D24C, M24B or N24A

FIGURE 2

Pin Descriptions

Pin	Description	Pin	Description
L_7–L_0	8-bit bidirectional I/O port with TRI-STATE	SK	Logic-controlled clock (or general purpose output)
G_3–G_0	4-bit bidirectional I/O port (G_2–G_0 for 20-pin package)	CKI	System oscillator input
D_3–D_0	4-bit general purpose output port (D_1–D_0 for 20-pin package)	CKO	Crystal oscillator output, or HALT mode I/O port (24-pin package only)
SI	Serial input (or counter input)	RESET	System reset input
SO	Serial output (or general purpose output)	V_{CC}	System power supply
		GND	System Ground

Timing Diagram



TL/DD/5015-4

FIGURE 3. Input/Output (Divide-by-8 Mode)

Functional Description

To ease reading of this description, only COP410C and/or COP411C are referenced; however, all such references apply equally to COP310C and/or COP311C, respectively.

A block diagram of the COP410C is given in *Figure 1.* Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1"; when a bit is reset, it is a logic "0".

PROGRAM MEMORY

Program memory consists of a 512-byte ROM. As can be seen by an examination of the COP410C/411C instruction set, these words may be program instructions, program data, or ROM addressing data. Because of the special characteristics associated with the JP, JSRP, JID, and LQID instructions, ROM must often be thought of as being organized into 8 pages of 64 words (bytes) each.

ROM ADDRESSING

ROM addressing is accomplished by a 9-bit PC register. Its binary value selects one of the 512 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 9-bit binary count value. Two levels of subroutine nesting are implemented by two 9-bit subroutine save registers, SA and SB.

ROM instruction words are fetched, decoded, and executed by the instruction decode, control and skip logic circuitry.

DATA MEMORY

Data Memory consists of a 128-bit RAM, organized as four data registers of 8 × 4-bit digits. RAM addressing is implemented by a 6-bit B register whose upper two bits (Br) selects one of four data registers and lower three bits of the 4-bit Bd select one of eight 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) are usually loaded into or from, or exchanged with, the A register (accumulator), they may also be loaded into the Q latches or loaded from the L ports. RAM addressing may also be performed directly by the XAD 3, 15 instruction. The Bd register also serves as a source register for 4-bit data sent directly to the D outputs.

The most significant bit of Bd is not used to select a RAM digit. Hence, each physical digit of RAM may be selected by two different values of Bd as shown in *Figure 4.* The skip condition for XIS and XDS instructions will be true if Bd changes between 0 to 15, but *not* between 7 and 8 (see Table III).

INTERNAL LOGIC

The internal logic of the COP410C/411C is designed to ensure fully static operation of the device.

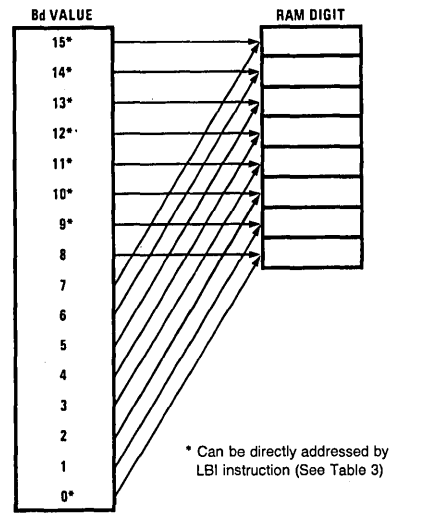
The 4-bit A register (accumulator) is the source and destination register for most I/O, arithmetic, logic and data memory access operations. It can also be used to load the Bd portion of the B register, to load four bits of the 8-bit Q latch data and to perform data exchanges with the SIO register.

The 4-bit adder performs the arithmetic and logic functions of the COP410C/411C, storing its results in A. It also outputs the carry information to a 1-bit carry register, most often employed to indicate arithmetic overflow. The C register, in conjunction with the XAS instruction and the EN register, also serves to control the SK output. C can be outputted directly to SK or can enable SK to be a sync clock each instruction cycle time. (See XAS instruction and EN register description below.)

The G register contents are outputs to four general purpose bidirectional I/O ports.

The Q register is an internal, latched, 8-bit register, used to hold data loaded from RAM and A, as well as 8-bit data from ROM. Its contents are output to the L I/O ports when the L drivers are enabled under program control. (See LEI instruction.)

The eight L drivers, when enabled, output the contents of latched Q data to the L I/O ports. Also, the contents of L may be read directly into A and RAM.



TL/DD/5015–5

FIGURE 4. RAM Digit Address to Physical RAM Digit Mapping

Functional Description (Continued)

The SIO register functions as a 4-bit serial-in/serial-out shift register or as a binary counter, depending upon the contents of the EN register. (See EN register description below.) Its contents can be exchanged with A, allowing it to input or output a continuous serial data stream. With SIO functioning as a serial-in/serial-out shift register and SK as a sync clock, the COP410C/411C is MICROWIRE compatible.

The D register provides four general purpose outputs and is used as the destination register for the 4-bit contents of Bd.

The XAS instruction copies C into the SKL latch. In the counter mode, SK is the output of SKL; in the shift register mode, SK is a sync clock, inhibited when SKL is a logic "0".

The EN register is an internal 4-bit register loaded under program control by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register (EN3–EN0).

1. The least significant bit of the enable register, EN0, selects the SIO register as either a 4-bit shift register or as a 4-bit binary counter. With EN0 set, SIO is an asynchronous binary counter, *decrementing* its value by one upon each low-going pulse ("1" to "0") occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output is equal to the value of EN3. With EN0 reset, SIO is a serial shift register, shifting left each instruction cycle time. The data present at SI is shifted into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each instruction cycle time. (See 4, below.) The SK output becomes a logic-controlled clock.
2. EN 1 is not used, it has no effect on the COP410C/411C.
3. With EN2 set, the L drivers are enabled to output the data in Q to the L I/O ports. Resetting EN2 disables the L drivers, placing the L I/O ports in a high impedance input state.
4. EN3, in conjunction with EN0, affects the SO output. With EN0 set (binary counter option selected), SO will output the value loaded into EN3. With EN0 reset (serial shift register option selected), setting EN3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN3 with the serial shift register option selected, disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains reset to "0".

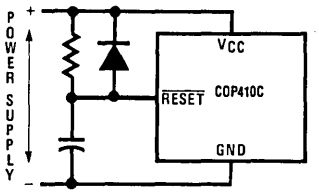
INITIALIZATION

The internal reset logic will initialize the device upon power-up if the power supply rise time is less than 1 ms and if the operating frequency at CKI is greater than 32 kHz, otherwise the external RC network shown in *Figure 5* must be connected to the $\overline{\text{RESET}}$ pin. The $\overline{\text{RESET}}$ pin is configured as a Schmitt trigger input. If not used, it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the $\overline{\text{RESET}}$ input, providing it stays low for at least three instruction cycle times.

When V_{CC} power is applied, the internal reset logic will keep the chip in initialization mode for up to 2500 instruction cycles. If the CKI clock is running at a low frequency, this could take a long time, therefore, the internal logic should be disabled by a mask option with initialization controlled solely by $\overline{\text{RESET}}$ pin.

Note: If CKI clock is less than 32 kHz, the internal reset logic (Option 25 = 1) *must* be disabled and the external RC network *must* be present.

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN, and G registers are cleared. The SK output is enabled as a SYNC output, providing a pulse each instruction cycle time. Data memory (RAM) is not cleared upon initialization. The first instruction at address 0 must be a CLRA (clear A register).



TL/DD/5015–6

RC > 5 × Power Supply Rise Time
and RC > 100 × CKI Period

FIGURE 5. Power-Up Clear Circuit

COP411C

If the COP410C is bonded as a 20-pin package, it becomes the COP411C, illustrated in *Figure 2,* COP410C/411C Connection Diagrams. Note that the COP411C does not contain D2, D3, G3, or CKO. Use of this option, of course, precludes use of D2, D3, G3, and CKO options. All other options are available for the COP411C.

TABLE I. Enable Register Modes — Bits EN0 and EN3

EN0	EN3	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL = 1, SK = clock If SKL = 0, SK = 0
0	1	Shift Register	Input to Shift Register	Serial out	If SKL = 1, SK = clock If SKL = 0, SK = 0
1	0	Binary Counter	Input to Counter	0	SK = SKL
1	1	Binary Counter	Input to Counter	1	SK = SKL

Functional Description (Continued)

HALT MODE

The COP410C/411C is a *fully static* circuit; therefore, the user may stop the system oscillator at any time to halt the chip. The chip also may be halted by the HALT instruction or by forcing CKO high when it is used as a HALT I/O port. Once in the HALT mode, the internal circuitry does not receive any clock signal, and is therefore frozen in the exact state it was in when halted. All information is retained until continuing. The HALT mode is the minimum power dissipation state.

The HALT mode has slight differences depending upon the type of oscillator used.

a. 1-pin oscillator—RC or external

The HALT mode may be entered into by either program control (HALT instruction) or by forcing CKO to a logic "1" state.

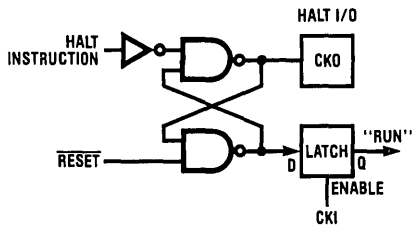
The circuit may be awakened by one of two different methods:

1) Continue function. By forcing CKO to a logic "0", the system clock is re-enabled and the circuit continues to operate from the point where it was stopped.

2) Restart. Forcing the $\overline{\text{RESET}}$ pin to a logic "0" will restart the chip regardless of HALT or CKO (see initialization).

b. 2-pin oscillator—crystal

The HALT mode may be entered into by program control (HALT instruction) which forces CKO to a logic "1" state. The circuit can be awakened only by the $\overline{\text{RESET}}$ function.



TL/DD/5015-7

Halt I/O Port

CKO Pin Options

In a crystal-controlled oscillator system, CKO is used as an output to the crystal network. CKO will be forced high during the execution of a HALT instruction, thus inhibiting the crystal network. If a 1-pin oscillator system is chosen (RC or external), CKO will be selected as HALT and is an I/O flip-flop which is an indicator of the HALT status. An external signal can override this pin to start and stop the chip. By forcing a high level to CKO, the chip will stop as soon as CKI is high and the CKO output will go high to keep the chip stopped. By forcing a low level to CKO, the chip will continue and CKO output will go low.

All features associated with the CKO I/O pin are available with the 24-pin package only.

OSCILLATOR OPTIONS

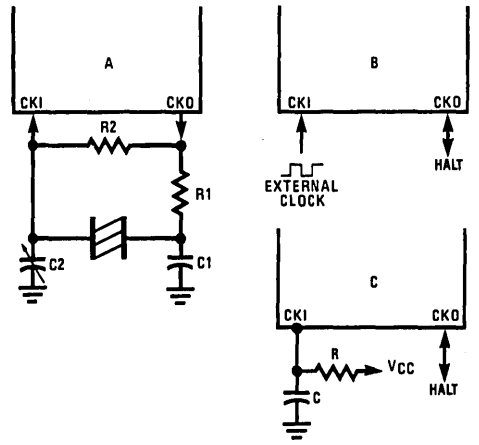
There are three options available that define the use of CKI and CKO.

a. Crystal-Controlled Oscillator. CKI and CKO are connected to an external crystal. The instruction cycle time equals the crystal frequency divided by 16 (optionally by 8 or 4).

b. External Oscillator. CKI is configured as LSTTL-compatible input accepting an external clock signal. The external frequency is divided by 16 (optionally by 8 or 4) to give the instruction cycle time. CKO is the HALT I/O port.

c. RC-Controlled Oscillator. CKI is configured as a single pin RC-controlled Schmitt trigger oscillator. The instruction cycle equals the oscillation frequency divided by 4. CKO is the HALT I/O port.

The RC oscillator is not recommended in systems that require accurate timing or low current. The RC oscillator draws more current than an external oscillator (typically an additional 100 μA at 5V). However, when the part halts, it stops with CKI high and the halt current is at the minimum.



TL/DD/5015-8

FIGURE 6. COP410C Oscillator

Crystal or Resonator

Crystal Value	Component Value R1	R2	C1 pF	C2 pF
32 kHz	220k	20M	30	5-36
455 kHz	5k	10M	80	40
2.096 MHz	2k	1M	30	6-36
4.0 MHz	1k	1M	30	6-36

RC-Controlled Oscillator

R	C	Cycle Time	V_{CC}
15k	82 pF	4-9 μs	$\geq$4.5V
30k	82 pF	8-16 μs	$\geq$4.5V
47k	100 pF	16-32 μs	2.4 to 4.5
Note: 15k $\leq$ R $\leq$ 150k, 50 pf $\leq$ C $\leq$ 150 pF			

COP410C/COP411C Instruction Set

Table II is a symbol table providing internal architecture, instruction operand and operational symbols used in the instruction set table.

Table III provides the mnemonic, operand, machine code, data flow, skip conditions and description associated with each instruction in the COP410C/411C instruction set.

TABLE II. COP410C/411C Instruction Set Table Symbols

Symbol	Definition
INTERNAL ARCHITECTURE SYMBOLS	
A	4-bit Accumulator
B	6-bit RAM Address Register
Br	Upper 2 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry Register
D	4-bit Data Output Port
EN	4-bit Enable Register
G	4-bit Register to latch data for G I/O Port
L	8-bit TRI-STATE I/O Port
M	4-bit contents of RAM Memory pointed to by B Register
PC	9-bit ROM Address Register (program counter)
Q	8-bit Register to latch data for L I/O Port
SA	9-bit Subroutine Save Register A
SB	9-bit Subroutine Save Register B
SIO	4-bit Shift Register and Counter
SK	Logic-Controlled Clock Output

Symbol	Definition
INSTRUCTION OPERAND SYMBOLS	
d	4-bit Operand Field, 0-15 binary (RAM Digit Select)
r	2-bit Operand Field, 0-3 binary (RAM Register Select)
a	9-bit Operand Field, 0-511 binary (ROM Address)
y	4-bit Operand Field, 0-15 binary (Immediate Data)
RAM(s)	Contents of RAM location addressed by s
ROM(t)	Contents of ROM location addressed by t
OPERATIONAL SYMBOLS	
+	Plus
−	Minus
→	Replaces
↔	Is exchanged with
=	Is equal to
$\overline{A}$	The one's complement of A
⊕	Exclusive-OR
:	Range of values

TABLE III. COP410C/411C Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	|0011|0000|	A + C + RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	|0011|0001|	A + RAM(B) → A	None	Add RAM to A
AISC	y	5−	|0101| y |	A + y → A	Carry	Add immediate, Skip on Carry (y ≠ 0)
CLRA		00	|0000|0000|	0 → A	None	Clear A
COMP		40	|0100|0000|	$\overline{A}$ → A	None	One's complement of A to A
NOP		44	|0100|0100|	None	None	No Operation
RC		32	|0011|0010|	"0" → C	None	Reset C
SC		22	|0010|0010|	"1" → C	None	Set C
XOR		02	|0000|0010|	A ⊕ RAM(B) → A	None	Exclusive-OR RAM with A

Instruction Set (Continued)

TABLE III. COP410C/411C Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
TRANSFER OF CONTROL INSTRUCTIONS						
JID		FF	1111 | 1111	ROM (PC_8, A,M) → $PC_{7:0}$	None	Jump Indirect (Note 2)
JMP	a	6–	0110 | 000 | a_8	a → PC	None	Jump
		–	$a_{7:0}$			
JP	a	–	1 | $a_{6:0}$ (pages 2,3 only)	a → $PC_{6:0}$	None	Jump within Page (Note 1)
			or			
		–	11 | $a_{5:0}$ (all other pages)	a → $PC_{5:0}$		
JSRP	a	–	10 | $a_{5:0}$	PC + 1 → SA → SB 010 → $PC_{8:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 2)
JSR	a	6–	0110 | 100 | a_8	PC + 1 → SA → SB a → PC	None	Jump to Subroutine
		–	$a_{7:0}$			
RET		48	0100 | 1000	SB → SA → PC	None	Return from Subroutine
RETSK		49	0100 | 10011	SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
HALT		33	0011 | 0011		None	Halt processor
		38	0011 | 1000			
MEMORY REFERENCE INSTRUCTIONS						
CAMQ		33	0011 | 0011	A → $Q_{7:4}$ RAM(B) → $Q_{3:0}$	None	Copy A, RAM to Q
		3C	0011 | 1100			
CQMA		33	0011 | 0011	$Q_{7:4}$ → RAM(B) $Q_{3:0}$ → A	None	Copy Q to RAM, A
		2C	0010 | 1100			
LD	r	–5	00 | r | 0101	RAM(B) → A Br ⊕ r → Br	None	Load RAM into A Exclusive-OR Br with r
LQID		BF	1011 | 1111	ROM(PC_8,A,M) → Q SA → SB	None	Load Q Indirect
RMB	0	4C	0100 | 1100	0 → $RAM(B)_0$	None	Reset RAM Bit
	1	45	0100 | 0101	0 → $RAM(B)_1$		
	2	42	0100 | 0010	0 → $RAM(B)_2$		
	3	43	0100 | 0011	0 → $RAM(B)_3$		
SMB	0	4D	0100 | 1101	1 → $RAM(B)_0$	None	Set RAM Bit
	1	47	0100 | 0111	1 → $RAM(B)_1$		
	2	46	0100 | 0110	1 → $RAM(B)_2$		
	3	4B	0100 | 1011	1 → $RAM(B)_3$		
STII	y	7–	0111 | y	y → RAM(B) Bd + 1 → Bd	None	Store Memory Immediate and Increment Bd
X	r	–6	00 | r | 0110	RAM(B) ⟷ A Br ⊕ r → Br	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	3,15	23	0010 | 0011	RAM(3,15) ⟷ A	None	Exchange A with RAM (3,15)
		BF	1011 | 1111			

Instruction Set (Continued)

TABLE III. COP410C/411C Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
MEMORY REFERENCE INSTRUCTIONS (Continued)						
XDS	r	−7	00 | r | 0111	RAM(B) ⟷ A Bd − 1 → Bd Br ⊕ r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd Exclusive-OR Br with r
XIS	r	−4	00 | r | 0100	RAM(B) ⟷ A Bd + 1 → Bd Br ⊕ r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd Exclusive-OR Br with r
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	0101 | 0000	A → Bd	None	Copy A to Bd
CBA		4E	0100 | 1110	Bd → A	None	Copy Bd to A
LBI	r,d	–	00 | r | (d - 1) (d = 0,9:15)	r,d → B	Skip until not a LBI	Load B Immediate with r,d
LEI	y	33 6–	0011 | 0011 0010 | y	y → EN	None	Load EN Immediate
TEST INSTRUCTIONS						
SKC		20	0010 | 0000		C = "1"	Skip if C is True
SKE		21	0010 | 0001		A = RAM(B)	Skip if A Equals RAM
SKGZ		33 21	0011 | 0011 0010 | 0001		$G_{3:0} = 0$	Skip if G is Zero (all 4 bits)
SKGBZ		33	0011 | 0011	1st byte		Skip if G Bit is Zero
	0	01	0000 | 0001	2nd byte	$G_0 = 0$	
	1	11	0001 | 0001		$G_1 = 0$	
	2	03	0000 | 0011		$G_2 = 0$	
	3	13	0010 | 0011		$G_3 = 0$	
SKMBZ	0	01	0000 | 0001		$RAM(B)_0 = 0$	Skip if RAM Bit is Zero
	1	11	0001 | 0001		$RAM(B)_1 = 0$	
	2	03	0000 | 0011		$RAM(B)_2 = 0$	
	3	13	0001 | 0011		$RAM(B)_3 = 0$	
INPUT/OUTPUT INSTRUCTIONS						
ING		33 2A	0011 | 0011 0010 | 1010	G → A	None	Input G Ports to A
INL		33 2E	0011 | 0011 0010 | 1110	$L_{7:4}$ → RAM(B) $L_{3:0}$ → A	None	Input L Ports to RAM, A
OBD		33 3E	0011 | 0011 0011 | 1110	Bd → D	None	Output Bd to D Outputs
OMG		33 3A	0011 | 0011 0011 | 1010	RAM(B) → G	None	Output RAM to G Ports
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO

Note 1: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 2: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Description of Selected Instructions

The following information is provided to assist the user in understanding the operation of several unique instructions and to provide notes useful to programmers in writing COP410C/411C programs.

XAS INSTRUCTION

XAS (Exchange A with SIO) exchanges the 4-bit contents of the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. An XAS instruction will also affect the SK output. (See Functional Description, EN Register). If SIO is selected as a shift register, an XAS instruction must be performed once every four instruction cycle times to effect a continuous data stream.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower eight bits of the ROM address register PC with the contents of ROM addressed by the 9-bit word, PC_8, A, M. PC_8 is not affected by this instruction.

Note: JID uses two instruction cycles if executed, one if skipped.

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 9-bit word PC_8, A, M. LQID can be used for table look-up or code conversion such as BCD to 7-segment. The LQID instruction "pushes" the stack (PC + 1 $\rightarrow$ SA $\rightarrow$ SB) and replaces the least significant eight bits of the PC as follows: A $\rightarrow$ $PC_{7:4}$, RAM(B) $\rightarrow$ $PC_{3:0}$, leaving PC_8 unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SB $\rightarrow$ SA $\rightarrow$ PC), restoring the saved value of the PC to continue sequential program execution. Since LQID pushes SA $\rightarrow$ SB, the previous contents of SB are lost.

Note: LQID uses two instruction cycles if executed, one if skipped.

INSTRUCTION SET NOTES

a. The first word of a COP410C/411C program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, one instruction cycle time is devoted to skipping each byte of the skipped instruction. Thus all program paths take the same number of cycle times whether instructions are skipped or executed (except JID and LQID).

c. The ROM is organized into eight pages of 64 words each. The program counter is a 9-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID, or LQID instruction is located in the last word of a page, the instruction operates as if it were in the next page. For example: A JP located in the last word of a page will jump to a location in the next page. Also, a LQID or JID located in the last word in page 3 or 7 will access data in the next group of four pages.

POWER DISSIPATION

The lowest power drain is when the clock is stopped. As the frequency increases so does current. Current is also lower at lower operating voltages. Therefore, to minimize power consumption, the user should run at the lowest speed and voltage that his application will allow. The user should take care that all pins swing to full supply levels to ensure that outputs are not loaded down and that inputs are not at some intermediate level which may draw current. Any input with a slow rise or fall time will draw additional current. A crystal- or resonator-generated clock will draw additional current. An RC oscillator will draw even more current since the input is a slow rising signal.

If using an external squarewave oscillator, the following equation can be used to calculate the COP410C current drain.

$$Ic = Iq + (V \times 20 \times Fi) + (V \times 1280 \times Fl/Dv)$$

where Ic = chip current drain in microamps
Iq = quiescent leakage current (from curve)
Fl = CKI frequency in megahertz
V = chip V_{CC} in volts
Dv = divide by option selected

For example, at 5V V_{CC} and 400 kHz (divide by 4),

$$Ic = 10 + (5 \times 20 \times 0.4) + (5 \times 1280 \times 0.4/4)$$

$$Ic = 10 + 40 + 640 = 690\ \mu A$$

I/O OPTIONS

COP410C/411C outputs have the following optional configurations, illustrated in *Figure 7*:

a. Standard. A CMOS push-pull buffer with an N-channel device to ground in conjunction with a P-channel device to V_{CC}, compatible with CMOS and LSTTL.

b. Low Current. This is the same configuration as (a) above except that the sourcing current is much less.

c. Open Drain. An N-channel device to ground only, allowing external pull-up as required by the user's application.

d. Standard TRI-STATE L Output. A CMOS output buffer similar to (a) which may be disabled by program control.

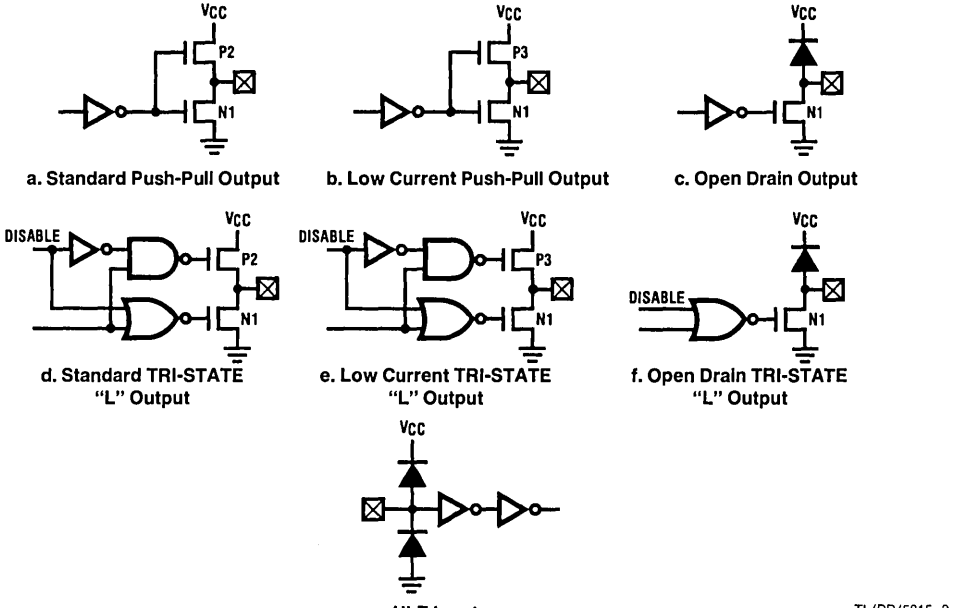
e. Low-Current TRI-STATE L Output. This is the same as (d) above except that the sourcing current is much less.

f. Open-Drain TRI-STATE L Output. This has the N-channel device to ground only.

The SI and $\overline{\text{RESET}}$ inputs are Hi-Z inputs (*Figure 7g*).

When using either the G or L I/O ports as inputs, a pull-up device is necessary. This can be an external device or the following alternative is available: Select the low-current output option. Now, by setting the output registers to a logic "1" level, the P-channel devices will act as the pull-up load. Note that when using the L ports in this fashion, the Q registers must be set to a logic "1" level and the L drivers *must be enabled* by an LEI instruction.

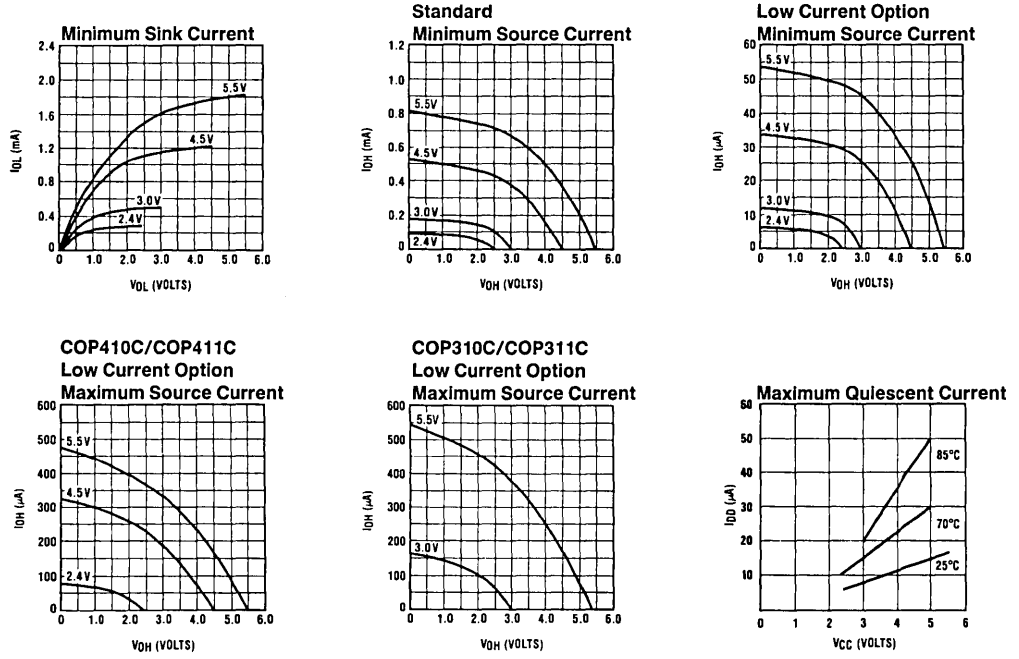
Functional Description (Continued)



TL/DD/5015-9

FIGURE 7. I/O Configurations

Typical Performance Characteristics



TL/DD/5015-10

FIGURE 8

All output drivers uses one or more of three common devices numbered 1 to 3. Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figure 8* for each of these devices to allow the designer to effectively use these I/O configurations.

Option List

The COP410C/411C mask-programmable options are assigned numbers which correspond with the COP410C pins.

The following is a list of COP410C options. When specifying a COP411 chip, options 20, 21, and 22 must be set to 0. The options are programmed at the same time as the ROM pattern to provide the user with the hardware flexibility to interface to various I/O components using little or no external circuitry.

Option 1: 0 = Ground Pin. No options available.

Option 2: CKO I/O Port Determined by Option 3.

Option 3: CKI Input.
- = 0: Crystal-controlled oscillator input (÷ 4).
- = 1: Single-pin RC-controlled oscillator (÷ 4).
- = 2: External oscillator input (÷ 4).
- = 3: Crystal oscillator input (÷ 8).
- = 4: External oscillator input (÷ 8).
- = 5: Crystal oscillator input (÷ 16).
- = 6: External oscillator input (÷ 16).

Option 4: $\overline{\text{RESET}}$ Input = 1: Hi-Z input. No option available.

Option 5: L_7 Driver
- = 0: Standard TRI-STATE push-pull output.
- = 1: Low-current TRI-STATE push-pull output.
- = 2: Open-drain TRI-STATE output.

Option 6: L_6 Driver. (Same as Option 5.)

Option 7: L_5 Driver. (Same as Option 5.)

Option 8: L_4 Driver. (Same as Option 5.)

Option 9: V_{CC} Pin = 0 no option.

Option 10: L_3 Driver. (Same as Option 5.)

Option 11: L_2 Driver. (Same as Option 5.)

Option 12: L_1 Driver. (Same as Option 5.)

Option 13: L_0 Driver. (Same as Option 5.)

Option 14: SI Input.
- No option available.
- = 1: Hi-Z input.

Option 15: SO Output.
- = 0: Standard push-pull output.
- = 1: Low-current push-pull output.
- = 2: Open-drain output.

Option 16: SK Driver. (Same as Option 15.)

Option 17: G_0 I/O Port. (Same as Option 15.)

Option 18: G_1 I/O Port. (Same as Option 15.)

Option 19: G_2 I/O Port. (Same as Option 15.)

Option 20: G_3 I/O Port. (Same as Option 15.)

Option 21: D_3 Output. (Same as Option 15.)

Option 22: D_2 Output. (Same as Option 15.)

Option 23: D_1 Output. (Same as Option 15.)

Option 24: D_0 Output. (Same as Option 15.)

Option 25: Internal Initialization Logic.
- = 0: Normal operation.
- = 1: No internal initialization logic.

Option 26: No option available.

Option 27: COP Bonding
- = 0: COP410C (24-pin device).
- = 1: COP411C (20-pin device). See note.
- = 2: COP410C and COP411C. See note.

Note: If opt. #27 = 1 or 2 then opt #20 must = 0.

Option Table

Please fill out a photocopy of the option table and send it along with your EPROM.

Option Table

Option 1 Value =	0	is: Ground Pin	
Option 2 Value =	0	is: CKO Pin	
Option 3 Value =		is: CKI Input	
Option 4 Value =	1	is: $\overline{\text{RESET}}$ Input	
Option 5 Value =		is: L_7 Driver	
Option 6 Value =		is: L_6 Driver	
Option 7 Value =		is: L_5 Driver	
Option 8 Value =		is: L_4 Driver	
Option 9 Value =	0	is: V_{CC} Pin	
Option 10 Value =		is: L_3 Driver	
Option 11 Value =		is: L_2 Driver	
Option 12 Value =		is: L_1 Driver	
Option 13 Value =		is: L_0 Driver	
Option 14 Value =	1	is: SI Input	
Option 15 Value =		is: SO Output	
Option 16 Value =		is: SK Driver	
Option 17 Value =		is: G_0 I/O Port	
Option 18 Value =		is: G_1 I/O Port	
Option 19 Value =		is: G_2 I/O Port	
Option 20 Value =		is: G_3 I/O Port	
Option 21 Value =		is: D_3 Output	
Option 22 Value =		is: D_2 Output	
Option 23 Value =		is: D_1 Output	
Option 24 Value =		is: D_0 Output	
Option 25 Value =		is: Internal Initialization Logic	
Option 26 Value =	0	is: N/A	
Option 27 Value =		is: COP Bonding	

National Semiconductor Corporation

COP410L/COP411L/COP310L/COP311L Single-Chip N-Channel Microcontrollers

General Description

The COP410L and COP411L Single-Chip N-Channel Microcontrollers are members of the COPS™ family, fabricated using N-channel, silicon gate MOS technology. These Controller Oriented Processors are complete microcomputers containing all system timing, internal logic, ROM, RAM and I/O necessary to implement dedicated control functions in a variety of applications. Features include single supply operation, a variety of output configuration options, with an instruction set, internal architecture and I/O scheme designed to facilitate keyboard input, display output and BCD data manipulation. The COP411L is identical to the COP410L, but with 16 I/O lines instead of 19. They are an appropriate choice for use in numerous human interface control environments. Standard test procedures and reliable high-density fabrication techniques provide the medium to large volume customers with a customized Controller Oriented Processor at a low end-product cost.

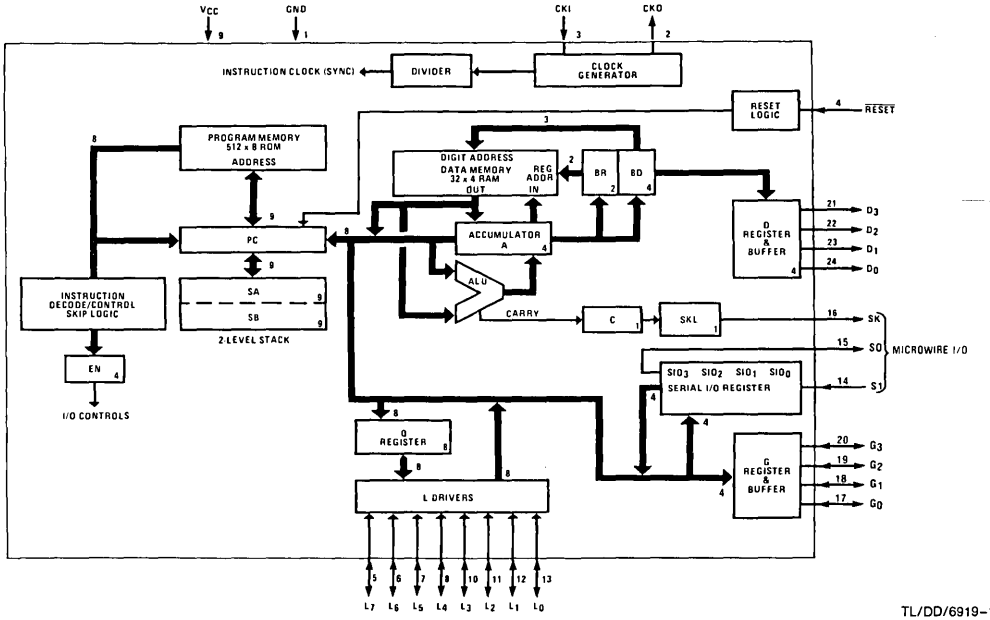
The COP310L and COP311L are exact functional equivalents but extended temperature versions of COP410L and COP411L respectively.

The COP401L should be used for exact emulation.

Features

- Low cost
- Powerful instruction set
- 512 x 8 ROM, 32 x 4 RAM
- 19 I/O lines (COP410L)
- Two-level subroutine stack
- 16 μs instruction time
- Single supply operation (4.5V–6.3V)
- Low current drain (6 mA max)
- Internal binary counter register with MICROWIRE™ serial I/O capability
- General purpose and TRI-STATE® outputs
- LSTTL/CMOS compatible in and out
- Direct drive of LED digit and segment lines
- Software/hardware compatible with other members of COP400 family
- Extended temperature range device — COP310L/COP311L (−40°C to +85°C)
- Wider supply range (4.5V–9.5V) optionally available

Block Diagram



TL/DD/6919–1

FIGURE 1. COP410L

COP410L/COP411L

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin Relative to GND	−0.5V to +10V
Ambient Operating Temperature	0°C to +70°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 seconds)	300°C
Power Dissipation	
COP410L	0.75W at 25°C 0.4W at 70°C
COP411L	0.65W at 25°C 0.3W at 70°C
Total Source Current	120 mA
Total Sink Current	100 mA

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 9.5V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Standard Operating Voltage (V_{CC})	(Note 1)	4.5	6.3	V
Optional Operating Voltage (V_{CC})		4.5	9.5	V
Power Supply Ripple	Peak to Peak		0.5	V
Operating Supply Current	All Inputs and Outputs Open		6	mA
Input Voltage Levels				
CKI Input Levels				
Ceramic Resonator Input (÷8)				
Logic High (V_{IH})	V_{CC} = Max	3.0		V
Logic High (V_{IH})	V_{CC} = 5V ±5%	2.0		V
Logic Low (V_{IL})		−0.3	0.4	V
Schmitt Trigger Input (÷4)				
Logic High (V_{IH})		0.7 V_{CC}		V
Logic Low (V_{IL})		−0.3	0.6	V
$\overline{\text{RESET}}$ Input Levels	(Schmitt Trigger Input)			
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.6	V
SO Input Level (Test Mode)	(Note 2)	2.0	2.5	V
All Other Inputs				
Logic High	V_{CC} = Max	3.0		V
Logic High	With TTL Trip Level Options	2.0		V
Logic Low	Selected, V_{CC} = 5V ±5%	−0.3	0.8	V
Logic High	With High Trip Level Options	3.6		V
Logic Low	Selected	−0.3	1.2	V
Input Capacitance			7	pF
Hi-Z Input Leakage		−1	+1	μA
Output Voltage Levels				
LSTTL Operation	V_{CC} = 5V ±10%			
Logic High (V_{OH})	I_{OH} = −25 μA	2.7		V
Logic Low (V_{OL})	I_{OL} = 0.36 mA		0.4	V
CMOS Operation (Note 3)				
Logic High	I_{OH} = −10 μA	V_{CC} − 1		V
Logic Low	I_{OL} = +10 μA		0.2	V

Note 1: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Note 2: SO output "0" level must be less than 0.8V for normal operation.

Note 3: TRI-STATE® and LED configurations are excluded.

COP410L/COP411L

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 9.5V$ unless otherwise noted (Continued)

Parameter	Conditions	Min	Max	Units
Output Current Levels				
Output Sink Current				
SO and SK Outputs (I_{OL})	$V_{CC} = 9.5V, V_{OL} = 0.4V$	1.8		mA
	$V_{CC} = 6.3V, V_{OL} = 0.4V$	1.2		mA
	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.9		mA
L_0–L_7 Outputs, G_0–G_3 and LSTTL D_0–D_3 Outputs (I_{OL})	$V_{CC} = 9.5V, V_{OL} = 0.4V$	0.4		mA
	$V_{CC} = 6.3V, V_{OL} = 0.4V$	0.4		mA
	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.4		mA
D_0–D_3 Outputs with High Current Options (I_{OL})	$V_{CC} = 9.5V, V_{OL} = 1.0V$	15		mA
	$V_{CC} = 6.3V, V_{OL} = 1.0V$	11		mA
	$V_{CC} = 4.5V, V_{OL} = 1.0V$	7.5		mA
D_0–D_3 Outputs with Very High Current Options (I_{OL})	$V_{CC} = 9.5V, V_{OL} = 1.0V$	30		mA
	$V_{CC} = 6.3V, V_{OL} = 1.0V$	22		mA
	$V_{CC} = 4.5V, V_{OL} = 1.0V$	15		mA
CKI (Single-Pin RC Oscillator)	$V_{CC} = 4.5V, V_{IH} = 3.5V$	2		mA
CKO	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.2		mA
Output Source Current				
Standard Configuration, All Outputs (I_{OH})	$V_{CC} = 9.5V, V_{OH} = 2.0V$	−140	−800	μA
	$V_{CC} = 6.3V, V_{OH} = 2.0V$	−75	−480	μA
	$V_{CC} = 4.5V, V_{OH} = 2.0V$	−30	−250	μA
Push-Pull Configuration SO and SK Outputs (I_{OH})	$V_{CC} = 9.5V, V_{OH} = 4.75V$	−1.4		mA
	$V_{CC} = 6.3V, V_{OH} = 2.4V$	−1.4		mA
	$V_{CC} = 4.5V, V_{OH} = 1.0V$	−1.2		mA
LED Configuration, L_0–L_7 Outputs, Low Current Driver Option (I_{OH})	$V_{CC} = 9.5V, V_{OH} = 2.0V$	−1.5	−18	mA
	$V_{CC} = 6.0V, V_{OH} = 2.0V$	−1.5	−13	mA
LED Configuration, L_0–L_7 Outputs, High Current Driver Option (I_{OH})	$V_{CC} = 9.5V, V_{OH} = 2.0V$	−3.0	−35	mA
	$V_{CC} = 6.0V, V_{OH} = 2.0V$	−3.0	−25	mA
TRI-STATE Configuration, L_0–L_7 Outputs, Low Current Driver Option (I_{OH})	$V_{CC} = 9.5V, V_{OH} = 5.5V$	−0.75		mA
	$V_{CC} = 6.3V, V_{OH} = 3.2V$	−0.8		mA
	$V_{CC} = 4.5V, V_{OH} = 1.5V$	−0.9		mA
TRI-STATE Configuration, L_0–L_7 Outputs, High Current Driver Option (I_{OH})	$V_{CC} = 9.5V, V_{OH} = 5.5V$	−1.5		mA
	$V_{CC} = 6.3V, V_{OH} = 3.2V$	−1.6		mA
	$V_{CC} = 4.5V, V_{OH} = 1.5V$	−1.8		mA
Input Load Source Current	$V_{CC} = 5.0V, V_{IL} = 0V$	−10	−140	μA
CKO Output RAM Power Supply Option Power Requirement	$V_R = 3.3V$		1.5	mA
TRI-STATE Output Leakage Current		−2.5	+2.5	μA
Total Sink Current Allowed				
All Outputs Combined			100	mA
D Port			100	mA
L_7–L_4, G Port			4	mA
L_3–L_0			4	mA
Any Other Pin			2.0	mA
Total Source Current Allowed				
All I/O Combined			120	mA
L_7–L_4			60	mA
L_3–L_0			60	mA
Each L Pin			25	mA
Any Other Pin			1.5	mA

COP310L/COP311L

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin Relative to GND	−0.5V to +10V
Ambient Operating Temperature	−40°C to +85°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 seconds)	300°C
Power Dissipation	
COP310L	0.75W at 25°C 0.25W at 85°C
COP311L	0.65W at 25°C 0.20W at 85°C
Total Source Current	120 mA
Total Sink Current	100 mA

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 7.5V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Standard Operating Voltage (V_{CC})	(Note 1)	4.5	5.5	V
Optional Operating Voltage (V_{CC})		4.5	7.5	V
Power Supply Ripple	Peak to Peak		0.5	V
Operating Supply Current	All Inputs and Outputs Open		8	mA
Input Voltage Levels				
Ceramic Resonator Input (÷ 8) Crystal Input				
Logic High (V_{IH})	V_{CC} = Max	3.0		V
Logic High (V_{IH})	V_{CC} = 5V ±5%	2.2		V
Logic Low (V_{IL})		−0.3	0.3	V
Schmitt Trigger Input (÷ 4)				
Logic High (V_{IH})		0.7 V_{CC}		V
Logic Low (V_{IL})		−0.3	0.4	V
$\overline{\text{RESET}}$ Input Levels	(Schmitt Trigger Input)			
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.4	V
SO Input Level (Test Mode)	(Note 2)	2.2	2.5	V
All Other Inputs				
Logic High	V_{CC} = Max	3.0		V
Logic High	With TTL Trip Level Options	2.2		V
Logic Low	Selected, V_{CC} = 5V ±5%	−0.3	0.6	V
Logic High	With High Trip Level Options	3.6		V
Logic Low	Selected	−0.3	1.2	V
Input Capacitance			7	pF
Hi-Z Input Leakage		−2	+2	µA
Output Voltage Levels				
LSTTL Operation	V_{CC} = 5V ±10%			
Logic High (V_{OH})	I_{OH} = −20 µA	2.7		V
Logic Low (V_{OL})	I_{OL} = 0.36 mA		0.4	V
CMOS Operation (Note 3)				
Logic High	I_{OH} = −10 µA	V_{CC} − 1		V
Logic Low	I_{OL} = +10 µA		0.2	V

Note 1: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Note 2: SO output "0" level must be less than 0.6V for normal operation.

Note 3: TRI-STATE and LED configurations are excluded.

COP310L/COP311L

DC Electrical Characteristics (Continued)

$-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 7.5V$ unless othewise noted

Parameter	Conditions	Min	Max	Units
Output Current Levels				
Output Sink Current				
SO and SK Outputs (I_{OL})	$V_{CC} = 7.5V, V_{OL} = 0.4V$	1.4		mA
	$V_{CC} = 5.5V, V_{OL} = 0.4V$	1.0		mA
	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.8		mA
L_0-L_7 Outputs, G_0-G_3 and LSTTL D_0-D_3 Outputs (I_{OL})	$V_{CC} = 7.5V, V_{OL} = 0.4V$	0.4		mA
	$V_{CC} = 5.5V, V_{OL} = 0.4V$	0.4		mA
	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.4		mA
D_0-D_3 Outputs with High Current Options (I_{OL})	$V_{CC} = 7.5V, V_{OL} = 1.0V$	12		mA
	$V_{CC} = 5.5V, V_{OL} = 1.0V$	9		mA
	$V_{CC} = 4.5V, V_{OL} = 1.0V$	7		mA
D_0-D_3 Outputs with Very High Current Options (I_{OL})	$V_{CC} = 7.5V, V_{OL} = 1.0V$	24		mA
	$V_{CC} = 5.5V, V_{OL} = 1.0V$	18		mA
	$V_{CC} = 4.5V, V_{OL} = 1.0V$	14		mA
CKI (Single-Pin RC Oscillator)	$V_{CC} = 4.5V, V_{IH} = 3.5V$	1.5		mA
CKO	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.2		mA
Output Source Current				
Standard Configuration, All Outputs (I_{OH})	$V_{CC} = 7.5V, V_{OH} = 2.0V$	−100	−900	μA
	$V_{CC} = 5.5V, V_{OH} = 2.0V$	−55	−600	μA
	$V_{CC} = 4.5V, V_{OH} = 2.0V$	−28	−350	μA
Push-Pull Configuration SO and SK Outputs (I_{OH})	$V_{CC} = 7.5V, V_{OH} = 3.75V$	−0.85		mA
	$V_{CC} = 5.5V, V_{OH} = 2.0V$	−1.1		mA
	$V_{CC} = 4.5V, V_{OH} = 1.0V$	−1.2		mA
LED Configuration, L_0-L_7 Outputs, Low Current Driver Option (I_{OH})	$V_{CC} = 7.5V, V_{OH} = 2.0V$	−1.4	−27	mA
	$V_{CC} = 5.5V, V_{OH} = 2.0V$	−0.7	−15	μA
LED Configuration, L_0-L_7 Outputs, High Current Driver Option (I_{OH})	$V_{CC} = 7.5V, V_{OH} = 2.0V$	−2.7	−54	mA
	$V_{CC} = 5.5V, V_{OH} = 2.0V$	−1.4	−30	μA
TRI-STATE Configuration, L_0-L_7 Outputs, Low Current Driver Option (I_{OH})	$V_{CC} = 7.5V, V_{OH} = 4.0V$	−0.7		mA
	$V_{CC} = 5.5V, V_{OH} = 2.7V$	−0.6		mA
	$V_{CC} = 4.5V, V_{OH} = 1.5V$	−0.9		mA
TRI-STATE Configuration, L_0-L_7 Outputs, High Current Driver Option (I_{OH})	$V_{CC} = 7.5V, V_{OH} = 4.0V$	−1.4		mA
	$V_{CC} = 5.5V, V_{OH} = 2.7V$	−1.2		mA
	$V_{CC} = 4.5V, V_{OH} = 1.5V$	−1.8		mA
Input Load Source Current	$V_{CC} = 5.0V, V_{IL} = 0V$	−10	−200	μA
CKO Output RAM Power Supply Option Power Requirement	$V_R = 3.3V$		2.0	mA
TRI-STATE Output Leakage Current		−5	+5	μA
Total Sink Current Allowed				
All Outputs Combined			100	mA
D Port			100	mA
L_7-L_4, G Port			4	mA
L_3-L_0			4	mA
Any Other Pins			1.5	mA
Total Source Current Allowed				
All I/O Combined			120	mA
L_7-L_4			60	mA
L_3-L_0			60	mA
Each L Pin			25	mA
Any Other Pins			1.5	mA

AC Electrical Characteristics

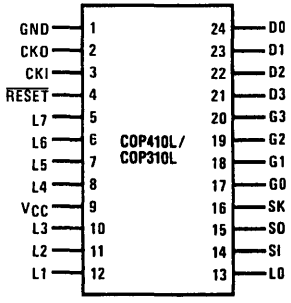
COP410L/411L: $0°C \le T_A \le 70°C$, $4.5V \le V_{CC} \le 9.5V$ unless otherwise noted
COP310L/311L: $-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 7.5V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time — t_C		16	40	μs
CKI				
Input Frequency — f_I	÷8 Mode	0.2	0.5	MHz
	÷4 Mode	0.1	0.25	MHz
Duty Cycle		30	60	%
Rise Time	f_I = 0.5 MHz		500	ns
Fall Time			200	ns
CKI Using RC (÷4) (Note 1)	R = 56 kΩ ±5% C = 100 pF ±10%			
Instruction Cycle Time		16	28	μs
CKO as SYNC Input				
t_{SYNC}		400		ns
INPUTS				
G_3–G_0, L_7–L_0				
t_{SETUP}		8.0		μs
t_{HOLD}		1.3		μs
SI				
t_{SETUP}		2.0		μs
t_{HOLD}		1.0		μs
OUTPUT PROPAGATION DELAY	Test Condition: C_L = 50 pF, R_L = 20 kΩ, V_{OUT} = 1.5V			
SO, SK Outputs				
t_{pd1}, t_{pd0}			4.0	μs
All Other Outputs				
t_{pd1}, t_{pd0}			5.6	μs

Note 1: Variation due to the device included.

Connection Diagrams

Dual-In-Line Package

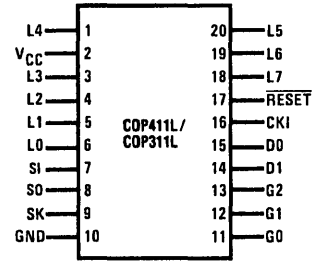


TL/DD/6919-2

Top View

Order Number COP310L-XXX/M, COP310L-XXX/N
COP410L-XXX/M, COP410L-XXX/N
See NS Package Number N24A or M24B

Dual-In-Line Package



TL/DD/6919-3

Top View

Order Number COP311L-XXX/M, COP311L-XXX/N
COP411L-XXX/M, COP411L-XXX/N
See NS Package Number N20A or M20B

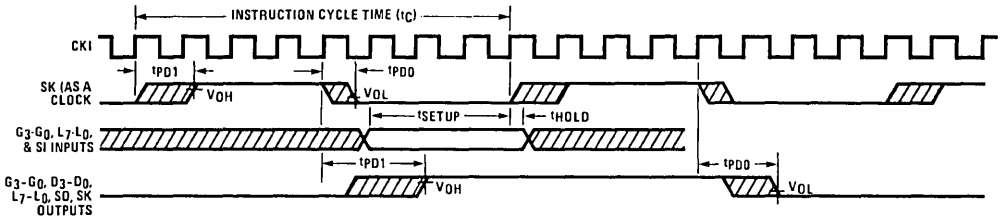
FIGURE 2

Pin Descriptions

Pin	Description
L_7–L_0	8 bidirectional I/O ports with TRI-STATE
G_3–G_0	4 bidirectional I/O ports (G_2–G_0 for COP411L)
D_3–D_0	4 general purpose outputs (D_1–D_0 for COP411L)
SI	Serial input (or counter input)
SO	Serial output (or general purpose output)
SK	Logic-controlled clock (or general purpose output)

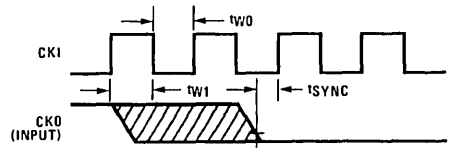
Pin	Description
CKI	System oscillator input
CKO	System oscillator output (or RAM power supply or SYNC input) (COP410L only)
RESET	System reset input
V_{CC}	Power supply
GND	Ground

Timing Diagrams



TL/DD/6919-4

FIGURE 3. Input/Output Timing Diagrams (Ceramic Resonator Divide-by-8 Mode)



TL/DD/6919-5

FIGURE 3a. Synchronization Timing

Functional Description

A block diagram of the COP410L is given in *Figure 1*. Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1" (greater than 2V). When a bit is reset, it is a logic "0" (less than 0.8V).

All functional references to the COP410L/COP411L also apply to the COP310L/COP311L.

PROGRAM MEMORY

Program Memory consists of a 512-byte ROM. As can be seen by an examination of the COP410L/411L instruction set, these words may be program instructions, program data or ROM addressing data. Because of the special characteristics associated with the JP, JSRP, JID and LQID instructions, ROM must often be thought of as being organized into 8 pages of 64 words each.

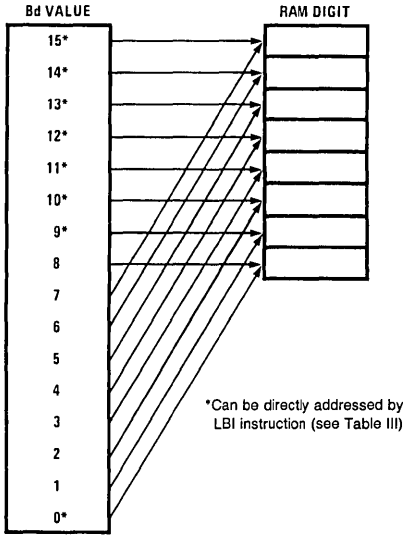
ROM addressing is accomplished by a 9-bit PC register. Its binary value selects one of the 512 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 9-bit binary count value. Two levels of subroutine nesting are implemented by the 9-bit subroutine save registers, SA and SB, providing a last-in, first-out (LIFO) hardware subroutine stack.

ROM instruction words are fetched, decoded and executed by the Instruction Decode, Control and Skip Logic circuitry.

DATA MEMORY

Data memory consists of a 128-bit RAM, organized as 4 data registers of 8 4-bit digits. RAM addressing is implemented by a 6-bit B register whose upper 2 bits (Br) select 1 of 4 data registers and lower 3 bits of the 4-bit Bd select 1 of 8 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) is usually loaded into or from, or exchanged with, the A register (accumulator), it may also be loaded into the Q latches or loaded from the L ports. RAM addressing may also be performed directly by the XAD 3,15 instruction. The Bd register also serves as a source register for 4-bit data sent directly to the D outputs.

The most significant bit of Bd is not used to select a RAM digit. Hence each physical digit of RAM may be selected by two different values of Bd as shown in *Figure 4* below. The skip condition for XIS and XDS instructions will be true if Bd changes between 0 and 15, but NOT between 7 and 8 (see Table III).



TL/DD/6919-6

FIGURE 4. RAM Digit Address to Physical RAM Digit Mapping

Functional Description (Continued)

INTERNAL LOGIC

The 4-bit A register (accumulator) is the source and destination register for most I/O, arithmetic, logic and data memory access operations. It can also be used to load the Bd portion of the B register, to load 4 bits of the 8-bit Q latch data, to input 4 bits of the 8-bit L I/O port data and to perform data exchanges with the SIO register.

A 4-bit adder performs the arithmetic and logic functions of the COP410L/411L, storing its results in A. It also outputs a carry bit to the 1-bit C register, most often employed to indicate arithmetic overflow. The C register, in conjunction with the XAS instruction and the EN register, also serves to control the SK output. C can be outputted directly to SK or can enable SK to be a sync clock each instruction cycle time. (See XAS instruction and EN register description, below.)

The G register contents are outputs to 4 general-purpose bidirectional I/O ports.

The Q register is an internal, latched, 8-bit register, used to hold data loaded from M and A, as well as 8-bit data from ROM. Its contents are output to the L I/O ports when the L drivers are enabled under program control. (See LEI instruction.)

The 8 L drivers, when enabled, output the contents of latched Q data to the L I/O ports. Also, the contents of L may be read directly into A and M. L I/O ports can be directly connected to the segments of a multiplexed LED display (using the LED Direct Drive output configuration option) with Q data being outputted to the Sa–Sg and decimal point segments of the display.

The SIO register functions as a 4-bit serial-in serial-out shift register or as a binary counter depending on the contents of the EN register. (See EN register description, below.) Its contents can be exchanged with A, allowing it to input or output a continuous serial data stream. SIO may also be used to provide additional parallel I/O by connecting SO to external serial-in/parallel-out shift registers.

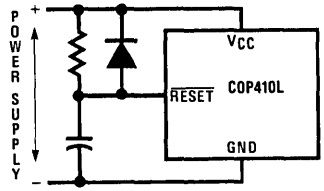
The XAS instruction copies C into the SKL Latch. In the counter mode, SK is the output of SKL in the shift register mode, SK outputs SKL ANDed with internal instruction cycle clock.

The EN register is an internal 4-bit register loaded under program control by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register (EN_3–EN_0).

1. The least significant bit of the enable register, EN_0, selects the SIO register as either a 4-bit shift register or a 4-bit binary counter. With EN_0 set, SIO is an asynchronous binary counter, *decrementing* its value by one upon each low-going pulse ("1" to "0") occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output is equal to the value of EN_3. With EN_0 reset, SIO is a serial shift register shifting left each instruction cycle time. The data present at SI goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. (See 4 below.) The SK output becomes a logic-controlled clock.
2. EN_1 is not used. It has no effect on COP410L/COP411L operation.
3. With EN_2 set, the L drivers are enabled to output the data in Q to the L I/O ports. Resetting EN_2 disables the L drivers, placing the L I/O ports in a high-impedance input state.
4. EN_3, in conjunction with EN_0, affects the SO output. With EN_0 set (binary counter option selected) SO will output the value loaded into EN_3. With EN_0 reset (serial shift register option selected), setting EN_3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN_3 with the serial shift register option selected disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains reset to "0." Table I provides a summary of the modes associated with EN_3 and EN_0.

INITIALIZATION

The Reset Logic will initialize (clear) the device upon power-up if the power supply rise time is less than 1 ms and greater than 1 μs. If the power supply rise time is greater than 1 ms, the user must provide an external RC network and diode to the $\overline{\text{RESET}}$ pin as shown below *(Figure 5)*. The $\overline{\text{RESET}}$ pin is configured as a Schmitt trigger input. If not used it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the $\overline{\text{RESET}}$ input, provided it stays low for at least three instruction cycle times.



RC ≥ 5 × Power Supply Rise Time TL/DD/6919–7

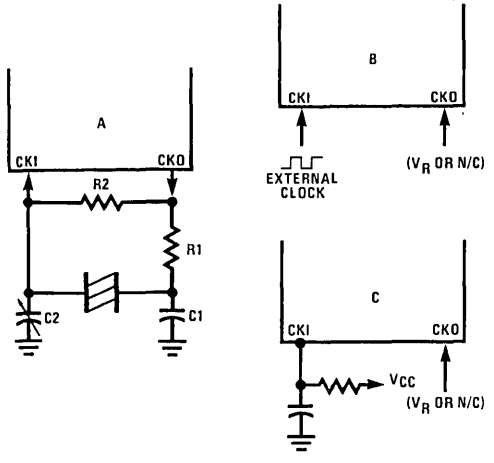
FIGURE 5. Power-Up Clear Circuit

TABLE I. Enable Register Modes—Bits EN_3 and EN_0

EN_3	EN_0	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL = 1, SK = Clock If SKL = 0, SK = 0
1	0	Shift Register	Input to Shift Register	Serial Out	If SKL = 1, SK = Clock If SKL = 0, SK = 0
0	1	Binary Counter	Input to Binary Counter	0	If SKL = 1, SK = 1 If SKL = 0, SK = 0
1	1	Binary Counter	Input to Binary Counter	1	If SKL = 1, SK = 1 If SKL = 0, SK = 0

Functional Description (Continued)

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN, and G registers are cleared. The SK output is enabled as a SYNC output, providing a pulse each instruction cycle time. *Data Memory (RAM) is not cleared upon initialization.* The first instruction at address 0 must be a CLRA.



TL/DD/6919–8

Ceramic Resonator Oscillator

Resonator Value	Components Values			
	R1 (Ω)	R2 (Ω)	C1 (pF)	C2 (pF)
455 kHz	4.7k	1M	220	220

RC Controlled Oscillator

R (kΩ)	C (pF)	Instruction Cycle Time in μs
51	100	19 ±15%
82	56	10 ±13%

Note: 200 kΩ ≥ R ≥ 25 kΩ. 360 pF ≥ C ≥ 50 pF. Does not include tolerances.

FIGURE 6. COP410L/411L Oscillator

OSCILLATOR

There are three basic clock oscillator configurations available as shown by *Figure 6*.

a. Resonator Controlled Oscillator. CKI and CKO are connected to an external ceramic resonator. The instruction cycle frequency equals the resonator frequency divided by 8. This is not available in the COP411L.

b. External Oscillator. CKI is an external clock input signal. The external frequency is divided by 8 to give the instruction frequency time. CKO is now available to be used as the RAM power supply (V_R), or no connection.

Note: No CKO on COP411L.

c. RC Controlled Oscillator. CKI is configured as a single pin RC controlled Schmitt trigger oscillator. The instruction cycle equals the oscillation frequency divided by 4. CKO is available as the RAM power supply (V_R) or no connection.

CKO PIN OPTIONS

In a resonator controlled oscillator system, CKO is used as an output to the resonator network. As an option, CKO can be a RAM power supply pin (V_R), allowing its connection to a standby/backup power supply to maintain the integrity of RAM data with minimum power drain when the main supply is inoperative or shut down to conserve power. Using no connection option is appropriate in applications where the COP410L system timing configuration does not require use of the CKO pin.

RAM KEEP-ALIVE OPTION

Selecting CKO as the RAM power supply (V_R) allows the user to shut off the chip power supply (V_{CC}) and maintain data in the RAM. To insure that RAM data integrity is maintained, the following conditions must be met:

1. $\overline{\text{RESET}}$ must go low before V_{CC} goes below spec during power-off; V_{CC} must be within spec before $\overline{\text{RESET}}$ goes high on power-up.
2. During normal operation, V_R must be within the operating range of the chip with $(V_{CC} - 1) \leq V_R \leq V_{CC}$.
3. V_R must be ≥ 3.3V with V_{CC} off.

I/O OPTIONS

COP410L/411L inputs and outputs have the following optional configurations, illustrated in *Figure 7*:

a. Standard—an enhancement-mode device to ground in conjunction with a depletion-mode device to V_{CC}, compatible with LSTTL and CMOS input requirements. Available on SO, SK, and all D and G outputs.

b. Open-Drain—an enhancement-mode device to ground only, allowing external pull-up as required by the user's application. Available on SO, SK, and all D and G outputs.

c. Push-Pull—an enhancement-mode device to ground in conjunction with a depletion-mode device paralleled by an enhancement-mode device to V_{CC}. This configuration has been provided to allow for fast rise and fall times when driving capacitive loads. Available on SO and SK outputs only.

d. Standard L—same as **a.**, but may be disabled. Available on L outputs only.

e. Open Drain L—same as **b.**, but may be disabled. Available on L outputs only.

f. LED Direct Drive—an enhancement mode device to ground and to V_{CC}, meeting the typical current sourcing requirements of the segments of an LED display. The sourcing device is clamped to limit current flow. These devices may be turned off under program control (see Functional Description, EN Register), placing the outputs in a high-impedance state to provide required LED segment blanking for a multiplexed display. Available on L outputs only.

g. TRI-STATE Push-Pull—an enhancement-mode device to ground and V_{CC}. These outputs are TRI-STATE outputs, allowing for connection of these outputs to a data bus shared by other bus drivers. Available on L outputs only.

Functional Description (Continued)

h. An on-chip depletion load device to V_{CC}.

i. A Hi-Z input which must be driven to a "1" or "0" by external components.

The above input and output configurations share common enhancement-mode and depletion-mode devices. Specifically, all configurations use one or more of six devices (numbered 1–6, respectively). Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figure 8* for each of these devices to allow the designer to effectively use these I/O configurations in designing a COP410L/411L system.

The SO, SK outputs can be configured as shown in **a.**, **b.**, or **c.** The D and G outputs can be configured as shown in **a.** or **b.** Note that when inputting data to the G ports, the G outputs should be set to "1". The L outputs can be configured as in **d.**, **e.**, **f.**, or **g.**

An important point to remember if using configuration **d.** or **f.** with the L drivers is that even when the L drivers are disabled, the depletion load device will source a small amount of current. (See *Figure 8*, device 2.) However, when the L port is used as input, the disabled depletion device CANNOT be relied on to source sufficient current to pull an input to a logic "1".

COP411L

If the COP410L is bonded as a 20-pin device, it becomes the COP411L, illustrated in *Figure 2*, COP410L/411L Connection Diagrams. Note that the COP411L does not contain D2, D3, G3, or CKO. Use of this option of course precludes use of D2, D3, G3, and CKO options. All other options are available for the COP411L.

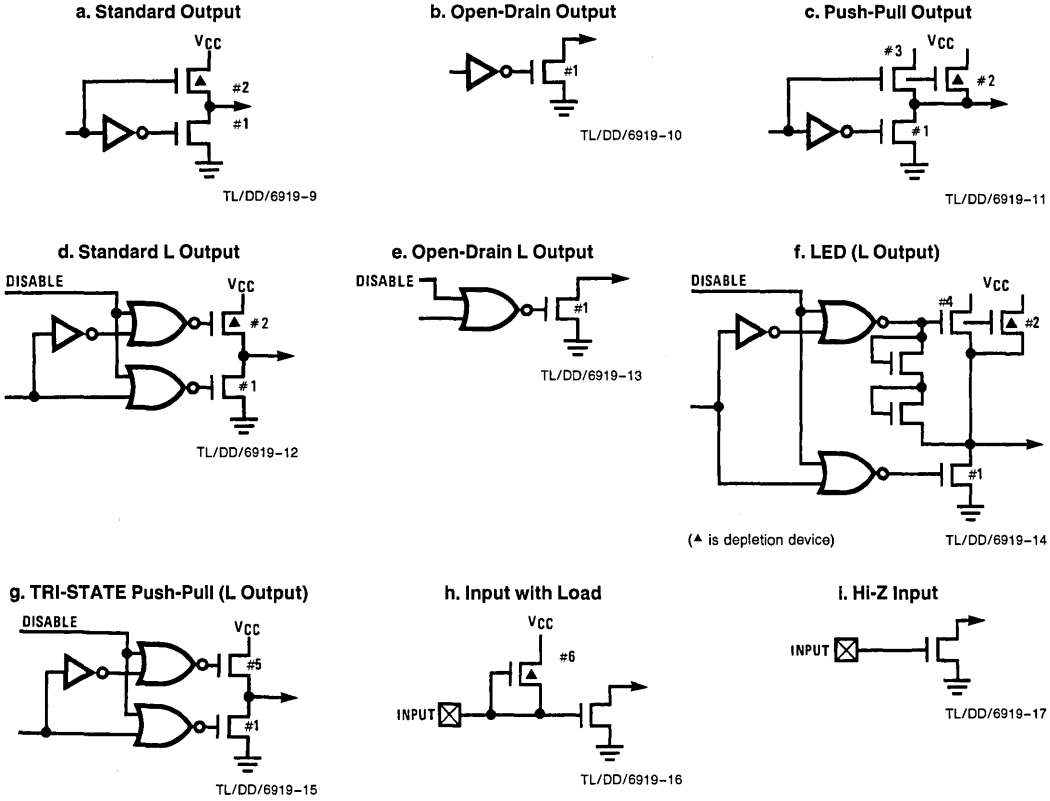
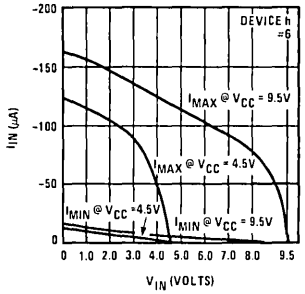


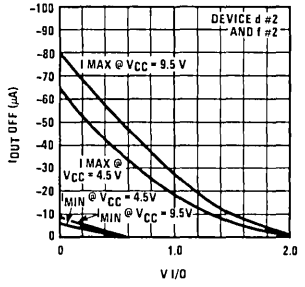
FIGURE 7. Input and Output Configurations

Typical Performance Characteristics

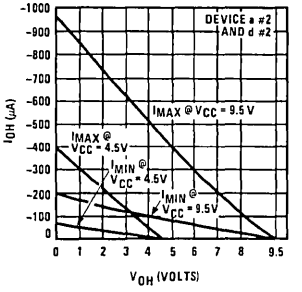
Input Current $\overline{RESET}$, SI



Input Current for L0 through L7 when Output Programmed Off by Software

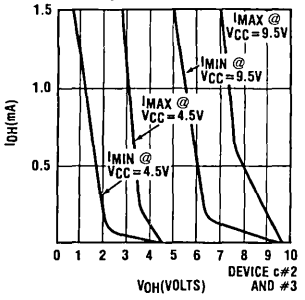


Source Current for Standard Output Configuration

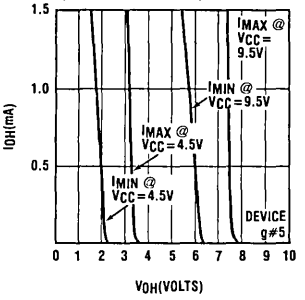


TL/DD/6919–18

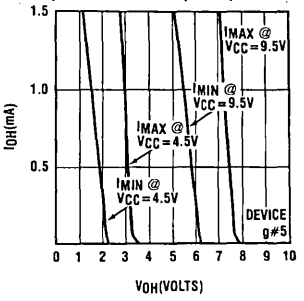
Source Current for SO and SK in Push-Pull Configuration



Source Current for L0 through L7 in TRI-STATE Configuration (High Current Option)

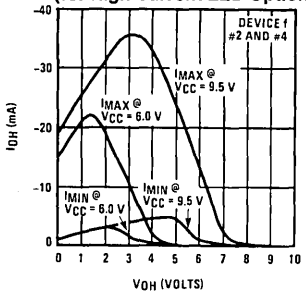


Source Current for L0 through L7 in TRI-STATE Configuration (Low Current Option)

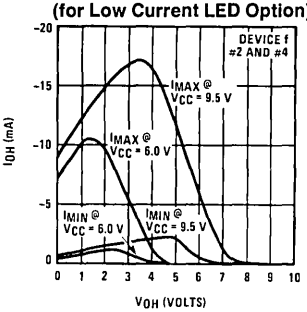


TL/DD/6919–19

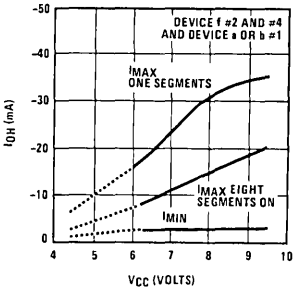
LED Output Source Current (for High Current LED Option)



LED Output Source Current (for Low Current LED Option)



LED Output Direct Segment and Direct Drive High Current Options on L0–L7 Very High Current Options on D0–D3

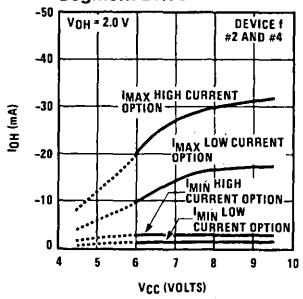


TL/DD/6919–20

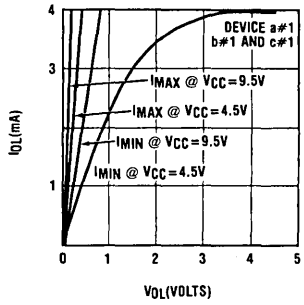
FIGURE 8a. COP410L/COP411L I/O DC Current Characteristics

Typical Performance Characteristics (Continued)

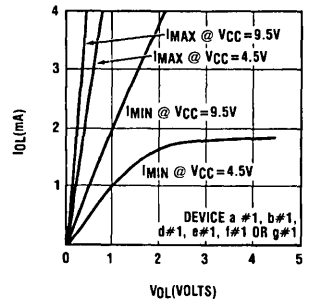
LED Output Direct Segment Drive



Output Sink Current for SO and SK

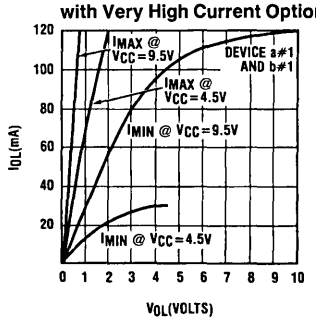


Output Sink Current for L0–L7 and Standard Drive Option for D0–D3 and G0–G3

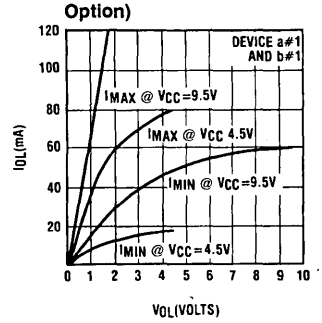


TL/DD/6919–21

Output Sink Current for D0–D3 with Very High Current Option



Output Sink Current for D0–D3 (for High Current Option)

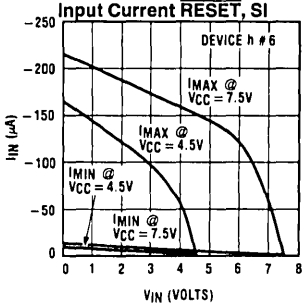


TL/DD/6919–22

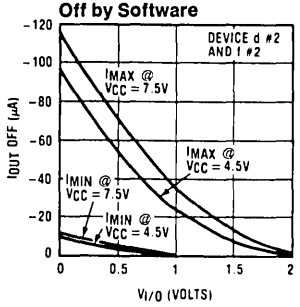
FIGURE 8a. COP410L/COP411L I/O DC Current Characteristics (Continued)

Typical Performance Characteristics (Continued)

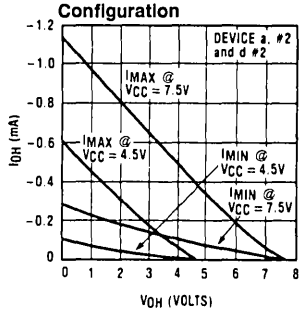
Input Current RESET, SI



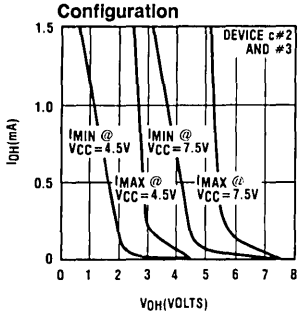
Input Current for L0–L7 when Output Programmed Off by Software



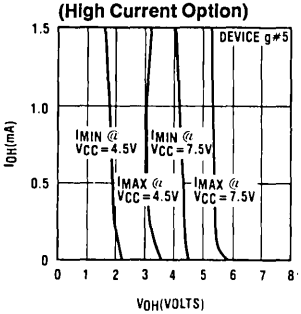
Source Current for Standard Output Configuration



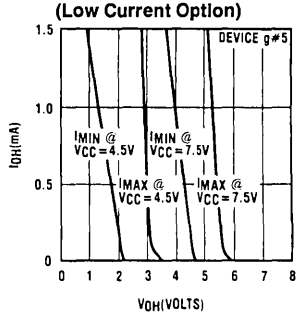
Source Current for SO and SK in Push-Pull Configuration



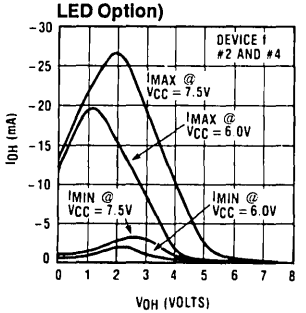
Source Current for L0–L7 in TRI-STATE Configuration (High Current Option)



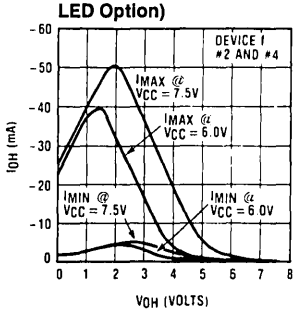
Source Current for L0–L7 in TRI-STATE Configuration (Low Current Option)



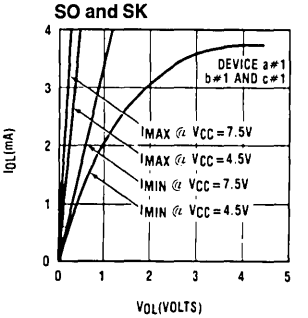
LED Output Source Current (for Low Current LED Option)



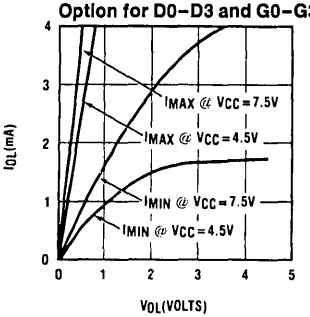
LED Output Source Current (for High Current LED Option)



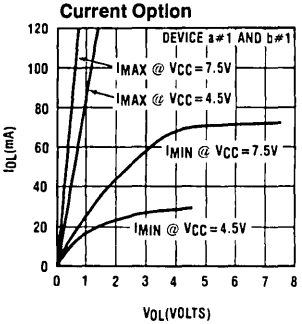
Output Sink Current for SO and SK



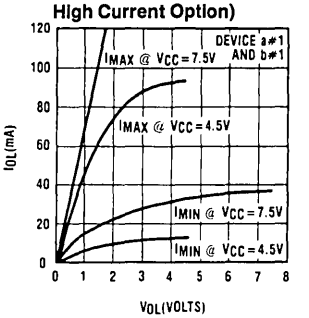
Output Sink Current for L0–L7 and Standard Drive Option for D0–D3 and G0–G3



Output Sink Current for D0–D3 with Very High Current Option



Output Sink Current for D0–D3 (for High Current Option)



TL/DD/6919–23

FIGURE 8b. COP310L/COP311L Input/Output Characteristics

COP410L/411L Instruction Set

Table II is a symbol table providing internal architecture, instruction operand and operational symbols used in the instruction set table.

Table III provides the mnemonic, operand, machine code, data flow, skip conditions and description associated with each instruction in the COP410L/411L instruction set.

TABLE II. COP410L/411L Instruction Set Table Symbols

Symbol	Definition
INTERNAL ARCHITECTURE SYMBOLS	
A	4-bit Accumulator
B	6-bit RAM Address Register
Br	Upper 2 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry Register
D	4-bit Data Output Port
EN	4-bit Enable Register
G	4-bit Register to latch data for G I/O Port
L	8-bit TRI-STATE I/O Port
M	4-bit contents of RAM Memory pointed to by B Register
PC	9-bit ROM Address Register (program counter)
Q	8-bit Register to latch data for L I/O Port
SA	9-bit Subroutine Save Register A
SB	9-bit Subroutine Save Register B
SIO	4-bit Shift Register and Counter
SK	Logic-Controlled Clock Output

Symbol	Definition
INSTRUCTION OPERAND SYMBOLS	
d	4-bit Operand Field, 0–15 binary (RAM Digit Select)
r	2-bit Operand Field, 0–3 binary (RAM Register Select)
a	9-bit Operand Field, 0–511 binary (ROM Address)
y	4-bit Operand Field, 0–15 binary (Immediate Data)
RAM(s)	Contents of RAM location addressed by s
ROM(t)	Contents of ROM location addressed by t
OPERATIONAL SYMBOLS	
+	Plus
−	Minus
→	Replaces
↔	Is exchanged with
=	Is equal to
$\overline{A}$	The one's complement of A
⊕	Exclusive-OR
:	Range of values

TABLE III. COP410L/411L Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	|0011|0000|	A + C + RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	|0011|0001|	A + RAM(B) → A	None	Add RAM to A
AISC	y	5–	|0101| y |	A + y → A	Carry	Add Immediate, Skip on Carry (y ≠ 0)
CLRA		00	|0000|0000|	0 → A	None	Clear A
COMP		40	|0100|0000|	$\overline{A}$ → A	None	One's complement of A to A
NOP		44	|0100|0100|	None	None	No Operation
RC		32	|0011|0010|	"0" → C	None	Reset C
SC		22	|0010|0010|	"1" → C	None	Set C
XOR		02	|0000|0010|	A ⊕ RAM(B) → A	None	Exclusive-OR RAM with A

Instruction Set (Continued)

TABLE III. COP410L/411L Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
TRANSFER OF CONTROL INSTRUCTIONS						
JID		FF	1111 1111	ROM (PC_8,A,M) → $PC_{7:0}$	None	Jump Indirect (Note 2)
JMP	a	6– – –	0110 000 a_8 $a_{7:0}$	a → PC	None	Jump
JP	a	– –	1 $a_{6:0}$ (pages 2,3 only) or	a → $PC_{6:0}$	None	Jump within Page (Note 3)
		– –	11 $a_{5:0}$ (all other pages)	a → $PC_{5:0}$		
JSRP	a	– –	10 $a_{5:0}$	PC + 1 → SA → SB 010 → $PC_{8:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 4)
JSR	a	6– – –	0110 100 a_8 $a_{7:0}$	PC + 1 → SA → SB a → PC	None	Jump to Subroutine
RET		48	0100 1000	SB → SA → PC	None	Return from Subroutine
RETSK		49	0100 1001	SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
MEMORY REFERENCE INSTRUCTIONS						
CAMQ		33 3C	0011 0011 0011 1100	A → $Q_{7:4}$ RAM(B) → $Q_{3:0}$	None	Copy A, RAM to Q
LD	r	–5	00 r 0101	RAM(B) → A Br ⊕ r → Br	None	Load RAM into A, Exclusive-OR Br with r
LQID		BF	1011 1111	ROM(PC_8,A,M) → Q SA → SB	None	Load Q Indirect (Note 2)
RMB	0 1 2 3	4C 45 42 43	0100 1100 0100 0101 0100 0010 0100 0011	0 → $RAM(B)_0$ 0 → $RAM(B)_1$ 0 → $RAM(B)_2$ 0 → $RAM(B)_3$	None	Reset RAM Bit
SMB	0 1 2 3	4D 47 46 4B	0100 1101 0100 0111 0100 0110 0100 1011	1 → $RAM(B)_0$ 1 → $RAM(B)_1$ 1 → $RAM(B)_2$ 1 → $RAM(B)_3$	None	Set RAM Bit
STII	y	7–	0111 y	y → RAM(B) Bd + 1 → Bd	None	Store Memory Immediate and Increment Bd
X	r	–6	00 r 0110	RAM(B) ⟷ A Br ⊕ r → Br	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	3,15	23 BF	0010 0011 1011 1111	RAM(3,15) ⟷ A	None	Exchange A with RAM (3,15)
XDS	r	–7	00 r 0111	RAM(B) ⟷ A Bd – 1 → Bd Br ⊕ r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd, Exclusive-OR Br with r
XIS	r	–4	00 r 0100	RAM(B) ⟷ A Bd + 1 → Bd Br ⊕ r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd Exclusive-OR Br with r

Instruction Set (Continued)

TABLE III. COP410L/411L Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	0101 | 0000	A → Bd	None	Copy A to Bd
CBA		4E	0100 | 1110	Bd → A	None	Copy Bd to A
LBI	r,d	– –	00 | r | (d − 1) (d = 0,9:15)	r,d → B	Skip until not a LBI	Load B Immediate with r,d (Note 5)
LEI	y	33 6–	0011 | 0011 0110 | y	y → EN	None	Load EN Immediate (Note 6)
TEST INSTRUCTIONS						
SKC		20	0010 | 0000		C = "1"	Skip if C is True
SKE		21	0010 | 0001		A = RAM(B)	Skip if A Equals RAM
SKGZ		33 21	0011 | 0011 0010 | 0001		$G_{3:0} = 0$	Skip if G is Zero (all 4 bits)
SKGBZ		33	0011 | 0011	1st byte		Skip if G Bit is Zero
	0	01	0000 | 0001	2nd byte	$G_0 = 0$	
	1	11	0001 | 0001	2nd byte	$G_1 = 0$	
	2	03	0000 | 0011	2nd byte	$G_2 = 0$	
	3	13	0001 | 0011	2nd byte	$G_3 = 0$	
SKMBZ	0	01	0000 | 0001		$RAM(B)_0 = 0$	Skip if RAM Bit is Zero
	1	11	0001 | 0001		$RAM(B)_1 = 0$	
	2	03	0000 | 0011		$RAM(B)_2 = 0$	
	3	13	0001 | 0011		$RAM(B)_3 = 0$	
INPUT/OUTPUT INSTRUCTIONS						
ING		33 2A	0011 | 0011 0010 | 1010	G → A	None	Input G Ports to A
INL		33 2E	0011 | 0011 0010 | 1110	$L_{7:4}$ → RAM(B) $L_{3:0}$ → A	None	Input L Ports to RAM, A
OBD		33 3E	0011 | 0011 0011 | 1110	Bd → D	None	Output Bd to D Outputs
OMG		33 3A	0011 | 0011 0011 | 1010	RAM(B) → G	None	Output RAM to G Ports
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO (Note 2)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicitly defined (e.g., Br and Bd are explicitly defined). Bits are numbered 0 to N where 0 signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit A register.

Note 2: For additional information on the operation of the XAS, JID, and LQID instructions, see below.

Note 3: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 4: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 5: The machine code for the lower 4 bits of the LBI instruction equals the binary value of the "d" data *minus 1*, e.g., to load the lower four bits of B (Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Note 6: Machine code for operand field y for LEI instruction should equal the binary value to be latched into EN, where a "1" or "0" in each bit of EN corresponds with the selection or deselection of a particular function associated with each bit. (See Functional Description, EN Register.)

Description of Selected Instructions

The following information is provided to assist the user in understanding the operation of several unique instructions and to provide notes useful to programmers in writing COP410L/411L programs.

XAS INSTRUCTION

XAS (Exchange A with SIO) exchanges the 4-bit contents of the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. An XAS instruction will also affect the SK output. (See Functional Description, EN Register, above.) If SIO is selected as a shift register, an XAS instruction must be performed once every 4 instruction cycles to effect a continuous data stream.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower 8 bits of the ROM address register PC with the *contents* of ROM addressed by the 9-bit word, PC_8, A, M. PC_8 is not affected by this instruction.

Note that JID requires 2 instruction cycles to execute.

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 9-bit word PC_8, A, M. LQID can be used for table lookup or code conversion such as BCD to seven-segment. The LQID instruction "pushes" the stack (PC + 1 → SA → SB) and replaces the least significant 8 bits of PC as follows: A → $PC_{7:4}$, RAM(B) → $PC_{3:0}$, leaving PC_8 unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SB → SA → PC), restoring the saved value of PC to continue sequential program execution. Since LQID pushes SA → SB, the previous contents of SB are lost. Also, when LQID pops the stack, the previously pushed contents of SA are left in SB. The net result is that the contents of SA are placed in SB (SA → SB). Note that LQID takes two instruction cycle times to execute.

INSTRUCTION SET NOTES

a. The first word of a COP410L/411L program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, one instruction cycle time is devoted to skipping each byte of the skipped instruction. Thus all program paths except JID and LQID take the same number of cycle times whether instructions are skipped or executed. JID and LQID instructions take 2 cycles if executed and 1 cycle if skipped.

c. The ROM is organized into 8 pages of 64 words each. The Program Counter is a 9-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID or LQID instruction is located in the last word of a page, the instruction operates as if it were in the next page. For example: a JP located in the last word of a page will jump to a location in the next page. Also, a LQID or JID located in the last word of page 3 or 7 will access data in the next group of 4 pages.

Option List

The COP410L/411L mask-programmable options are assigned numbers which correspond with the COP410L pins.

The following is a list of COP410L options. When specifying a COP411L chip, Option 2 must be set to 3, Options 20, 21, and 22 to 0. The options are programmed at the same time as the ROM pattern to provide the user with the hardware flexibility to interface to various I/O components using little or no external circuitry.

Option 1 = 0: Ground Pin — no options available

Option 2: CKO Output (no option available for COP411L)
- = 0: Clock output to ceramic resonator
- = 1: Pin is RAM power supply (V_R) input
- = 3: No connection

Option 3: CKI Input
- = 0: Oscillator input divided by 8 (500 kHz max)
- = 1: Single-pin RC controlled oscillator divided by 4
- = 2: External Schmitt trigger level clock divided by 4

Option 4: $\overline{\text{RESET}}$ Input
- = 0: Load device to V_{CC}
- = 1: Hi-Z input

Option 5: L_7 Driver
- = 0: Standard output
- = 1: Open-drain output
- = 2: High current LED direct segment drive output
- = 3: High current TRI-STATE push-pull output
- = 4: Low-current LED direct segment drive output
- = 5: Low-current TRI-STATE push-pull output

Option 6: L_6 Driver
same as Option 5

Option 7: L_5 Driver
same as Option 5

Option 8: L_4 Driver
same as Option 5

Option 9: Operating voltage

	COP41XL	COP31XL
= 0:	+4.5V to +6.3V	+4.5V to +5.5V
= 1:	+4.5V to +9.5V	+4.5V to +7.5V

Option 10: L_3 Driver
same as Option 5

Option 11: L_2 Driver
same as Option 5

Option 12: L_1 Driver
same as Option 5

Option 13: L_0 Driver
same as Option 5

Option 14: SI Input
- = 0: load device to V_{CC}
- = 1: Hi-Z input

Option 15: SO Driver
- = 0: Standard Output
- = 1: Open-drain output
- = 2: Push-pull output

Option 16: SK Driver
same as Option 15

Option List (Continued)

Option 17: G_0 I/O Port
= 0: Standard output
= 1: Open-drain output

Option 18: G_1 I/O Port
same as Option 17

Option 19: G_2 I/O Port
same as Option 17

Option 20: G_3 I/O Port (no option available for COP411L)
same as Option 17

Option 21: D_3 Output (no option available for COP411L)
= 0: Very-high sink current standard output
= 1: Very-high sink current open-drain output
= 2: High sink current standard output
= 3: High sink current open-drain output
= 4: Standard LSTTL output (fanout = 1)
= 5: Open-drain LSTTL output (fanout = 1)

Option 22: D_2 Output (no option available for COP411L)
same as Option 21

Option 23: D_1 Output
same as Option 21

Option 24: D_0 Output
same as Option 21

Option 25: L Input Levels
= 0: Standard TTL input levels ("0" = 0.8V, "1" = 2.0V)
= 1: Higher voltage input levels ("0" = 1.2V, "1" = 3.6V)

Option 26: G Input Levels
same as Option 25

Option 27: SI Input Levels
same as Option 25

Option 28: COP Bonding
= 0: COP410L (24-pin device)
= 1: COP411L (20-pin device)
= 2: Both 24- and 20-pin versions

TEST MODE (NON-STANDARD OPERATION)

The SO output has been configured to provide for standard test procedures for the custom-programmed COP410L. With SO forced to logic "1", two test modes are provided, depending upon the value of SI:

a. RAM and Internal Logic Test Mode (SI = 1)

b. ROM Test Mode (SI = 0)

These special test modes should not be employed by the user; they are intended for manufacturing test only.

Option Table

The following option information is to be sent to National along with the EPROM.

Option Data		Option Data	
OPTION 1 VALUE =	IS: GROUND PIN	OPTION 15 VALUE =	IS: SO DRIVER
OPTION 2 VALUE =	IS: CKO PIN	OPTION 16 VALUE =	IS: SK DRIVER
OPTION 3 VALUE =	IS: CKI INPUT	OPTION 17 VALUE =	IS: G_0 I/O PORT
OPTION 4 VALUE =	IS: RESET INPUT	OPTION 18 VALUE =	IS: G_1 I/O PORT
OPTION 5 VALUE =	IS: L(7) DRIVER	OPTION 19 VALUE =	IS: G_2 I/O PORT
OPTION 6 VALUE =	IS: L(6) DRIVER	OPTION 20 VALUE =	IS: G_3 I/O PORT
OPTION 7 VALUE =	IS: L(5) DRIVER	OPTION 21 VALUE =	IS: D_3 OUTPUT
OPTION 8 VALUE =	IS: L(4) DRIVER	OPTION 22 VALUE =	IS: D_2 OUTPUT
OPTION 9 VALUE =	IS: V_{CC} PIN	OPTION 23 VALUE =	IS: D_1 OUTPUT
OPTION 10 VALUE =	IS: L(3) DRIVER	OPTION 24 VALUE =	IS: D_0 OUTPUT
OPTION 11 VALUE =	IS: L(2) DRIVER	OPTION 25 VALUE =	IS: L INPUT LEVELS
OPTION 12 VALUE =	IS: L(1) DRIVER	OPTION 26 VALUE =	IS: G INPUT LEVELS
OPTION 13 VALUE =	IS: L(0) DRIVER	OPTION 27 VALUE =	IS: SI INPUT LEVELS
OPTION 14 VALUE =	IS: SI INPUT	OPTION 28 VALUE =	IS: COPS BONDING

National Semiconductor Corporation

PRELIMINARY

COP413L/COP313L Single Chip Microcontrollers

General Description

The COP413L and COP313L Single-Chip N-Channel Microcontrollers are members of the COPS™ family, fabricated using N-channel, silicon gate MOS technology. These Control Oriented Processors are complete microcomputers containing all system timing, internal logic, ROM, RAM, and I/O necessary to implement dedicated control functions in a variety of applications. Features include single supply operation, 15 I/O lines with an instruction set, internal architecture and I/O scheme designed to facilitate keyboard input, display output and BCD data manipulation. They are an appropriate choice for use in numerous human interface control environments. Standard test procedures and reliable high-density fabrication techniques provide the medium to large volume customers with a customized Control Oriented Processor at a very low end-product cost.

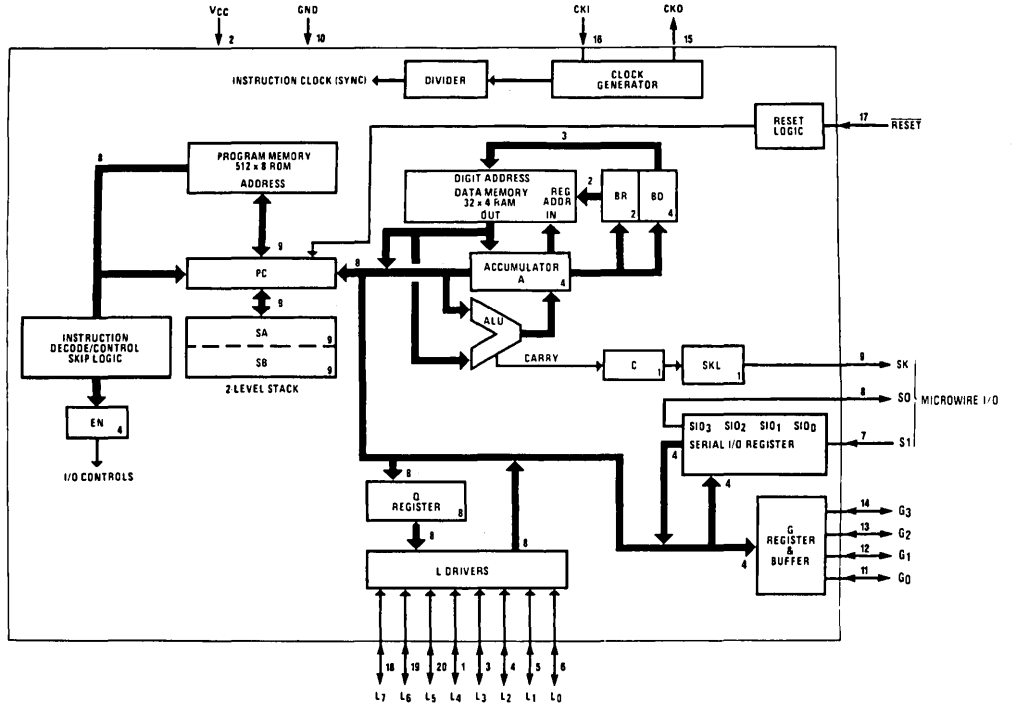
The COP313L is an exact functional equivalent but extended temperature version of the COP413L.

The COP401L-R13 and COP410L-X13 should be used for exact emulation.

Features

- Low cost
- Powerful instruction set
- 512 x 8 ROM, 32 x 4 RAM
- 15 I/O lines
- Two-Level subroutine stack
- 16 μs instruction time
- Single supply operation (4.5V–6.3V)
- Low current drain (6 mA max.)
- Internal binary counter register with MICROWIRE™ serial I/O capability
- General purpose outputs
- High noise immunity inputs (V_{IL}=1.2V, V_{IH}=3.6V)
- Software/hardware compatible with other members of COP400 family
- Extended temperature range device COP313L (−40°C to +85°C)

Block Diagram



TL/DD/8371–1

FIGURE 1

COP413L Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin Relative to GND	−0.3 to +7V
Ambient Operating Temperature	0°C to +70°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temp. (Soldering, 10 seconds)	300°C
Power Dissipation COP413L	0.3 Watt at 70°C
Total Source Current	25 mA
Total Sink Current	25 mA

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics

$0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 6.3V$ unless otherwise noted.

Parameter	Conditions	Min	Max	Units
Standard Operating Voltage (V_{CC})	(Note 1)	4.5	6.3	V
Power Supply Ripple	Peak to Peak		0.4	V
Operating Supply Current	All Inputs and Outputs Open		6	mA
Input Voltage Levels				
CKI Input Levels				
Ceramic Resonator Input (÷8)				
Logic High (V_{IH})		3.0		V
Logic Low (V_{IL})			0.4	V
CKI (RC), Reset Input Levels	(Schmitt Trigger Input)			
Logic High		0.7 V_{CC}		V
Logic Low			0.6	V
SO Input Level (Test Mode)	(Note 2)	2.5		V
SI Input Level				
Logic High	(TTL Level)	2.0		V
Logic Low			0.8	V
L, G Inputs				
Logic High	(High Trip Levels)	3.6		V
Logic Low			1.2	V
Input Capacitance			7	pF
Reset Input Leakage		−1	+1	μA
Output Current Levels				
Output Sink Current				
SO and SK Outputs (I_{OL})	$V_{OL}=0.4V$	0.9		mA
L0–L7 Outputs, G0–G3	$V_{OL}=0.4V$	0.4		mA
CKO (I_{OL})	$V_{OL}=0.4V$	0.2		mA
Output Source Current				
L0–L7 and G0–G3	$V_{OH}=2.4V$	−25		μA
SO and SK Outputs (I_{OH})	$V_{OH}=1.0V$	−1.2		mA
Push-Pull	$V_{OH}=2.4V$	−25		μA
SI Input Load Source Current	$V_{IL}=0V$	−10	−140	μA
Total Sink Current Allowed				
L7–L4, G Port			4	mA
L3–L0			4	mA
Any Other Pin			2.0	mA
Total Source Current Allowed				
Each Pin			1.5	mA

Note 1: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Note 2: SO output "0" level must be less than 0.8V for normal operation.

COP313L Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin Relative to GND	−0.3 to +7V
Ambient Operating Temperature	−40°C to +85°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temp. (Soldering, 10 seconds)	300°C
Power Dissipation COP313L	0.20 Watt at 85°C
Total Source Current	25 mA
Total Sink Current	25 mA

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 5.5V$ unless otherwise noted.

Parameter	Conditions	Min	Max	Units
Standard Operating Voltage (V_{CC})	(Note 1)	4.5	5.5	V
Power Supply Ripple	Peak to Peak		0.4	V
Operating Supply Current	All Inputs and Outputs Open		8	mA
Input Voltage Levels				
Ceramic Resonator Input (÷8)				
Logic High (V_{IH})		3.0		V
Logic Low (V_{IL})			0.3	V
CKI (RC), Reset Input Levels	(Schmitt Trigger Input)			
Logic High		0.7 V_{CC}		V
Logic Low			0.4	V
SO Input (Test Mode)	(Note 2)	2.5		V
SI Input Level				
Logic High	(TTL Level)	2.2		V
Logic Low			0.6	V
L, G Inputs				
Logic High	(High Trip Levels)	3.6		V
Logic Low			1.2	V
Input Capacitance			7	pF
Reset Input Leakage		−2	+2	μA
Output Current Levels				
Output Sink Current				
SO and SK Outputs (I_{OL})	V_{OL} = 0.4V	0.8		mA
L0–L7 Outputs, G0–G3 (I_{OL})	V_{OL} = 0.4V	0.4		mA
CKO (I_{OL})	V_{OL} = 0.4V	0.2		mA
Output Source Current				
L0–L7 and G0–G3	V_{OH} = 2.4V	−23		μA
SO and SK Outputs (I_{OH})	V_{OH} = 1.0V	−1.0		mA
(Push-Pull)	V_{OH} = 2.4V	−23		μA
SI Input Load Source Current	V_{IL} = 0V	−10	−200	μA
Total Sink Current Allowed				
L7–L4, G Port			4	mA
L3–L0			4	mA
Any Other Pin			1.5	mA
Total Source Current Allowed				
Each Pin			1.5	mA

Note 1: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Note 2: SO output "0" level must be less than 0.6V for normal operation.

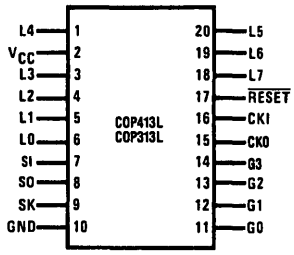
AC Electrical Characteristics

COP413L: $0°C \le T_A \le 70°C$, $4.5V \le V_{CC} \le 6.3V$
COP313L: $-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 5.5V$

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time - t_C		16	40	μs
CKI				
Input Frequency - fi	÷8 Mode	0.2	0.5	MHz
Duty Cycle		30	60	%
Rise Time	fi=0.5 MHz		500	ns
Fall Time			200	ns
CKI Using RC (÷4)	R=56 kΩ ±5% C=100 pF ±10%			
Instruction Cycle Time (Note 1)		16	28	μs
Inputs:				
G3–G0, L7–L0				
t_{SETUP}			8.0	μs
t_{HOLD}			1.3	μs
SI				
t_{SETUP}			2.0	μs
t_{HOLD}			1.0	μs
Output Propagation Delay	Test Condition: $C_L=50$ pF, $R_L=20$ kΩ, $V_{OUT}=1.5V$			
SO, SK Outputs			4.0	μs
tpd1, tpd0				
All Other Outputs				
tpd1, tpd0			5.6	μs

Note 1: Variation due to the device included.

Connection Diagram



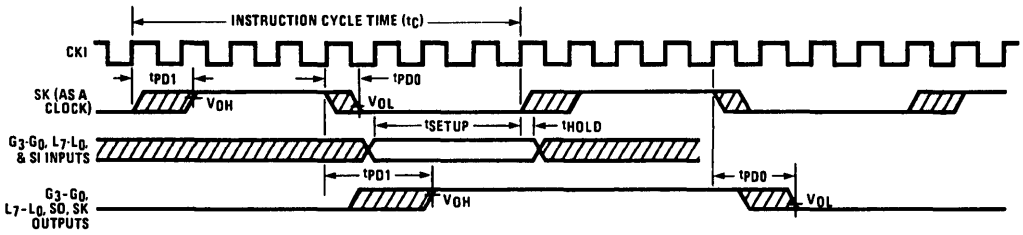
TL/DD/8371-2

FIGURE 2

Order Number COP313L-XXX/D, COP313LXXX/N or COP413L-XXX/D or COP413L-XXX/N or COP313L-XXX/M or COP413L-XXX/M
See NS Package Number D20A or N20A

Pin Descriptions

Pin	Description
L7–L0	8-bit bidirectional I/O port
G3–G0	4-bit bidirectional I/O port
SI	Serial input (or counter input)
SO	Serial output (or general purpose output)
SK	Logic-controlled clock (or general purpose output)
CKI	System oscillator input
CKO	System oscillator output or NC
$\overline{RESET}$	System reset input
V_{CC}	Power Supply
GND	Ground



TL/DD/8371-3

FIGURE 3. Input/Output Timing Diagrams (Ceramic Resonator Divide-by-8 Mode)

Functional Description

A block diagram of the COP413L is given in *Figure 1*. Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1" (greater than 2V). When a bit is reset, it is a logic "0" (less than 0.8V).

All functional references to the COP413L also apply to the COP313L.

PROGRAM MEMORY

Program Memory consists of a 512-byte ROM. As can be seen by an examination of the COP413L instruction set, these words may be program instructions, program data, or ROM addressing data. Because of the special characteristics associated with the JP, JSRP, JID and LQID instructions, ROM must often be thought of as being organized into 8 pages of 64 words each.

ROM addressing is accomplished by a 9-bit PC register. Its binary value selects one of the 512 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 9-bit binary count value. Two levels of subroutine nesting are implemented by the 9-bit subroutine save registers, SA and SB, providing a last-in, first out (LIFO) hardware subroutine stack.

ROM instruction words are fetched, decoded and executed by the Instruction Decode, Control and Skip Logic circuitry.

DATA MEMORY

Data memory consists of a 128-bit RAM, organized as 4 data registers of 8 4-bit digits. RAM addressing is implemented by a 6-bit B register whose upper 2 bits (Br) select 1 of 4 data registers and lower 3 bits of the 4-bit Bd select 1 of 8 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) is usually loaded into or from, or exchanged with, the A register (accumulator), it may also be loaded into the Q latches or loaded from the L ports. RAM addressing may also be performed directly by the XAD 3, 15 instruction.

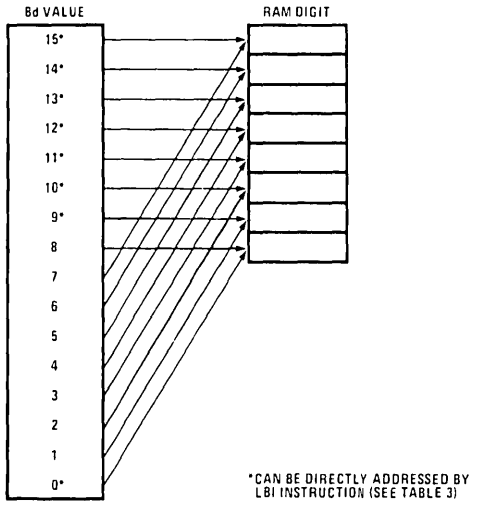
The most significant bit of Bd is not used to select a RAM digit. Hence each physical digit of RAM may be selected by two different values of Bd as shown in *Figure 4* below. The skip condition for XIS and XDS instructions will be true if Bd changes between 0 and 15, but NOT between 7 and 8 (see Table III).

INTERNAL LOGIC

The 4-bit A register (accumulator) is the source and destination register for most I/O, arithmetic, logic and data memory access operations. It can also be used to load the Bd portion of the B register, to load 4 bits of the 8-bit Q latch data, to input 4 bits of the 8-bit L I/O port data and to perform data exchanges with the SIO register.

A 4-bit adder performs the arithmetic and logic functions of the COP413L, storing its results in A. It also outputs a carry bit to the 1-bit C register, most often employed to indicate arithmetic overflow. The C register, in conjunction with the XAS instruction and the EN register, also serves to control the SK output. C can be outputted directly to SK or can enable SK to be a sync clock each instruction cycle time. (See XAS instruction and EN register description, below.)

The G register contents are outputs to 4 general purpose bidirectional I/O ports.



TL/DD/8371–4

FIGURE 4. RAM Digit Address to Physical RAM Digit Mapping

The Q register is an internal, latched, 8-bit register, used to hold data loaded from M and A, as well as 8-bit data from ROM. Its contents are output to the L I/O ports when the L drivers are enabled under program control. (See LEI instruction.)

The 8 L drivers, when enabled, output the contents of latched Q data to the L I/O ports. Also, the contents of L may be read directly into A and M.

The SIO register functions as a 4-bit serial-in/serial-out shift register or as a binary counter depending on the contents of the EN register. (See EN register description, below.) Its contents can be exchanged with A, allowing it to input or output a continuous serial data stream. SIO may also be used to provide additional parallel I/O by connecting SO to external serial-in/parallel-out shift registers.

The XAS instruction copies C into the SKL Latch. In the counter mode, SK is the output of SKL in the shift register mode, SK outputs SKL ANDed with internal instruction cycle clock.

The EN register is an internal 4-bit register loaded under program control by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register (EN_3–EN_0).

1. The least significant bit of the enable register, EN_0 selects the SIO register as either a 4-bit shift register or a 4-bit binary counter. With EN_0 set, SIO is an asynchronous binary counter, decrementing its value by one upon each low-going pulse ("1" to "0") occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output is equal to the value of EN_3. With EN_0 reset, SIO is a serial shift register shifting with each instruction cycle time. The data present at SO goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. (See 4 below.) The SK output becomes a logic-controlled clock.
2. EN_1 is not used. It has no effect on COP413L operation.

Functional Description (Continued)

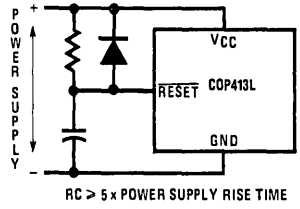
TABLE I. Enable Register Modes - Bits EN_3 and EN_0

EN_3	EN_0	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL = 1, SK = Clock If SKL = 0, SK = 0
1	0	Shift Register	Input to Shift Register	Serial Out	If SKL = 1, SK = Clock If SKL = 0, SK = 0
0	1	Binary Counter	Input to Binary Counter	0	If SKL = 1, SK = 1 If SKL = 0, SK = 0
1	1	Binary Counter	Input to Binary Counter	1	If SKL = 1, SK = 1 If SKL = 0, SK = 0

3. With EN_2 set, the L drivers are enabled to output the data in Q to the L I/O ports. Resetting EN_2 disables the L drivers, placing the L I/O ports in a high impedance input state.
4. EN_3, in conjunction with EN_0, affects the SO output. With EN_0 set (binary counter option selected) SO will output the value loaded into EN_3. With EN_0 reset (serial shift register option selected), setting EN_3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN_3 with the serial shift register option selected disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains reset to "0". Table I provides a summary of the modes associated with EN_3 and EN_0.

INITIALIZATION

The Reset Logic will initialize (clear) the device upon power-up if the power supply rise time is less than 1 ms and greater than 1 μs. If the power supply rise time is greater than 1 ms, the user must provide an external RC network and diode to the $\overline{\text{RESET}}$ pin as shown below (*Figure 5*). The $\overline{\text{RESET}}$ pin is configured as a Schmitt trigger input. If not used it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the $\overline{\text{RESET}}$ input, provided it stays low for at least three instruction cycle times.



TL/DD/8371-5

FIGURE 5. Power-Up Clear Circuit

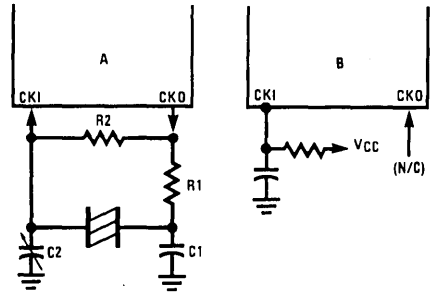
Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, EN, and G registers are cleared. The SK output is enabled as a SYNC output, providing a pulse each instruction cycle time. *Data Memory (RAM) is not cleared upon initialization.* The first instruction at address 0 must be a CLRA.

OSCILLATOR

There are two basic clock oscillator configurations available as shown by *Figure 6*.

a. Resonator Controlled Oscillator. CKI and CKO are connected to an external ceramic resonator. The instruction cycle frequency equals the resonator frequency divided by 8.

b. RC Controlled Oscillator. CKI is configured as a single pin RC controlled Schmitt trigger oscillator. The instruction cycle equals the oscillation frequency divided by 4. CKO becomes no connection.



TL/DD/8371-6

FIGURE 6. COP413L Oscillator

Ceramic Resonator Oscillator

Resonator Value	Component Values			
	R1 (Ω)	R2 (Ω)	C1 (pF)	C2 (pF)
455 kHz	4.7k	1M	220	220

RC Controlled Oscillator

R (kΩ)	C (pF)	Instruction Cycle Time (In μs)
51	100	19 ± 15%
82	56	19 ± 13%

Note: 200 kΩ ≥ R ≥ 25 kΩ
220 pF ≥ C ≥ 50 pF

Functional Description (Continued)

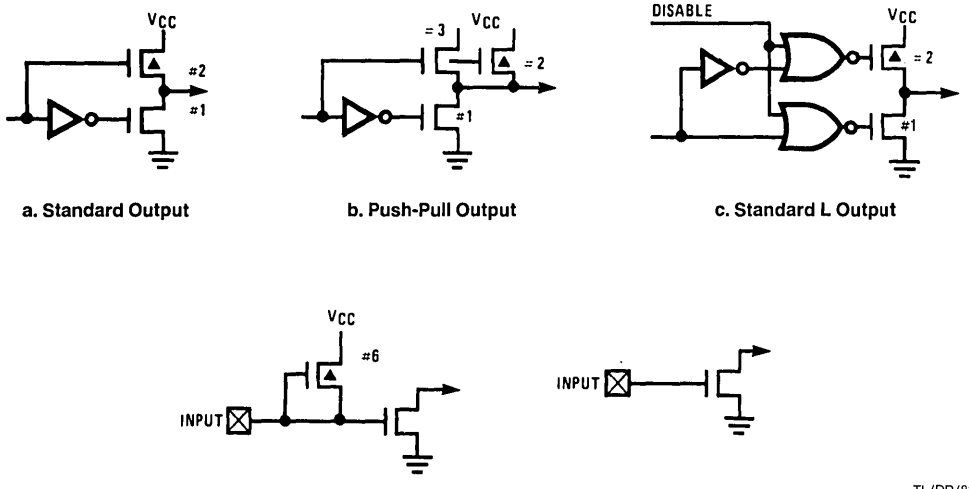


FIGURE 7. Input and Output Configurations

I/O CONFIGURATIONS

COP413L inputs and outputs have the following configurations, illustrated in *Figure 7*:

a. G0–G3—an enhancement mode device to ground in conjunction with a depletion-mode device to V_{CC}.

b. SO, SK—an enhancement mode device to ground in conjunction with a depletion-mode device paralleled by an enhancement-mode device to V_{CC}. This configuration has been provided to allow for fast rise and fall times when driving capacitive loads.

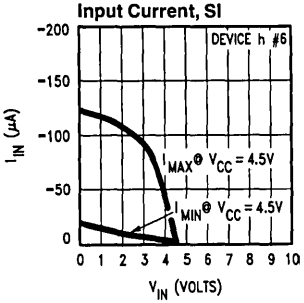
c. L0–L7—same as a., but may be disabled.

d. SI has on-chip depletion load device to V_{CC}.

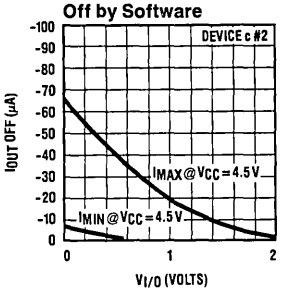
e. $\overline{\text{RESET}}$ has a Hi-Z input which must be driven to a "1" or "0" by external components.

Typical Performance Characteristics

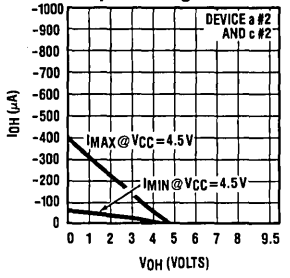
Input Current, SI



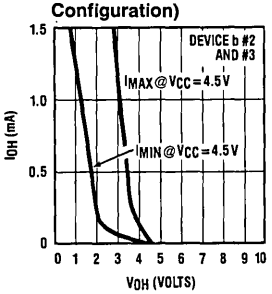
Input Current for L0 through L7 when Output Programmed Off by Software



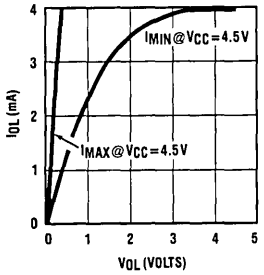
Source Current L7–L0, G3–G0 Standard Output Configuration



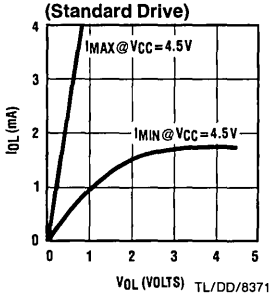
Source Current for SO and SK (Push-Pull Configuration)



Output Sink Current for SO and SK



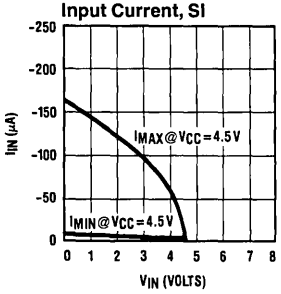
Output Sink Current for L0–L7, GO–G3 (Standard Drive)



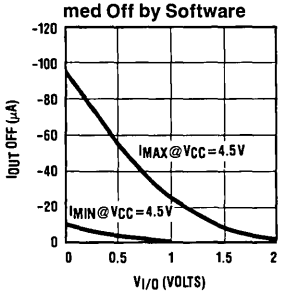
TL/DD/8371–8

FIGURE 8a. COP413L I/O DC Current Characteristics

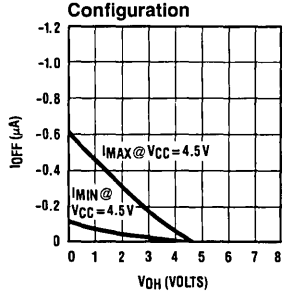
Input Current, SI



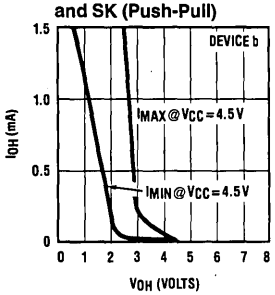
Input Current for L0–L7 when Output Programmed Off by Software



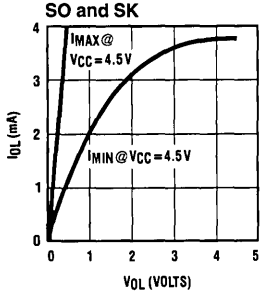
Source Current for L0–L7, G0–G3 Standard Output Configuration



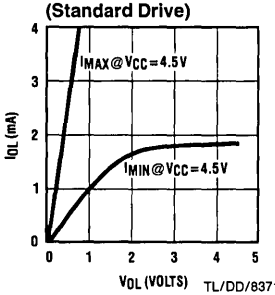
Source Current for SO and SK (Push-Pull)



Output Sink Current for SO and SK



Output Sink Current for L0–L7, G0–G3 (Standard Drive)



TL/DD/8371–9

FIGURE 8b. COP313L I/O DC Current Characteristics

COP413L Instruction Set

Table II is a symbol table providing internal architecture, instruction operand and operational symbols used in the instruction set table. Table III provides the mnemonic, operand, machine code data flow, skip conditions and description associated with each instruction in the COP413L instruction set.

TABLE II. COP413L Instruction Set Table Symbols

Symbol	Definition
Internal Architecture Symbols	
A	4-bit Accumulator
B	6-bit RAM Address Register
Br	Upper 2 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry Register
EN	4-bit Enable Register
G	4-bit Register to latch data for G I/O Port
L	8-bit TRI-STATE® I/O Port
M	4-bit contents of RAM Memory pointed to by B Register
PC	9-bit ROM Address Register (program counter)
Q	8-bit Register to latch data for L I/O Port
SA	9-bit Subroutine Save Register A
SB	9-bit Subroutine Save Register B
SIO	4-bit Shift Register and Counter
SK	Logic Controlled Clock Output
Instruction Operand Symbols	
d	4-bit Operand Field, 0–15 binary (RAM Digit Select)
r	2-bit Operand Field, 0–3 binary (RAM Register Select)
a	9-bit Operand Field, 0–511 binary (ROM Address)
y	4-bit Operand Field, 0–15 binary (Immediate Data)
RAM(s)	Contents of RAM location addressed by s
ROM(t)	Contents of ROM location addressed by t
Operational Symbols	
+	Plus
−	Minus
→	Replaces
⟷	Is exchanged with
=	Is equal to
$\overline{A}$	The one's complement of A
⊕	Exclusive-OR
:	Range of values

COP413L Instruction Set (Continued)

TABLE III. COP413L Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	|0011|0000|	A+C+RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	|0011|0001|	A+RAM(B) → A	None	Add RAM to A
AISC	y	5–	|0101| y |	A+y → A	Carry	Add Immediate, Skip on Carry (y≠0)
CLRA		00	|0000|0000|	0 → A	None	Clear A
COMP		40	|0100|0000|	$\overline{A}$ → A	None	One's complement of A to A
NOP		44	|0100|0100|	None	None	No Operation
RC		32	|0011|0010|	"0" → C	None	Reset C
SC		22	|0010|0010|	"1" → C	None	Set C
XOR		02	|0000|0010|	A⊕RAM(B) → A	None	Exclusive-OR RAM with A
TRANSFER OF CONTROL INSTRUCTIONS						
JID		FF	|1111|1111|	ROM(PC_8,A,M) → $PC_{7:0}$	None	Jump Indirect (Note 2)
JMP	a	6–	|0110|000|a_8|	a → PC	None	Jump
		–	| $a_{7:0}$ |			
JP	a	–	|1| $a_{6:0}$ | (pages 2, 3 only) or	a → $PC_{6:0}$	None	Jump within-Page (Note 3)
		–	|11| $a_{5:0}$ | (all other pages)	a → $PC_{5:0}$		
JSRP	a	–	|10| $a_{5:0}$ |	PC+1 → SA → SB 010 → $PC_{8:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 4)
JSR	a	6–	|0110|100|a_8|	PC+1 → SA → SB a → PC	None	Jump to Subroutine
		–	| $a_{7:0}$ |			
RET		48	|0100|1000|	SB → SA → PC	None	Return from Subroutine
RETSK		49	|0100|1001|	SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
MEMORY REFERENCE INSTRUCTIONS						
CAMQ		33	|0011|0011|	A → $Q_{7:4}$	None	Copy A, RAM to Q
		3C	|0011|1100|	RAM(B) → $Q_{3:0}$		
LD	r	–5	|00|r|0101|	RAM(B) → A Br⊕r → Br	None	Load RAM into A, Exclusive-OR Br with r
LQID		BF	|1011|1111|	ROM(PC_8, A,M) → Q SA → SB	None	Load Q Indirect (Note 2)
RMB	0	4C	|0100|1100|	0 → $RAM(B)_0$	None	Reset RAM Bit
	1	45	|0100|0101|	0 → $RAM(B)_1$		
	2	42	|0100|0010|	0 → $RAM(B)_2$		
	3	43	|0100|0011|	0 → $RAM(B)_3$		
SMB	0	4D	|0100|1101|	1 → $RAM(B)_0$	None	Set RAM Bit
	1	47	|0100|0111|	1 → $RAM(B)_1$		
	2	46	|0100|0110|	1 → $RAM(B)_2$		
	3	4B	|0100|1011|	1 → $RAM(B)_3$		

COP413L Instruction Set (Continued)

TABLE III. COP413L Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
MEMORY REFERENCE INSTRUCTIONS (Continued)						
STII	y	7–	|0111| y |	y → RAM(B) Bd+1 → Bd	None	Store Memory Immediate and Increment Bd
X	r	–6	|00|r|0110|	RAM(B) ⟷ A Br⊕r → Br	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	3,15	23 BF	|0010|0011| |1011|1111|	RAM(3,15) ⟷ A	None	Exchange A with RAM (3,15)
XDS	r	–7	|00|r|0111|	RAM(B) ⟷ A Bd–1 → Bd Br⊕r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd. Exclusive-OR Br with r
XIS	r	–4	|00|r|0100|	RAM(B) ⟷ A Bd+1 → Bd Br⊕r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd, Exclusive-OR Br with r
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	|0101|0000|	A → Bd	None	Copy A to Bd
CBA		4E	|0100|1110|	Bd → A	None	Copy Bd to A
LBI	r,d	–	|00|r|(d–1)| (d=0,9:15)	r,d → B	Skip until not a LBI	Load B immediate with r,d (Note 5)
LEI	y	33 6–	|0011|0011| |0110| y |	y → EN	None	Load EN Immediate (Note 6)
TEST INSTRUCTIONS						
SKC		20	|0010|0000|		C="1"	Skip if C is True
SKE		21	|0010|0001|		A=RAM(B)	Skip if A Equals RAM
SKGZ		33 21	|0011|0011| |0010|0001|		$G_{3:0}=0$	Skip if G is Zero (all 4 bits)
SKGBZ		33	|0011|0011|	1st byte		Skip if G Bit is Zero
	0	01	|0000|0001|	2nd byte	$G_0=0$	
	1	11	|0001|0001|	2nd byte	$G_1=0$	
	2	03	|0000|0011|	2nd byte	$G_2=0$	
	3	13	|0001|0011|	2nd byte	$G_3=0$	
SKMBZ	0	01	|0000|0001|		$RAM(B)_0=0$	Skip if RAM Bit is Zero
	1	11	|0001|0001|		$RAM(B)_1=0$	
	2	03	|0000|0011|		$RAM(B)_2=0$	
	3	13	|0001|0011|		$RAM(B)_3=0$	

COP413L Instruction Set (Continued)

TABLE III. COP413L Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
INPUT/OUTPUT INSTRUCTIONS						
ING		33	0011 | 0011	G → A	None	Input G Ports to A
		2A	0010 | 1010			
INL		33	0011 | 0011	$L_{7:4}$ → RAM(B)	None	Input L Ports to RAM, A
		2E	0010 | 1110	$L_{3:0}$ → A		
OMG		33	0011 | 0011	RAM(B) → G	None	Output RAM to G Ports
		3A	0011 | 1010			
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO (Note 2)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicity defined (e.g., Br and Bd are explicitly defined) Bits are numbered 0 to N where 0 signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit A register.

Note 2: For additional information on the operation of the XAS, JID, and LQID instructions, see below.

Note 3: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 4: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 5: The machine code for the lower 4 bits of the LBI instruction equals the binary value of the "d" data *minus 1* e.g., to load the lower four bits of B (Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Note 6: Machine code for operand field y for LEI instruction should equal the binary value to be latched into EN, where a "1" or "0" in each bit of EN corresponds with the selection or deselection of a particular function associated with each bit. (See Functional Description EN Register.)

Description of Selected Instructions

The following information is provided to assist the user in understanding the operation of several unique instructions and to provide notes useful to programmers in writing COP413L programs.

XAS INSTRUCTION

XAS (Exchange A with SIO) exchanges the 4-bit contents of the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. An XAS instruction will also affect the SK output. (See Functional Description, EN Register, above.) If SIO is selected as a shift register, an XAS instruction must be performed once every 4 instruction cycles to effect a continuous data stream.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower 8 bits of the ROM address register PC with the *contents* of ROM addressed by the 9-bit word, PC_8, A, M. PC_8 is not affected by this instruction.

Note that JID requires 2 instruction cycles to execute.

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 9-bit word PC_8, A, M. LQID can be used for table lookup or code conversion such as BCD to seven-segment. The LQID instruction "pushes" the stack (PC + 1 → SA → SB) and replaces the least significant 8 bits of PC as follows: A → $PC_{7:4}$, RAM (B) → $PC_{3:0}$, leaving PC_8 unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SB → SA → PC), restoring the saved value of PC to continue sequential program execution. Since LQID pushes SA → SB, the previous contents of SB are lost. Also, when LQID pops the stack, the previously pushed contents of SA are left in SB. The net result is that the contents of SA are placed in SB (SA → SB). Note that LQID takes two instruction cycle times to execute.

INSTRUCTION SET NOTES

a. The first word of a COP413L program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, one instruction cycle time is devoted to skipping each byte of the skipped instruction. Thus all program paths except JID and LQID take the same number of cycle times whether instructions are skipped or executed. JID and LQID instructions take 2 cycles if executed and 1 cycle if skipped.

c. The ROM is organized into 8 pages of 64 words each. The Program Counter is a 9-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID or LQID instruction is located in the last word of a page, the instruction operates as if it were in the next page. For example: a JP located in the last word of a page will jump to a location in the next page. Also, a LQID or JID located in the last word of page 3 or will access data in the next group of 4 pages.

Description of Selected Instructions (Continued)

TEST MODE (NON-STANDARD OPERATION)

The SO output has been configured to provide for standard test procedures for the custom-programmable COP413L. With SO forced to logic "1", two test modes are provided, depending upon the value of SI:

a. RAM and internal Logic Test Mode (SI = 1)

b. ROM Test Mode (SI = 0)

These special test modes should not be employed by the user; they are intended for manufacturing test only.

Option List

The option selected must be sent in with the EPROM of ROM Code for a Mask order of 413L. Make xerox copy of the table, select the appropriate option, and send it in with the EPROM.

COP 413L/COP 313L

Option 1: Oscillator Selection

= 0 Ceramic Resonator or external input frequency divided by 8. CKO is oscillator output.

= 1 Single pin RC controlled oscillator divided by 4. CKO is no connection.

NOTE:

The following option information is to be sent to National along with the EPROM

Option 1: Value = _____ is: Oscillator Selection

National Semiconductor Corporation

PRELIMINARY

microCMOS

COP413C/COP413CH/COP313C/COP313CH Single-Chip CMOS Microcontrollers

General Description

The COP413C, COP413CH, COP313C, and COP313CH fully static, single-chip CMOS microcontrollers are members of the COPS™ family, fabricated using double-poly, silicon-gate CMOS technology. These controller-oriented processors are complete microcomputers containing all system timing, internal logic, ROM, RAM, and I/O necessary to implement dedicated control functions in a variety of applications. Features include single supply operation, with an instruction set, internal architecture, and I/O scheme designed to facilitate keyboard input, display output, and BCD data manipulation. The COP413CH is identical to the COP413C except for operating voltage and frequency. They are an appropriate choice for use in numerous human interface control environments. Standard test procedures and reliable high-density fabrication techniques provide a customized controller-oriented processor at a low end-product cost.

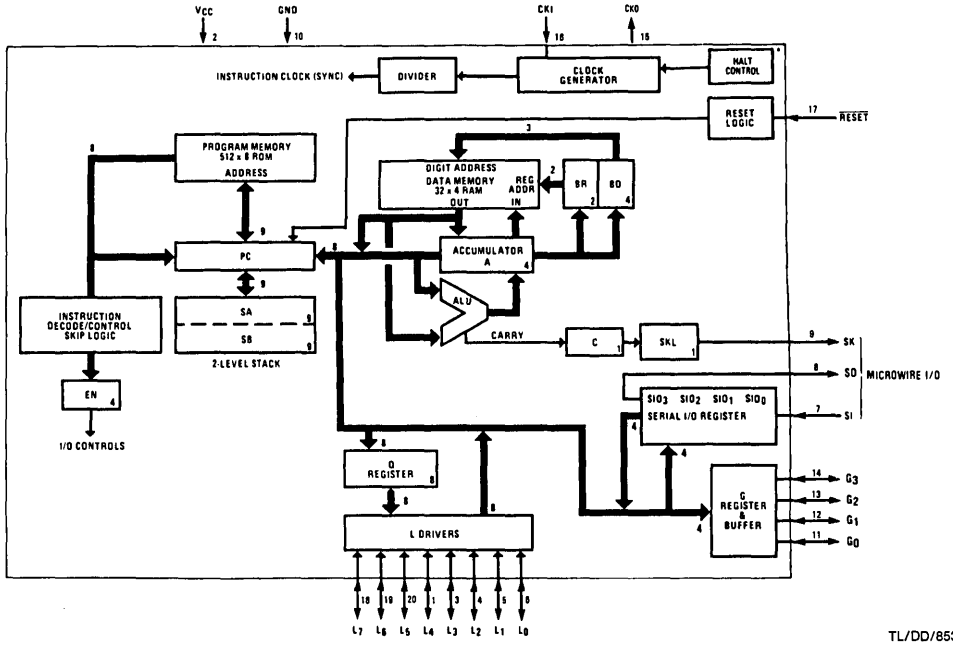
The COP313C/COP313CH is the extended temperature range version of the COP413C/COP413CH.

For emulation use the ROMless COP404C.

Features

- Lowest power dissipation (40 μW typical)
- Low cost
- Power-saving HALT Mode
- Powerful instruction set
- 512 x 8 ROM, 32 x 4 RAM
- 16 I/O lines
- Two-level subroutine stack
- DC to 4 μs instruction time
- Single supply operation (3V to 5.5V)
- General purpose and TRI-STATE® outputs
- Internal binary counter register with MICROWIRE™ compatible serial I/O
- Software/hardware compatible with other members of the COP400 family
- Extended temperature (−40°C to +85°C) devices available

Block Diagram



TL/DD/8537–1

FIGURE 1. COP413C/413CH

COP413C/COP413CH

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage	6V
Voltage at Any Pin	−0.3V to V_{CC} + 0.3V
Total Allowable Source Current	25 mA
Total Allowable Sink Current	25 mA
Operating Temperature Range	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $0°C \le T_A \le +70°C$ unless otherwise specified

Parameter	Conditions	COP413C		COP413CH		Units
		Min	Max	Min	Max	
Operating Voltage		3.0	5.5	4.5	5.5	V
Power Supply Ripple (Note 4)			0.1 V_{CC}		0.1 V_{CC}	V
Supply Current (Note 1)	V_{CC} = 5.0V, t_c = Min		500		2000	μA
	V_{CC} = 3.0V, t_c = Min		300			μA
	(t_c is inst. cycle)					
HALT Mode Current (Note 2)	V_{CC} = 5.0V, F_I = 0 kHz		20		20	μA
	V_{CC} = 3.0V, F_I = 0 kHz		10			μA
Input Voltage Levels						
RESET, CKI						
Logic High		0.9 V_{CC}		0.9 V_{CC}		V
Logic Low			0.1 V_{CC}		0.1 V_{CC}	V
All Other Inputs						
Logic High		0.7 V_{CC}		0.7 V_{CC}		V
Logic Low			0.2 V_{CC}		0.2 V_{CC}	V
RESET, SI Input Leakage		−1	+1	−1	+1	μA
Input Capacitance			7		7	pF
Output Voltage Levels						
(SO, SK, L Port)						
Logic High	I_{OH} = −10 μA	V_{CC} − 0.2		V_{CC} − 0.2		V
Logic Low	I_{OL} = 10 μA		0.2		0.2	V
Output Current Levels						
Sink (Note 3)	V_{CC} = Min, V_{OUT} = V_{CC}	0.2		1.2		mA
Source (SO, SK, L Port)	V_{CC} = Min, V_{OUT} = 0V	−0.1		−0.5		mA
Source (G Port)	V_{CC} = Min, V_{OUT} = 0V	−8	−150	−30	−330	μA
Allowable Sink/Source Current Per Pin (Note 3)			5		5	mA
TRI-STATE Leakage Current		−2	+2	−2	+2	μA

1

COP413C/COP413CH

AC Electrical Characteristics $0°C \leq T_A \leq 70°C$ unless otherwise specified

Parameter	Conditions	COP413C		COP413CH		Units
		Min	Max	Min	Max	
Instruction Cycle Time		16	DC	4	DC	μs
Operating CKI Frequency	÷8 Mode	DC	500	DC	2000	kHz
Instruction Cycle Time RC Oscillator ÷ 4	R = 30k ±5%, V_{CC} = 5V C = 82 pF ± 5%			8	16	μs
Instruction Cycle Time RC Oscillator ÷ 4 (Note 6)	R = 56k ±5%, V_{CC} = 5V C = 100 pF ± 5%	16	32	16	32	μs
Duty Cycle (Note 5)	Fi = Max freq ext clk	40	60	40	60	%
Rise Time (Note 5)	Fi = Max freq ext clk		60		60	ns
Fall Time (Note 5)	Fi = Max freq ext clk		40		40	ns
Inputs (See *Figure 3*)						
t_{SETUP}	G Inputs	tc/4 + 2.8		tc/4 + 0.7		μs
	SI Input	1.2		0.3		μs
	L Inputs	6.8		1.7		μs
t_{HOLD}		1.0		0.25		μs
Output Propagation Delay t_{PD1}, t_{PD0}	V_{OUT} = 1.5, C_L = 100 pF R_L = 5k		4.0		1.0	μs

Note 1: Supply current is measured after running for 2000 cycle times with a square-wave clock on CKI, CKO open, and all other pins pulled to V_{CC} with 5k resistors. See current drain equation on page 13.

Note 2: The Halt mode will stop CKI from oscillating.

Note 3: SO output sink current must be limited to keep V_{OL} less tha 0.2 V_{CC} when part is running in order to prevent entering test mode.

Note 4: Voltage change must be less than 0.5V in a 1 ms period.

Note 5: This parameter is only sampled and not 100% tested.

Note 6: Variation due to the device included.

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage	6V
Voltage at Any Pin	−0.3V to V_{CC} + 0.3V
Total Allowable Source Current	25 mA
Total Allowable Sink Current	25 mA
Operating Temperature Range	−40°C to +85°C
Storage Temperature Range	−65°C to +150°C

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $-40°C \le T_A \le +85°C$ unless otherwise specified

Parameter	Conditions	COP313C		COP313CH		Units
		Min	Max	Min	Max	
Operating Voltage		3.0	5.5	4.5	5.5	V
Power Supply Ripple (Note 4)			0.1 V_{CC}		0.1 V_{CC}	V
Supply Current (Note 1)	V_{CC} = 5.0V, t_c = Min V_{CC} = 3.0V, t_c = Min (t_c is inst. cycle)		600 360		2500	μA μA
Halt Mode Current (Note 2)	V_{CC} = 5.0V, Fi = 0 kHz V_{CC} = 3.0V, Fi = 0 kHz		20 10		20	μA μA
Input Voltage Levels $\overline{RESET}$, CKI Logic High Logic Low All Other Inputs Logic High Logic Low		 0.9 V_{CC} 0.7 V_{CC}	 0.1 V_{CC} 0.2 V_{CC}	 0.9 V_{CC} 0.7 V_{CC}	 0.1 V_{CC} 0.2 V_{CC}	 V V V V
$\overline{RESET}$, SI Input Leakage		−2	+2	−2	+2	μA
Input Capacitance			7		7	pF
Output Voltage Levels (SO, SK, L Port) Logic High Logic Low	 I_{OH} = −10 μA I_{OL} = 10 μA	 V_{CC} − 0.2	 0.2	 V_{CC} − 0.2	 0.2	 V V
Output Current Levels Sink (Note 3) Source (SO, SK, L Port) Source (G Port)	 V_{CC} = Min, V_{OUT} = V_{CC} V_{CC} = Min, V_{OUT} = 0V V_{CC} = Min, V_{OUT} = 0V	 0.2 −0.1 −8	 −200	 1.2 −0.5 −30	 −440	 mA mA μA
Allowable Sink/Source Current Per Pin (Note 3)			5		5	mA
TRI-STATE Leakage Current[3]		−4	+4	−4	+4	μA

COP313C/COP313CH

AC Electrical Characteristics $-40°C \leq T_A \leq +85°C$ unless otherwise specified

Parameter	Conditions	COP313C		COP313CH		Units
		Min	Max	Min	Max	
Instruction Cycle Time		16	DC	4	DC	μs
Operating CKI Frequency	÷ 8 Mode	DC	500	DC	2000	kHz
Instruction Cycle Time RC Oscillator ÷ 4	R = 30k ±5%, V_{CC} = 5V C = 82 pF ± 5%			8	16	μs
Instruction Cycle Time RC Oscillator ÷ 4 (Note 6)	R = 56k ±5%, V_{CC} = 5V C = 100 pF ± 5%	16	32	16	32	μs
Duty Cycle (Note 5)	Fi = Max Freq Ext Clk	40	60	40	60	%
Rise Time (Note 5)	Fi = Max Freq Ext Clk		60		60	ns
Fall Time (Note 5)	Fi = Max Freq Ext Clk		40		40	ns
Inputs (See *Figure 3*)						
t_{SETUP}	G Inputs	tc/4 + 2.8		tc/4 + 0.7		μs
	SI Input	1.2		0.3		μs
	L Inputs	6.8		1.7		μs
t_{HOLD}		1.0		0.25		μs
Output Propagation Delay t_{PD1}, t_{PD0}	V_{OUT} = 1.5V, C_L = 100 pF R_L = 5k		4.0		1.0	μs

Note 1: Supply current is measured after running for 2000 cycle times with a square-wave clock on CKI, CKO open, and all other pins pulled up to V_{CC} with 5k resistors. See current drain equation on page 13.

Note 2: The Halt mode will stop CKI from oscillating.

Note 3: SO output sink current must be limited to keep V_{OL} less than 0.2 V_{CC} when part is running in order to prevent entering test mode.

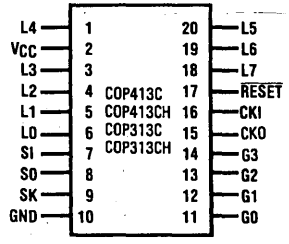
Note 4: Voltage change must be less than 0.5V in a 1 ms period.

Note 5: This parameter is only sampled and not 100% tested.

Note 6: Variation due to the device included.

Connection Diagram

Dual-In-Line



TL/DD/8537-2

Top View

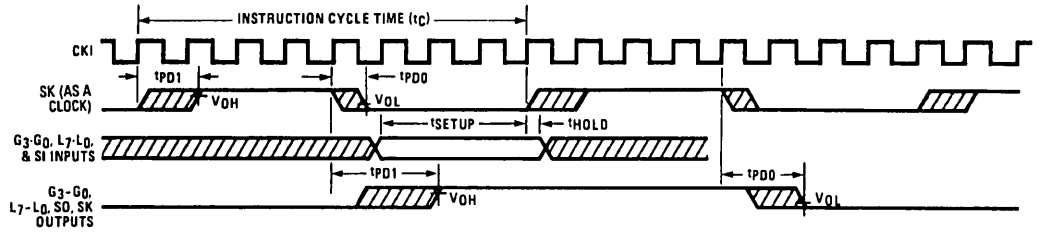
Pin Descriptions

Pin	Description
L_7–L_0	8-bit bidirectional I/O port with TRI-STATE
G_3–G_0	4-bit bidirectional I/O port
SI	Serial input (or counter input)
SO	Serial output (or general purpose output)
SK	Logic-controlled clock (or general purpose output)
CKI	System oscillator input
CKO	Crystal oscillator output, or NC
$\overline{RESET}$	System reset input
V_{CC}	System power supply
GND	System Ground

FIGURE 2

Order Number COP313C-XXX/D, COP313C-XXX/M, COP313C-XXX/N, COP313CH-XXX/D, COP313CH-XXX/M, COP313CH-XXX/N, COP413C-XXX/D, COP413C-XXX/M, COP413C-XXX/N, COP413CH-XXX/D, COP413CH-XXX/M or COP413CH-XXX/N
See NS Package Number D20A, M20B or N20A

Timing Waveform



TL/DD/8537-3

FIGURE 3. Input/Output Timing Diagrams (Divide-by-8 Mode)

Development Support

The MOLE (Microcontroller On Line Emulator) is a low cost development system and real time emulator for COPS' products. They also include TMP, 8050 and the new 16 bit HPC microcontroller family. The MOLE provides effective support for the development of both software and hardware in the user's application.

The purpose of the MOLE is to provide a tool to write and assemble code, emulate code for the target microcontroller and assist in debugging of the system.

The MOLE can be connected to various hosts, IBM PC, STARPLEX™, Kaypro, Apple and Intel systems, via RS-232 port. This link facilitates the up loading/down loading of code, supports host assembly and mass storage.

The MOLE consists of three parts; brain, personality and optional host software.

The brain board is the computing engine of the system. It is a self-contained computer with its own firmware which provides for all system operation, emulation control, communication, from programming and diagnostic operation. It has three serial ports which can be connected to a terminal, host system, printer, modem or to other MOLE's in a multi-MOLE environment.

The personality board contains the necessary hardware and firmware needed to emulate the target microcontroller. The emulation cable which replaces the target controller attaches to this board. The software contains a cross assembler and a communications program for up loading and down loading code from the MOLE.

MOLE Ordering Information

P/N	Description
MOLE-BRAIN	MOLE Computer Board
MOLE-COPS-PB1	COPS Personality Board
MOLE-XXX-YYY	Optional Software

Where XXX = COPS
YYY = Host System, IBM, Apple, KAY (Kaypro), CP/M

Functional Description

To ease reading of this description, only COP413C is referenced; however, all such references apply equally to COP413CH, COP313C, and COP313CH.

A block diagram of the COP413C is given in *Figure 1.* Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1"; when a bit is reset, it is a logic "0".

PROGRAM MEMORY

Program memory consists of a 512-byte ROM. As can be seen by an examination of the COP413C instruction set, these words may be program instructions, program data, or ROM addressing data. Because of the special characteristics associated with the JP, JSRP, JID, and LQID instructions, ROM must often be thought of as being organized into 8 pages of 64 words (bytes) each.

ROM ADDRESSING

ROM addressing is accomplished by a 9-bit PC register. Its binary value selects one of the 512 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 9-bit binary count value. Two levels of subroutine nesting are implemented by two 9-bit subroutine save registers, SA and SB.

ROM instruction words are fetched, decoded, and executed by the instruction decode, control and skip logic circuitry.

DATA MEMORY

Data Memory consists of a 128-bit RAM, organized as four data registers of 8 × 4-bit digits. RAM addressing is implemented by a 6-bit B register whose upper two bits (Br) selects one of four data registers and lower three bits of the 4-bit Bd select one of eight 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) are usually loaded into or from, or exchanged with, the A register (accumulator), they may also be loaded into the Q latches or loaded from the L ports. RAM addressing may also be performed directly by the XAD 3, 15 instruction.

The most significant bit of Bd is not used to select a RAM digit. Hence, each physical digit of RAM may be selected by two different values of Bd as shown in *Figure 4.* The skip condition for XIS and XDS instructions will be true if Bd changes between 0 to 15, but *not* between 7 and 8 (see Table III).

INTERNAL LOGIC

The internal logic of the COP413C is designed to ensure fully static operation of the device.

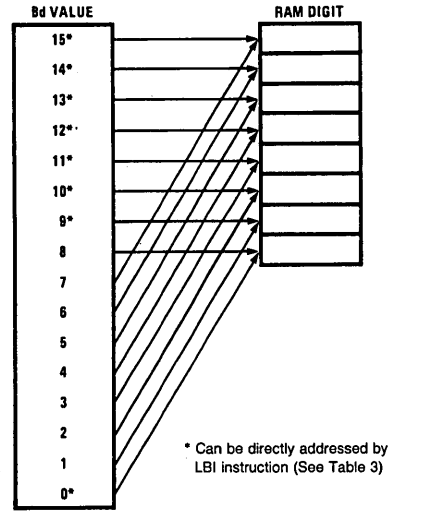
The 4-bit A register (accumulator) is the source and destination register for most I/O, arithmetic, logic and data memory access operations. It can also be used to load the Bd portion of the B register, to load four bits of the 8-bit Q latch data and to perform data exchanges with the SIO register.

The 4-bit adder performs the arithmetic and logic functions of the COP413C, storing its results in A. It also outputs the carry information to a 1-bit carry register, most often employed to indicate arithmetic overflow. The C register, in conjunction with the XAS instruction and the EN register, also serves to control the SK output. C can be outputted directly to SK or can enable SK to be a sync clock each instruction cycle time. (See XAS instruction and EN register description below.)

The G register contents are outputs to four general purpose bidirectional I/O ports.

The Q register is an internal, latched, 8-bit register, used to hold data loaded from RAM and A, as well as 8-bit data from ROM. Its contents are output to the L I/O ports when the L drivers are enabled under program control. (See LEI instruction.)

The eight L drivers, when enabled, output the contents of latched Q data to the L I/O ports. Also, the contents of L may be read directly into A and RAM.



TL/DD/8537–4

FIGURE 4. RAM Digit Address to Physical RAM Digit Mapping

Functional Description (Continued)

The SIO register functions as a 4-bit serial-in/serial-out shift register or as a binary counter, depending upon the contents of the EN register. (See EN register description below.) Its contents can be exchanged with A, allowing it to input or output a continuous serial data stream. With SIO functioning as a serial-in/serial-out shift register and SK as a sync clock, the COP413C is MICROWIRE compatible.

The XAS instruction copies C into the SKL latch. In the counter mode, SK is the output of SKL; in the shift register mode, SK is a sync clock, inhibited when SKL is a logic "0".

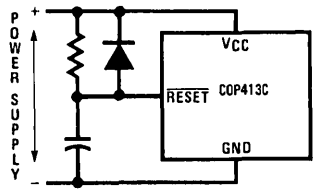
The EN register is an internal 4-bit register loaded under program control by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register (EN3–EN0).

1. The least significant bit of the enable register, EN0, selects the SIO register as either a 4-bit shift register or as a 4-bit binary counter. With EN0 set, SIO is an asynchronous binary counter, *decrementing* its value by one upon each low-going pulse ("1" to "0") occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output is equal to the value of EN3. With EN0 reset, SIO is a serial shift register, shifting left each instruction cycle time. The data present at SI is shifted into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each instruction cycle time. (See 4, below.) The SK output becomes a logic-controlled clock.
2. EN 1 is not used, it has no effect on the COP413C.
3. With EN2 set, the L drivers are enabled to output the data in Q to the L I/O ports. Resetting EN2 disables the L drivers, placing the L I/O ports in a high impedance input state.
4. EN3, in conjunction with EN0, affects the SO output. With EN0 set (binary counter option selected), SO will output the value loaded into EN3. With EN0 reset (serial shift register option selected), setting EN3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN3 with the serial shift register option selected, disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains reset to "0".

INITIALIZATION

The external RC network shown in *Figure 5* must be connected to the $\overline{\text{RESET}}$ pin. The $\overline{\text{RESET}}$ pin is configured as a Schmitt trigger input. If not used, it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the $\overline{\text{RESET}}$ input, providing it stays low for at least three instruction cycle times.

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, EN, and G registers are cleared. The SK output is enabled as a SYNC output, providing a pulse each instruction cycle time. Data memory (RAM) is not cleared upon initialization. The first instruction at address 0 must be a CLRA (clear A register).



TL/DD/8537–5

RC > 5 × Power Supply Rise Time
and RC > 100 × CKI Period

FIGURE 5. Power-Up Clear Circuit

TABLE I. Enable Register Modes—Bits EN0 and EN3

EN0	EN3	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL = 1, SK = clock If SKL = 0, SK = 0
0	1	Shift Register	Input to Shift Register	Serial out	If SKL = 1, SK = clock If SKL = 0, SK = 0
1	0	Binary Counter	Input to Counter	0	SK = SKL
1	1	Binary Counter	Input to Counter	1	SK = SKL

Functional Description (Continued)

HALT MODE

The COP413C is a *fully static* circuit; therefore, the user may stop the system oscillator at any time to halt the chip. The chip may be halted by the HALT instruction. Once in the HALT mode, the internal circuitry does not receive any clock signal, and is therefore frozen in the exact state it was in when halted. All information is retained until continuing. The HALT mode is the minimum power dissipation state.

The HALT mode may be entered into by program control (HALT instruction) which forces CKO to a logic "1" state. The circuit can be awakened only by the $\overline{\text{RESET}}$ function.

POWER DISSIPATION

The lowest power drain is when the clock is stopped. As the frequency increases so does current. Current is also lower at lower operating voltages. Therefore, to minimize power consumption, the user should run at the lowest speed and voltage that his application will allow. The user should take care that all pins swing to full supply levels to ensure that outputs are not loaded down and that inputs are not at some intermediate level which may draw current. Any input with a slow rise or fall time will draw additional current. A crystal- or resonator-generated clock will draw more than a square-wave input. An RC oscillator will draw even more current since the input is a slow rising signal.

If using an external squarewave oscillator, the following equation can be used to calculate the COP413C current drain.

$Ic = Iq + (V \times 20 \times Fi) + (V \times 1280 \times Fl/Dv)$

where Ic = chip current drain in microamps

Iq = quiescent leakage current (from curve)

Fl = CKI frequency in megahertz

V = chip V_{CC} in volts

Dv = divide by option selected

For example, at 5V V_{CC} and 400 kHz (divide by 8),

$Ic = 30 + (5 \times 20 \times 0.4) + (5 \times 1280 \times 0.4/8)$

$Ic = 30 + 40 + 320 = 390\ \mu A$

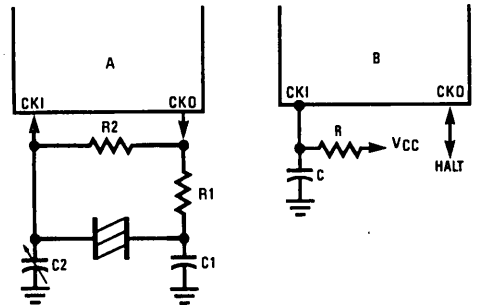
OSCILLATOR OPTIONS

There are two options available that define the use of CKI and CKO.

a. Cyrstal-Controlled Oscillator. CKI and CKO are connected to an external crystal. The instruction cycle time equals the crystal frequency divided by 8.

b. RC-Controlled Oscillator. CKI is configured as a single pin RC-controlled Schmitt trigger oscillator. The instruction cycle equals the oscillation frequency divided by 4. CKO is NC.

The RC oscillator is not recommended in systems that require accurate timing or low current. The RC oscillator draws more current than an external oscillator (typically an additional 100 μA at 5V). However, when the part halts, it stops with CKI high and the halt current is at the minimum.



TL/DD/8537-6

FIGURE 6. COP413C Oscillator

Crystal or Resonator					RC-Controlled Oscillator			
Crystal Value	Component Value R1	R2	C1 pF	C2 pF	R	C	Cycle Time	V_{CC}
32 kHz	220k	20M	30	5–36	15k	82 pF	4–9 μs	≥ 4.5V COP413CH Only
455 kHz	5k	10M	80	40	30k	82 pF	8–16 μs	≥ 4.5V COP413CH Only
2.000 MHz	2k	1M	30	6–36	47k	100 pF	16–32 μs	3.0 to 4.5V COP413C Only
					56k	100 pF	16–32 μs	≥ 4.5V
					Note: 15k ≤ R ≤ 150k, 50 pF ≤ C ≤ 150 pF			

Functional Description (Continued)

I/O CONFIGURATIONS

COP413C outputs have the following configurations, illustrated in *Figure 7*:

a. Standard SO, SK Output. A CMOS push-pull buffer with an N-channel device to ground in conjunction with a P-channel device to V_{CC}, compatible with CMOS and LSTTL.

b. Low Current G Output. This is the same configuration as (a) above except that the sourcing current is much less.

c. Standard TRI-STATE L Output. L output is a CMOS output buffer similar to (a) which may be disabled by program control.

The SI and $\overline{\text{RESET}}$ inputs are Hi-Z inputs (*Figure 7d*).

When using the G I/O port as an input, set the output register to a logic "1" level. The P-channel device will act as a pull-up load. When using the L I/O port as an input, disable the L drivers with the LEI instruction. The drivers are then in TRI-STATE mode and can be driven externally.

All output drivers use one or more of three common devices numbered 1 to 3. Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figure 8* for each of these devices to allow the designer to effectively use these I/O configurations.

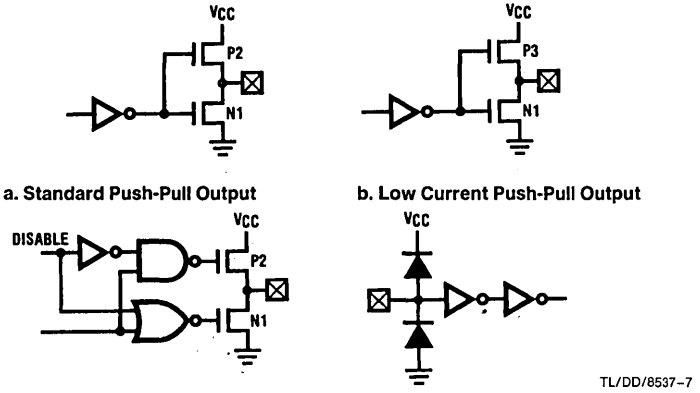


FIGURE 7. I/O Configurations

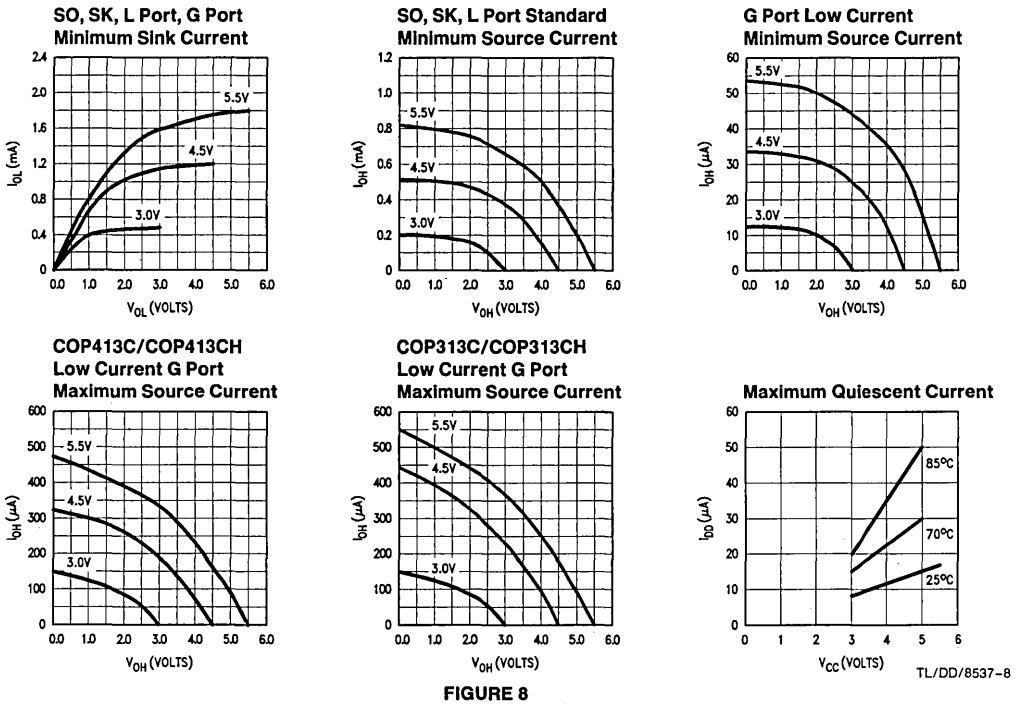


FIGURE 8

COP413C Instruction Set

Table II is a symbol table providing internal architecture, instruction operand and operational symbols used in the instruction set table.

Table III provides the mnemonic, operand, machine code, data flow, skip conditions and description associated with each instruction in the COP413C instruction set.

TABLE II. COP413C Instruction Set Table Symbols

Symbol	Definition
INTERNAL ARCHITECTURE SYMBOLS	
A	4-bit Accumulator
B	6-bit RAM Address Register
Br	Upper 2 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry Register
EN	4-bit Enable Register
G	4-bit Register to latch data for G I/O Port
L	8-bit TRI-STATE I/O Port
M	4-bit contents of RAM Memory pointed to by B Register
PC	9-bit ROM Address Register (program counter)
Q	8-bit Register to latch data for L I/O Port
SA	9-bit Subroutine Save Register A
SB	9-bit Subroutine Save Register B
SIO	4-bit Shift Register and Counter
SK	Logic-Controlled Clock Output

Symbol	Definition
INSTRUCTION OPERAND SYMBOLS	
d	4-bit Operand Field, 0–15 binary (RAM Digit Select)
r	2-bit Operand Field, 0–3 binary (RAM Register Select)
a	9-bit Operand Field, 0–511 binary (ROM Address)
y	4-bit Operand Field, 0–15 binary (Immediate Data)
RAM(s)	Contents of RAM location addressed by s
ROM(t)	Contents of ROM location addressed by t
OPERATIONAL SYMBOLS	
+	Plus
−	Minus
→	Replaces
⟷	Is exchanged with
=	Is equal to
$\overline{A}$	The one's complement of A
⊕	Exclusive-OR
:	Range of values

TABLE III. COP413C Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	|0011|0000|	A + C + RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	|0011|0001|	A + RAM(B) → A	None	Add RAM to A
AISC	y	5–	|0101| y |	A + y → A	Carry	Add immediate, Skip on Carry (y ≠ 0)
CLRA		00	|0000|0000|	0 → A	None	Clear A
COMP		40	|0100|0000|	$\overline{A}$ → A	None	One's complement of A to A
NOP		44	|0100|0100|	None	None	No Operation
RC		32	|0011|0010|	"0" → C	None	Reset C
SC		22	|0010|0010|	"1" → C	None	Set C
XOR		02	|0000|0010|	A ⊕ RAM(B) → A	None	Exclusive-OR RAM with A

Instruction Set (Continued)

TABLE III. COP413C Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
TRANSFER OF CONTROL INSTRUCTIONS						
JID		FF	1111 | 1111	ROM (PC_8, A,M) → $PC_{7:0}$	None	Jump Indirect (Note 2)
JMP	a	6–	0110 | 000 | a_8	a → PC	None	Jump
		–	$a_{7:0}$			
JP	a	–	1 | $a_{6:0}$ (pages 2, 3 only)	a → $PC_{6:0}$	None	Jump within Page (Note 1)
			or			
		–	11 | $a_{5:0}$ (all other pages)	a → $PC_{5:0}$		
JSRP	a	–	10 | $a_{5:0}$	PC + 1 → SA → SB 010 → $PC_{8:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 2)
JSR	a	6–	0110 | 100 | a_8	PC + 1 → SA → SB a → PC	None	Jump to Subroutine
		–	$a_{7:0}$			
RET		48	0100 | 1000	SB → SA → PC	None	Return from Subroutine
RETSK		49	0100 | 10011	SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
HALT		33	0011 | 0011		None	Halt processor
		38	0011 | 1000			
MEMORY REFERENCE INSTRUCTIONS						
CAMQ		33	0011 | 0011	A → $Q_{7:4}$	None	Copy A, RAM to Q
		3C	0011 | 1100	RAM(B) → $Q_{3:0}$		
CQMA		33	0011 | 0011	$Q_{7:4}$ → RAM(B)	None	Copy Q to RAM, A
		2C	0010 | 1100	$Q_{3:0}$ → A		
LD	r	−5	00 | r | 0101	RAM(B) → A Br ⊕ r → Br	None	Load RAM into A Exclusive-OR Br with r
LQID		BF	1011 | 1111	ROM(PC_8,A,M) → Q SA → SB	None	Load Q Indirect
RMB	0	4C	0100 | 1100	0 → $RAM(B)_0$	None	Reset RAM Bit
	1	45	0100 | 0101	0 → $RAM(B)_1$		
	2	42	0100 | 0010	0 → $RAM(B)_2$		
	3	43	0100 | 0011	0 → $RAM(B)_3$		
SMB	0	4D	0100 | 1101	1 → $RAM(B)_0$	None	Set RAM Bit
	1	47	0100 | 0111	1 → $RAM(B)_1$		
	2	46	0100 | 0110	1 → $RAM(B)_2$		
	3	4B	0100 | 1011	1 → $RAM(B)_3$		
STII	y	7–	0111 | y	y → RAM(B) Bd + 1 → Bd	None	Store Memory Immediate and Increment Bd
X	r	−6	00 | r | 0110	RAM(B) ⟷ A Br ⊕ r → Br	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	3,15	23	0010 | 0011	RAM(3,15) ⟷ A	None	Exchange A with RAM (3,15)
		BF	1011 | 1111			

Instruction Set (Continued)

TABLE III. COP413C Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
MEMORY REFERENCE INSTRUCTIONS (Continued)						
XDS	r	−7	00 | r | 0111	RAM(B) ⟷ A Bd − 1 → Bd Br ⊕ r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd Exclusive-OR Br with r
XIS	r	−4	00 | r | 0100	RAM(B) ⟷ A Bd + 1 → Bd Br ⊕ r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd Exclusive-OR Br with r
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	0101 | 0000	A → Bd	None	Copy A to Bd
CBA		4E	0100 | 1110	Bd → A	None	Copy Bd to A
LBI	r,d	–	00 | r | (d - 1) (d = 0,9:15)	r,d → B	Skip until not a LBI	Load B Immediate with r,d
LEI	y	33 6–	0011 | 0011 0010 | y	y → EN	None	Load EN Immediate
TEST INSTRUCTIONS						
SKC		20	0010 | 0000		C = "1"	Skip if C is True
SKE		21	0010 | 0001		A = RAM(B)	Skip if A Equals RAM
SKGZ		33 21	0011 | 0011 0010 | 0001		$G_{3:0} = 0$	Skip if G is Zero (all 4 bits)
SKGBZ		33	0011 | 0011	1st byte		Skip if G Bit is Zero
	0	01	0000 | 0001	2nd byte	$G_0 = 0$	
	1	11	0001 | 0001	2nd byte	$G_1 = 0$	
	2	03	0000 | 0011	2nd byte	$G_2 = 0$	
	3	13	0010 | 0011	2nd byte	$G_3 = 0$	
SKMBZ	0	01	0000 | 0001		$RAM(B)_0 = 0$	Skip if RAM Bit is Zero
	1	11	0001 | 0001		$RAM(B)_1 = 0$	
	2	03	0000 | 0011		$RAM(B)_2 = 0$	
	3	13	0001 | 0011		$RAM(B)_3 = 0$	
INPUT/OUTPUT INSTRUCTIONS						
ING		33 2A	0011 | 0011 0010 | 1010	G → A	None	Input G Ports to A
INL		33 2E	0011 | 0011 0010 | 1110	$L_{7:4}$ → RAM(B) $L_{3:0}$ → A	None	Input L Ports to RAM, A
OMG		33 3A	0011 | 0011 0011 | 1010	RAM(B) → G	None	Output RAM to G Ports
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO

Note 1: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 2: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Description of Selected Instructions

The following information is provided to assist the user in understanding the operation of several unique instructions and to provide notes useful to programmers in writing COP413C programs.

XAS INSTRUCTION

XAS (Exchange A with SIO) exchanges the 4-bit contents of the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. An XAS instruction will also affect the SK output. (See Functional Description, EN Register.) If SIO is selected as a shift register, an XAS instruction must be performed once every four instruction cycle times to effect a continuous data stream.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower eight bits of the ROM address register PC with the contents of ROM addressed by the 9-bit word, PC_8, A, M. PC_8 is not affected by this instruction.

Note: JID uses two instruction cycles if executed, one if skipped.

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 9-bit word PC_8, A, M. LQID can be used for table look-up or code conversion such as BCD to 7-segment. The LQID instruction "pushes" the stack (PC + 1 $\rightarrow$ SA $\rightarrow$ SB) and replaces the least significant eight bits of the PC as follows: A $\rightarrow$ $PC_{7:4}$, RAM(B) $\rightarrow$ $PC_{3:0}$, leaving PC_8 unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SB $\rightarrow$ SA $\rightarrow$ PC), restoring the saved value of the PC to continue sequential program execution. Since LQID pushes SA $\rightarrow$ SB, the previous contents of SB are lost.

Note: LQID uses two instruction cycles if executed, one if skipped.

INSTRUCTION SET NOTES

a. The first word of a COP413C program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, one instruction cycle time is devoted to skipping each byte of the skipped instruction. Thus all program paths take the same number of cycle times whether instructions are skipped or executed (except JID and LQID).

c. The ROM is organized into eight pages of 64 words each. The program counter is a 9-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID, or LQID instruction is located in the last word of a page, the instruction operates as if it were in the next page. For example: A JP located in the last word of a page will jump to a location in the next page. Also, a LQID or JID located in the last word in page 3 or 7 will access data in the next group of four pages.

COPS Programming Manual

For detailed information on writing. COPS programs, the COPS Programming Manual 424410284-001 provides an in-depth discussion of the COPS architecture, instruction set and general techniques of COPS programming. This manual is written with the programmer in mind.

OPTION LIST—OSCILLATOR SELECTION

The oscillator option selected must be sent in with the EPROM of ROM Code for masking into the COP413C. Select the appropriate option, make a photocopy of the table and send it with the EPROM.

COP413C/COP313C

Option 1: Oscillator selection

= 0 Ceramic Resonator input frequency divided by 8. CKO is oscillator output.

= 1 Single pin RC controlled oscillator divided by 4. CKO is no connection.

Note: The following option information is to be sent to National along with the EPROM.

Option 1: Value = ___ is Oscillator Selected.

National Semiconductor Corporation

COP414L/COP314L/COP214L Single-Chip N-Channel Microcontrollers

General Description

The COP414L Single-Chip N-Channel Microcontrollers are members of the COPS™ family, fabricated using N-channel, silicon gate MOS technology. This Controller Oriented Processor is a complete microcomputer containing all system timing, internal logic, ROM, RAM and I/O necessary to implement dedicated control functions in a variety of applications. Features include single supply operation, a variety of output configuration options, with an instruction set, internal architecture and I/O scheme designed to facilitate keyboard input, display output and BCD data manipulation. The COP414L is an appropriate choice for use in numerous human interface control environments. Standard test procedures and reliable high-density fabrication techniques provide the medium to large volume customers with a customized Controller Oriented Processor at a low end-product cost.

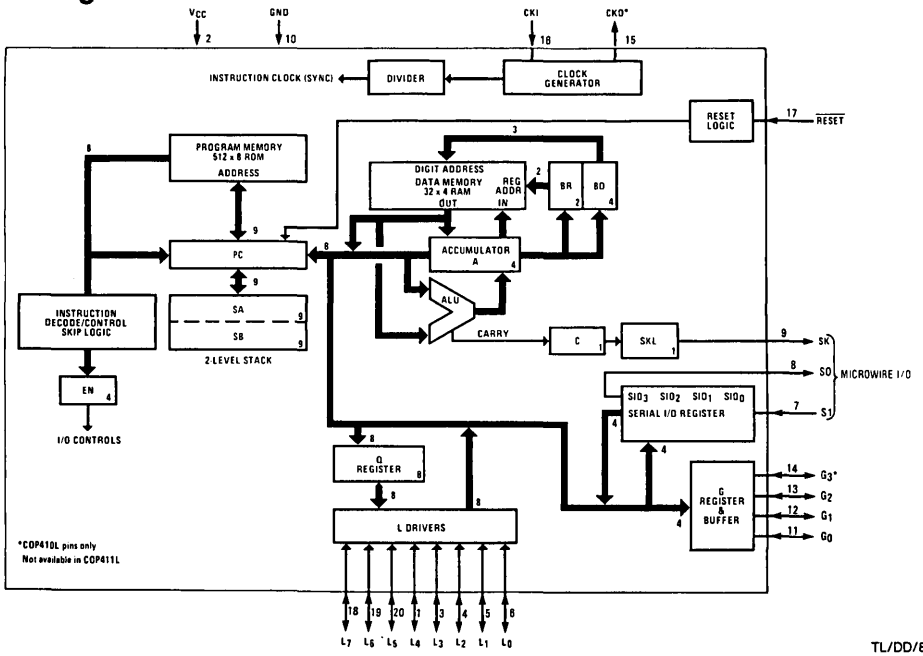
The COP314L and COP214L are exact functional equivalents but extended temperature versions of COP414L.

The COP414L can be emulated by the COP404C.

Features

- Late waferfab programming of ROM and I/O for fast delivery of units
- Low cost
- Powerful instruction set
- 512 x 8 ROM, 32 x 4 RAM
- 15 I/O lines (COP410L)
- Two-level subroutine stack
- 16 μs instruction time
- Single supply operation (4.5V–6.3V)
- Low current drain (6 mA max)
- Internal binary counter register with MICROWIRE™ serial I/O capability
- General purpose and TRI-STATE® outputs
- LSTTL/CMOS compatible in and out
- Inputs and open drain outputs able to withstand 15V
- Software/hardware compatible with other members of COP400 family
- Extended temperature range device
  — COP314L (−40°C to +85°C)
  — COP214L (−40°C to +110°C)
- Wider supply range (4.5V–9.5V) optionally available

Block Diagram



TL/DD/8814-1

FIGURE 1

COP414L

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin Relative to GND	−0.5V to +10V
Ambient Operating Temperature	0°C to +70°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C
Power Dissipation	
COP414L	0.65W at 25°C 0.3W at 70°C
Total Source Current	120 mA
Total Sink Current	100 mA

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 9.5V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Standard Operating Voltage (V_{CC})	(Note 1)	4.5	6.3	V
Optional Operating Voltage (V_{CC})		4.5	9.5	V
Power Supply Ripple	Peak to Peak		0.5	V
Operating Supply Current	All Inputs and Outputs Open		6	mA
Input Voltage Levels				
CKI Input Levels				
Ceramic Resonator Input (÷8)				
Logic High (V_{IH})	V_{CC} = Max	3.0		V
Logic High (V_{IH})	V_{CC} = 5V ±5%	2.0		
Logic Low (V_{IL})		−0.3	0.4	V
Schmitt Trigger Input (÷4)				
Logic High (V_{IH})		0.7 V_{CC}		V
Logic Low (V_{IL})		−0.3	0.6	V
$\overline{\text{RESET}}$ Input Levels	(Schmitt Trigger Input)			
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.6	V
SO Input Level (Test Mode)	(Note 2)	2.0	2.5	V
All Other Inputs				
Logic High	V_{CC} = Max	3.0		V
Logic High	With TTL Trip Level Options	2.0		V
Logic Low	Selected, V_{CC} = 5V ±5%	−0.3	0.8	V
Logic High	With High Trip Level Options	3.6		V
Logic Low	Selected	−0.3	1.2	V
Input Capacitance			7	pF
Hi-Z Input Leakage	$V_{IN} < 9.5V$	−1	+1	µA
	$9.5V \le V_{IN} \le 15V$	−10	+10	µA
Output Voltage Levels				
LSTTL Operation	V_{CC} = 5V ±10%			
Logic High (V_{OH})	I_{OH} = −25 µA	2.7		V
Logic Low (V_{OL})	I_{OL} = 0.36 mA		0.4	V
CMOS Operation (Note 3)				
Logic High	I_{OH} = −10 µA	V_{CC} − 1		V
Logic Low	I_{OL} = +10 µA		0.2	V

Note 1: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Note 2: SO output "0" level must be less than 0.8V for normal operation.

Note 3: TRI-STATE® and LED configurations are excluded.

COP414L

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 9.5V$ unless otherwise noted (Continued)

Parameter	Conditions	Min	Max	Units
Output Current Levels				
Output Sink Current				
SO and SK Ouputs (I_{OL})	$V_{CC} = 9.5V, V_{OL} = 0.4V$	1.8		mA
	$V_{CC} = 6.3V, V_{OL} = 0.4V$	1.2		mA
	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.9		mA
L_0–L_7 Outputs, G_0–G_3 and	$V_{CC} = 9.5V, V_{OL} = 0.4V$	0.4		mA
LSTTL D_0–D_3 Outputs (I_{OL})	$V_{CC} = 6.3V, V_{OL} = 0.4V$	0.4		mA
	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.4		mA
CKI (Single-pin RC Oscillator)	$V_{CC} = 4.5, V_{IH} = 3.5V$	2		mA
CKO	$V_{CC} = 4.5, V_{OL} = 0.4V$	0.2		mA
Output Source Current				
Standard Configuration,	$V_{CC} = 9.5V, V_{OH} = 2.0V$	−140	−800	μA
All Outputs (I_{OH})	$V_{CC} = 6.3V, V_{OH} = 2.0V$	−75	−480	μA
	$V_{CC} = 4.5V, V_{OH} = 2.0V$	−30	−250	μA
Push-Pull Configuration	$V_{CC} = 9.5V, V_{OH} = 4.75V$	−1.4		mA
SO and SK Outputs (I_{OH})	$V_{CC} = 6.3V, V_{OH} = 2.4V$	−1.4		mA
	$V_{CC} = 4.5V, V_{OH} = 1.0V$	−1.2		mA
Input Load Source Current	$V_{CC} = 5.0V, V_{IL} = 0V$	−10	−140	μA
Open Drain Output Leakage	$V_{OUT} < 9.5V$	−2.5	+2.5	μA
	$9.5V \le V_{OUT} \le 15V$	−25	+25	
Total Sink Current Allowed				
All Outputs Combined			100	mA
D Port			100	mA
L_7–L_4, G Port			4	mA
L_3–L_0			4	mA
Any Other Pin			2.0	mA
Total Source Current Allowed				
All I/O Combined			120	mA
L_7–L_4			60	mA
L_3–L_0			60	mA
Each L Pin			25	mA
Any Other Pin			1.5	mA

COP314L/COP214L

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin Relative to GND	−0.5V to +10V
Ambient Operating Temperature	−40°C to +85°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 seconds)	300°C
Power Dissipation COP314L/COP214L	0.65W at 25°C 0.20W at 85°C
Total Source Current	120 mA
Total Sink Current	100 mA

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics

COP214L: $-40°C \le T_A \le +110°C$, $4.5V \le V_{CC} \le 7.5V$ unless otherwise noted

COP314L: $-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 7.5V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Standard Operating Voltage (V_{CC})	(Note 1)	4.5	5.5	V
Optional Operating Voltage (V_{CC})		4.5	7.5	V
Power Supply Ripple	Peak to Peak		0.5	V
Operating Supply Current	All Inputs and Outputs Open		8	mA
Input Voltage Levels				
Ceramic Resonator Input (÷8)				
Crystal Input				
Logic High (V_{IH})	V_{CC} = Max	3.0		
Logic High (V_{IH})	V_{CC} = 5V ±5%	2.2		V
Logic Low (V_{IL})		−0.3	0.3	V
Schmitt Trigger Input (÷4)				
Logic High (V_{IH})		0.7 V_{CC}		V
Logic Low (V_{IL})		−0.3	0.4	V
$\overline{\text{RESET}}$ Input Levels	(Schmitt Trigger Input)			
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.4	V
SO Input Level (Test Mode)	(Note 2)	2.2	2.5	V
All Other Inputs				
Logic High	V_{CC} = Max	3.0		V
Logic High	With TTL Trip Level Options	2.2		V
Logic Low	Selected, V_{CC} = 5V ±5%	−0.3	0.6	V
Logic High	With High Trip Level Options	3.6		V
Logic Low	Selected	−0.3	1.2	V
Input Capacitance			7	pF
Hi-Z Input Leakage	$V_{IN} < 7.5V$	−2	+2	µA
	$7.5V \le V_{IN} \le 15V$	−20	+20	µA
Output Voltage Levels				
LSTTL Operation	V_{CC} = 5V ±10%			
Logic High (V_{OH})	I_{OH} = −20 µA	2.7		V
Logic Low (V_{OL})	I_{OL} = 0.36 mA		0.4	V
CMOS Operation (Note 3)				
Logic High	I_{OH} = −10 µA	V_{CC} − 1		V
Logic Low	I_{OL} = +10 µA		0.2	V

Note 1: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Note 2: SO output "0" level must be less than 0.6V for normal operation.

Note 3: TRI-STATE and LED configurations are excluded.

COP314L/COP214L

DC Electrical Characteristics (Continued)

COP214L: $-40°C \le T_A \le +110°C$, $4.5V \le V_{CC} \le 7.5V$ unless otherwise noted

COP314L: $-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 7.5V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Output Current Levels				
Output Sink Current				
SO and SK Outputs(I_{OL})	$V_{CC} = 7.5V$, $V_{OL} = 0.4V$	1.4		mA
	$V_{CC} = 5.5V$, $V_{OL} = 0.4V$	1.0		mA
	$V_{CC} = 4.5V$, $V_{OL} = 0.4V$	0.8		mA
L_0–L_7 Outputs, G_0–G_3 and LSTTL, D_0–D_3 Outputs (I_{OL})	$V_{CC} = 7.5V$, $V_{OL} = 0.4V$	0.4		mA
	$V_{CC} = 5.5V$, $V_{OL} = 0.4V$	0.4		mA
	$V_{CC} = 4.5V$, $V_{OL} = 0.4V$	0.4		mA
CKI (Single-pin RC Oscillator)	$V_{CC} = 4.5V$, $V_{IH} = 3.5V$	1.5		mA
CKO	$V_{CC} = 4.5V$, $V_{OL} = 0.4V$	0.2		mA
Output Source Current				
Standard Configuration, All Outputs (I_{OH})	$V_{CC} = 7.5V$, $V_{OH} = 2.0V$	−100	−900	μA
	$V_{CC} = 5.5V$, $V_{OH} = 2.0V$	−55	−600	μA
	$V_{CC} = 4.5V$, $V_{OH} = 2.0V$	−28	−350	μA
Push-Pull Configuration SO and SK Outputs (I_{OH})	$V_{CC} = 7.5V$, $V_{OH} = 3.75V$	−0.85		mA
	$V_{CC} = 5.5V$, $V_{OH} = 2.0V$	−1.1		mA
	$V_{CC} = 4.5V$, $V_{OH} = 1.0V$	−1.2		mA
Input Load Source Current	$V_{CC} = 5.0V$, $V_{IL} = 0V$	−10	−200	μA
Open Drain Output Leakage	$V_{OUT} < 7.5V$	−5	+5	μA
	$7.5V \le V_{OUT} \le 15V$	−50	+50	μA
Total Sink Current Allowed				
All Outputs Combined			100	mA
D Port			100	mA
L_7–L_4, G Port			4	mA
L_3–L_0			4	mA
Any Other Pins			1.5	mA
Total Source Current Allowed				
All I/O Combined			120	mA
L_7–L_4			60	mA
L_3–L_0			60	mA
Each L Pin			25	mA
Any Other Pins			1.5	mA

AC Electrical Characteristics

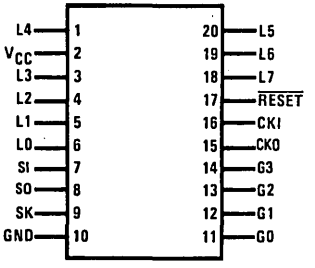
COP414L: 0°C ≤ T_A ≤ 70°C, 4.5V ≤ V_{CC} ≤ 9.5V unless otherwise noted
COP314L: −40°C ≤ T_A ≤ +85°C, 4.5V ≤ V_{CC} ≤ 7.5V unless otherwise noted
COP214L: −40°C ≤ T_A ≤ +110°C, 4.5V ≤ V_{CC} ≤ 7.5V unless otherwise noted

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time — t_C		16	40	μs
CKI				
Input Frequency — f_I	÷8 Mode	0.2	0.5	MHz
	÷4 Mode	0.1	0.25	MHz
Duty Cycle		30	60	%
Rise Time	f_I = 0.5 MHz		500	ns
Fall Time			200	ns
CKI Using RC (÷4)	R = 56 kΩ ±5% C = 100 pF ±10%			
Instruction Cycle Time (Note 1)		16	28	μs
CKO as SYNC Input				
t_{SYNC}		400		ns
Inputs				
G_3–G_0, L_7–L_0				
t_{SETUP}		8.0		μs
t_{HOLD}		1.3		μs
SI				
t_{SETUP}		2.0		μs
t_{HOLD}		1.0		μs
Output Propagation Delay	Test Condition: C_L = 50 pF, R_L = 20 kΩ, V_{OUT} = 1.5V			
SO, SK Outputs				
t_{pd1}, t_{pd0}			4.0	μs
All Other Outputs				
t_{pd1}, t_{pd0}			5.6	μs

Note 1: Variation due to the device included.

Connection Diagram

Dual-In-Line Package



TL/DD/8814–2

Top View

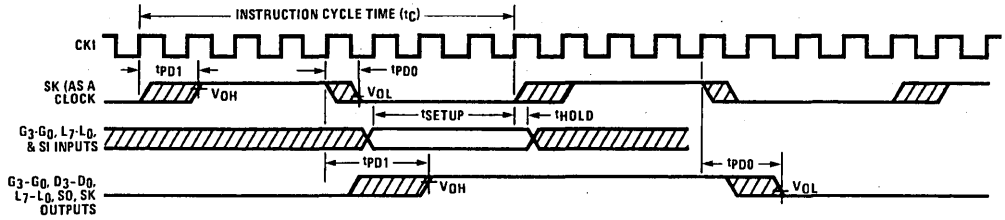
Order Number COP414L/M or N, COP314L/M or N, or COP214L/M or N
See NS Package Number M20B or N20A

FIGURE 2

Pin Descriptions

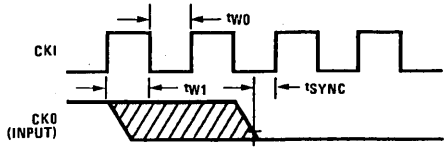
Pin	Description	Pin	Description
L_7–L_0	8 bidirectional I/O ports with TRI-STATE	SK	Logic-controlled clock (or general purpose output)
G_3–G_0	4 bidirectional I/O ports (G_2–G_0 for COP411L)	CKI	System oscillator input
SI	Serial input (or counter input)	CKO	System oscillator output
SO	Serial output (or general purpose output)	RESET	System reset input
		V_{CC}	Power supply
		GND	Ground

Timing Diagrams



TL/DD/8814-3

FIGURE 3. Input/Output Timing Diagrams (Ceramic Resonator Divide-by-8 Mode)



TL/DD/8814-4

FIGURE 3a. Synchronization Timing

Functional Description

A block diagram of the COP414L is given in *Figure 1*. Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1" (greater than 2V). When a bit is reset, it is a logic "0" (less than 0.8V).

All functional references to the COP414L also apply to the COP314L.

PROGRAM MEMORY

Program Memory consists of a 512-byte ROM. As can be seen by an examination of the COP414L instruction set, these words may be program instructions, program data or ROM addressing data. Because of the special characteristics associated with the JP, JSRP, JID and LQID instructions, ROM must often be thought of as being organized into 8 pages of 64 words each.

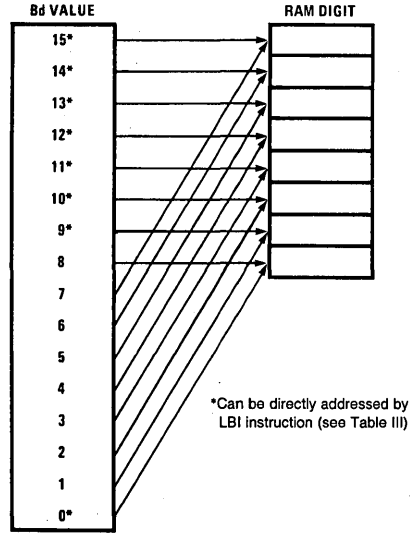
ROM addressing is accomplished by a 9-bit PC register. Its binary value selects one of the 512 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 9-bit binary count value. Two levels of subroutine nesting are implemented by the 9-bit subroutine save registers, SA and SB, providing a last-in, first-out (LIFO) hardware subroutine stack.

ROM instruction words are fetched, decoded and executed by the Instruction Decode, Control and Skip Logic circuitry.

DATA MEMORY

Data memory consists of a 128-bit RAM, organized as 4 data registers of 8 4-bit digits. RAM addressing is implemented by a 6-bit B register whose upper 2 bits (Br) select 1 of 4 data registers and lower 3 bits of the 4-bit Bd select 1 of 8 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) is usually loaded into or from, or exchanged with, the A register (accumulator), it may also be loaded into the Q latches or loaded from the L ports. RAM addressing may also be performed directly by the XAD 3,15 instruction.

The most significant bit of Bd is not used to select a RAM digit. Hence each physical digit of RAM may be selected by two different values of Bd as shown in *Figure 4* below. The skip condition for XIS and XDS instructions will be true if Bd changes between 0 and 15, but NOT between 7 and 8 (see Table III).



TL/DD/8814-5

FIGURE 4. RAM Digit Address to Physical RAM Digit Mapping

Functional Description (Continued)

INTERNAL LOGIC

The 4-bit A register (accumulator) is the source and destination register for most I/O, arithmetic, logic and data memory access operations. It can also be used to load the Bd portion of the B register, to load 4 bits of the 8-bit Q latch data, to input 4 bits of the 8-bit L I/O port data and to perform data exchanges with the SIO register.

A 4-bit adder performs the arithmetic and logic functions of the COP410L/411L, storing its results in A. It also outputs a carry bit to the 1-bit C register, most often employed to indicate arithmetic overflow. The C register, in conjunction with the XAS instruction and the EN register, also serves to control the SK output. C can be outputted directly to SK or can enable SK to be a sync clock each instruction cycle time. (See XAS instruction and EN register description, below.)

The G register contents are outputs to 4 general-purpose bidirectional I/O ports.

The Q register is an internal, latched, 8-bit register, used to hold data loaded from M and A, as well as 8-bit data from ROM. Its contents are output to the L I/O ports when the L drivers are enabled under program control. (See LEI instruction.)

The 8 L drivers, when enabled, output the contents of latched Q data to the L I/O ports. Also, the contents of L may be read directly into A and M. L I/O ports can be directly connected to the segments of a multiplexed LED display (using the LED Direct Drive output configuration option) with Q data being outputted to the Sa–Sg and decimal point segments of the display.

The SIO register functions as a 4-bit serial-in serial-out shift register or as a binary counter depending on the contents of the EN register. (See EN register description, below.) Its contents can be exchanged with A, allowing it to input or output a continuous serial data stream. SIO may also be used to provide additional parallel I/O by connecting SO to external serial-in/parallel-out shift registers.

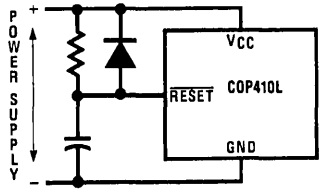
The XAS instruction copies C into the SKL Latch. In the counter mode, SK is the output of SKL in the shift register mode, SK outputs SKL ANDed with internal instruction cycle clock.

The EN register is an internal 4-bit register loaded under program control by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register (EN_3–EN_0).

1. The least significant bit of the enable register, EN_0, selects the SIO register as either a 4-bit shift register or a 4-bit binary counter. With EN_0 set, SIO is an asynchronous binary counter, *decrementing* its value by one upon each low-going pulse ("1" to "0") occuring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output is equal to the value of EN_3. With EN_0 reset, SIO is a serial shift register shifting left each instruction cycle time. The data present at SI goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. (See 4 below.) The SK output becomes a logic-controlled clock.
2. EN_1 is not used. It has no effect on COP414L operation.
3. With EN_2 set, the L drivers are enabled to output the data in Q to the L I/O ports. Resetting EN_2 disables the L drivers, placing the L I/O ports in a high-impedance input state.
4. EN_3, in conjunction with EN_0, affects the SO output. With EN_0 set (binary counter option selected) SO will output the value loaded into EN_3. With EN_0 reset (serial shift register option selected), setting EN_3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN_3 with the serial shift register option selected disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains reset to "0". Table I provides a summary of the modes associated with EN_3 and EN_0.

INITIALIZATION

The Reset Logic will initialize (clear) the device upon power-up if the power supply rise time is less than 1 ms and greater than 1 μs. If the power supply rise time is greater than 1 ms, the user must provide an external RC network and diode to the $\overline{\text{RESET}}$ pin as shown below *(Figure 5)*. The $\overline{\text{RESET}}$ pin is configured as a Schmitt trigger input. If not used it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the $\overline{\text{RESET}}$ input, provided it stays low for at least three instruction cycle times.



RC ≥ 5 × Power Supply Rise Time TL/DD/8814–6

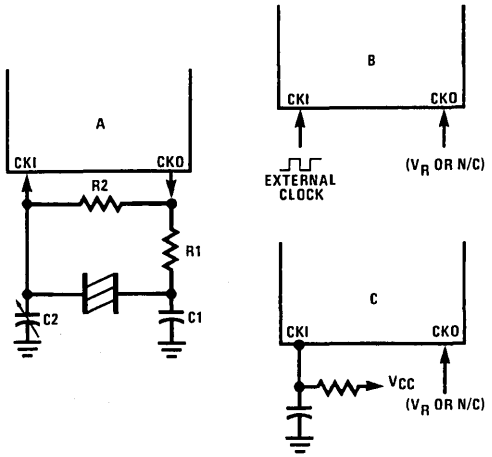
FIGURE 5. Power-Up Clear Circuit

TABLE I. Enable Register Modes—Bits EN_3 and EN_0

EN_3	EN_0	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL = 1, SK = Clock If SKL = 0, SK = 0
1	0	Shift Register	Input to Shift Register	Serial Out	If SKL = 1, SK = Clock If SKL = 0, SK = 0
0	1	Binary Counter	Input to Binary Counter	0	If SKL = 1, SK = 1 If SKL = 0, SK = 0
1	1	Binary Counter	Input to Binary Counter	1	If SKL = 1, SK = 1 If SKL = 0, SK = 0

Functional Description (Continued)

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN and G registers are cleared. The SK output is enabled as a SYNC output, providing a pulse each instruction cycle time. *Data Memory (RAM) is not cleared upon initialization.* The first instruction at address 0 must be a CLRA.



TL/DD/8814–7

Ceramic Resonator Oscillator

Resonator Value	Components Values			
	R1 (Ω)	R2 (Ω)	C1 (pF)	C2 (pF)
455 kHz	4.7k	1M	220	220

RC Controlled Oscillator

R (kΩ)	C (pF)	Instruction Cycle Time In μs
51	100	19 ±15%
82	56	10 ±13%

Note: 200 kΩ ≥ R ≥ 25 kΩ. 360 pF ≥ C ≥ 50 pF. Does not include tolerances.

FIGURE 6. COP414L Oscillator

OSCILLATOR

There are four basic clock oscillator configurations available as shown by *Figure 6*.

a. **Resonator Controlled Oscillator.** CKI and CKO are connected to an external ceramic resonator. The instruction cycle frequency equals the resonator frequency divided by 8.

b. **External Oscillator.** CKI is an external clock input signal. The external frequency is divided by 8 to give the instruction frequency time. CKO is no connection.

c. **RC Controlled Oscillator.** CKI is configured as a single pin RC controlled Schmitt trigger oscillator. The instruction cycle equals the oscillation frequency divided by 4. CKO is no connection.

CKO PIN OPTIONS

In a resonator controlled oscillator system, CKO is used as an output to the resonator network. CKO is no connection for External or RC controlled oscillator.

I/O OPTIONS

COP414 inputs and outputs have the following optional configurations, illustrated in *Figure 7*:

a. **Standard**—an enhancement-mode device to ground in conjunction with a depletion-mode device to V_{CC}, compatible with LSTTL and CMOS input requirements. Available on SO, SK and all D and G outputs.

b. **Open-Drain**—an enhancement-mode device to ground only, allowing external pull-up as required by the user's application. Available on SO, SK and all D and G outputs.

c. **Push-Pull**—an enhancement-mode device to ground in conjunction with a depletion-mode device paralleled by an enhancement-mode device to V_{CC}. This configuration has been provided to allow for fast rise and fall times when driving capacitive loads. Available on SO and SK outputs only.

d. **Standard L**—same as **a.**, but may be disabled. Available on L outputs only.

e. **Open Drain L**—same as **b.**, but may be disabled. Available on L outputs only.

f. An on-chip depletion load device to V_{CC}.

g. A Hi-Z input which must be driven to a "1" or "0" by external components.

The above input and output configurations share common enhancement-mode and depletion-mode devices. Specifically, all configurations use one or more of six devices (numbered 1–6, respectively). Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figure 8* for each of these devices to allow the designer to effectively use these I/O configurations in designing a COP414 system.

The SO, SK outputs can be configured as shown in **a.**, **b.**, or **c.** The G outputs can be configured as shown in **a.** or **b.** Note that when inputting data to the G ports, the G outputs should be set to "1". The L outputs can be configured as in **d.**

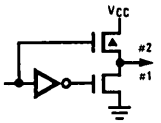
An important point to remember if using configuration **d.** with the L drivers is that even when the L drivers are disabled, the depletion load device will source a small amount of current. (See *Figure 8*, device 2.) However, when the L port is used as input, the disabled depletion device CANNOT be relied on to source sufficient current to pull an input to a logic "1".

COP411L

If the COP410L is bonded as a 20-pin device, it becomes the COP411L, illustrated in *Figure 2,* COP410L/411L Connection Diagrams. Note that the COP411L does not contain D2, D3, G3 or CKO. Use of this option of course precludes use of D2, D3, G3 and CKO options. All other options are available for the COP411L.

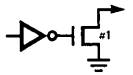
Functional Description (Continued)

a. Standard Output



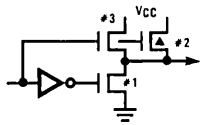
TL/DD/8814-8

b. Open-Drain Output



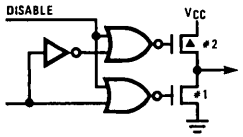
TL/DD/8814-9

c. Push-Pull Output



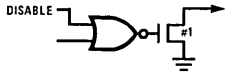
TL/DD/8814-10

d. Standard L Output



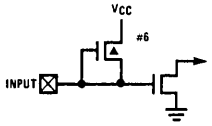
TL/DD/8814-11

e. Open-Drain L Output



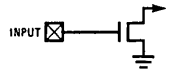
TL/DD/8814-12

f. Input with Load



TL/DD/8814-13

g. Hi-Z Input

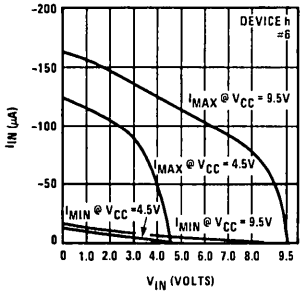


TL/DD/8814-14

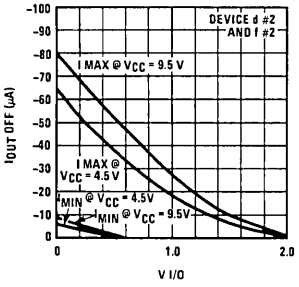
FIGURE 7. Input and Output Configurations

Typical Performance Curves

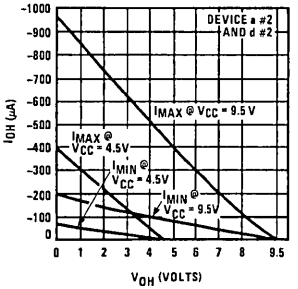
Input Current RESET, SI



Input Current for L0 through L7 when Output Programmed Off by Software

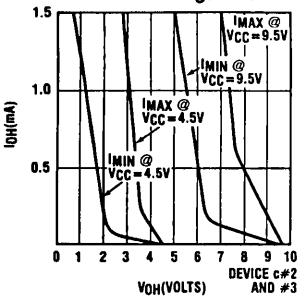


Source Current for Standard Output Configuration

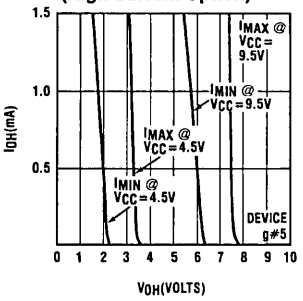


TL/DD/8814-15

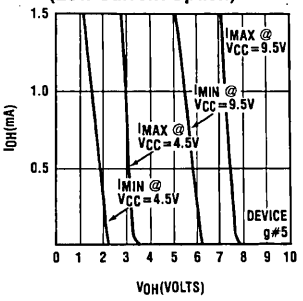
Source Current for SO and SK in Push-Pull Configuration



Source Current for L0 through L7 in TRI-STATE Configuration (High Current Option)



Source Current for L0 through L7 in TRI-STATE Configuration (Low Current Option)

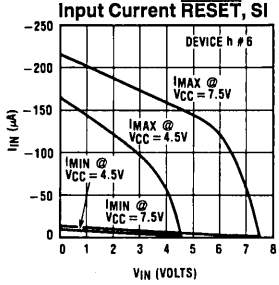


TL/DD/8814-16

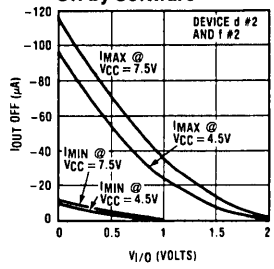
FIGURE 8a. COP414 I/O DC Current Characteristics

Typical Performance Curves (Continued)

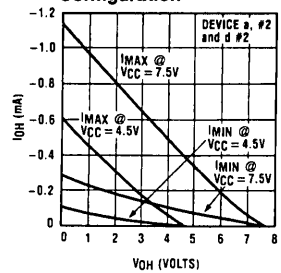
Input Current $\overline{\text{RESET}}$, SI



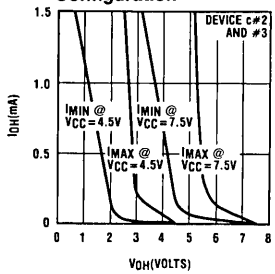
Input Current for L0–L7 when Output Programmed Off by Software



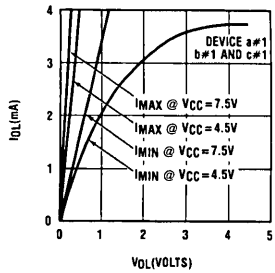
Source Current for Standard Output Configuration



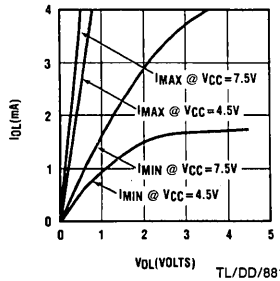
Source Current for SO and SK in Push-Pull Configuration



Output Sink Current for SO and SK



Output Sink Current for L0–L7 and Standard Drive Option for G0–G3



TL/DD/8814–17

FIGURE 8b. COP314L/COP214L Input/Output Characteristics

COP414 Instruction Set

Table II is a symbol table providing internal architecture, instruction operand and operational symbols used in the instruction set table.

Table III provides the mnemonic, operand, machine code, data flow, skip conditions and description associated with each instruction in the COP410L/411L instruction set.

TABLE II. COP414L Instruction Set Table Symbols

Symbol	Definition
INTERNAL ARCHITECTURE SYMBOLS	
A	4-bit Accumulator
B	6-bit RAM Address Register
Br	Upper 2 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry Register
D	4-bit Data Output Port
EN	4-bit Enable Register
G	4-bit Register to latch data for G I/O Port
L	8-bit TRI-STATE I/O Port
M	4-bit contents of RAM Memory pointed to by B Register
PC	9-bit ROM Address Register (program counter)
Q	8-bit Register to latch data for L I/O Port
SA	9-bit Subroutine Save Register A
SB	9-bit Subroutine Save Register B
SIO	4-bit Shift Register and Counter
SK	Logic-Controlled Clock Output

Symbol	Definition
INSTRUCTION OPERAND SYMBOLS	
d	4-bit Operand Field, 0–15 binary (RAM Digit Select)
r	2-bit Operand Field, 0–3 binary (RAM Register Select)
a	9-bit Operand Field, 0–511 binary (ROM Address)
y	4-bit Operand Field, 0–15 binary (Immediate Data)
RAM(s)	Contents of RAM location addressed by s
ROM(t)	Contents of ROM location addressed by t
OPERATIONAL SYMBOLS	
+	Plus
−	Minus
→	Replaces
↔	Is exchanged with
=	Is equal to
$\overline{A}$	The one's complement of A
⊕	Exclusive-OR
:	Range of values

TABLE III. COP414L Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	|0011|0000|	A + C + RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	|0011|0001|	A + RAM(B) → A	None	Add RAM to A
AISC	y	5–	|0101| y |	A + y → A	Carry	Add Immediate, Skip on Carry (y ≠ 0)
CLRA		00	|0000|0000|	0 → A	None	Clear A
COMP		40	|0100|0000|	$\overline{A}$ → A	None	One's complement of A to A
NOP		44	|0100|0100|	None	None	No Operation
RC		32	|0011|0010|	"0" → C	None	Reset C
SC		22	|0010|0010|	"1" → C	None	Set C
XOR		02	|0000|0010|	A ⊕ RAM(B) → A	None	Exclusive-OR RAM with A

COP414L Instruction Set (Continued)

TABLE III. COP414L Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
TRANSFER OF CONTROL INSTRUCTIONS						
JID		FF	1111 | 1111	ROM (PC_8,A,M) $PC_{7:0}$	None	Jump Indirect (Note 2)
JMP	a	6– – –	0110 | 000 | a_8 $a_{7:0}$	a → PC	None	Jump
JP	a	– –	1 | $a_{6:0}$ (pages 2, 3 only) or	a → $PC_{6:0}$	None	Jump within Page (Note 3)
		– –	11 | $a_{5:0}$ (all other pages)	a → $PC_{5:0}$		
JSRP	a	– –	10 | $a_{5:0}$	PC + 1 → SA → SB 010 → $PC_{8:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 4)
JSR	a	6– – –	0110 | 100 | a_8 $a_{7:0}$	PC + 1 → SA → SB a → PC	None	Jump to Subroutine
RET		48	0100 | 1000	SB → SA → PC	None	Return from Subroutine
RETSK		49	0100 | 1001	SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
MEMORY REFERENCE INSTRUCTIONS						
CAMQ		33 3C	0011 | 0011 0011 | 1100	A → $Q_{7:4}$ RAM(B) → $Q_{3:0}$	None	Copy A, RAM to Q
COMA		23 2C	0011 | 0011 0010 | 1100	$Q_{7:4}$ → RAM(B) $Q_{3:0}$ → A	None	Copy Q to RAM, A
LD	r	–5	00 | r | 0101	RAM(B) → A Br ⊕ r → Br	None	Load RAM into A, Exclusive-OR Br with r
LQID		BF	1011 | 1111	ROM(PC_8, A, M) → Q SA → SB	None	Load Q Indirect (Note 2)
RMB	0 1 2 3	4C 45 42 43	0100 | 1100 0100 | 0101 0100 | 0010 0100 | 0011	0 → $RAM(B)_0$ 0 → $RAM(B)_1$ 0 → $RAM(B)_2$ 0 → $RAM(B)_3$	None	Reset RAM Bit
SMB	0 1 2 3	4D 47 46 4B	0100 | 1101 0100 | 0111 0100 | 0110 0100 | 1011	1 → $RAM(B)_0$ 1 → $RAM(B)_1$ 1 → $RAM(B)_2$ 1 → $RAM(B)_3$	None	Set RAM Bit
STII	y	7–	0111 | y	y → RAM(B) Bd + 1 → Bd	None	Store Memory Immediate and Increment Bd
X	r	–6	00 | r | 0110	RAM(B) ⟷ A Br ⊕ r → Br	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	3, 15	23 BF	0010 | 0011 1011 | 1111	RAM(3,15) ⟷ A	None	Exchange A with RAM (3,15)
XDS	r	–7	00 | r | 0111	RAM(B) ⟷ A Bd – 1 → Bd Br ⊕ r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd, Exclusive-OR Br with r
XIS	r	–4	00 | r | 0100	RAM(B) ⟷ A Bd + 1 → Bd Br ⊕ r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd Exclusive-OR Br with r

COP414L Instruction Set (Continued)

TABLE III. COP414L Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	0101 | 0000	A → Bd	None	Copy A to Bd
CBA		4E	0100 | 1110	Bd → A	None	Copy Bd to A
LBI	r,d	– –	00 | r | (d − 1) (d = 0,9:15)	r,d → B	Skip until not a LBI	Load B Immediate with r,d (Note 5)
LEI	y	33 6–	0011 | 0011 0010 | y	y → EN	None	Load EN Immediate (Note 6)
TEST INSTRUCTIONS						
SKC		20	0010 | 0000		C = "1"	Skip if C is True
SKE		21	0010 | 0001		A = RAM(B)	Skip if A Equals RAM
SKGZ		33 21	0011 | 0011 0010 | 0001		$G_{3:0} = 0$	Skip if G is Zero (all 4 bits)
SKGBZ		33	0011 | 0011	1st byte		Skip if G Bit is Zero
	0	01	0000 | 0001	2nd byte	$G_0 = 0$	
	1	11	0001 | 0001		$G_1 = 0$	
	2	03	0000 | 0011		$G_2 = 0$	
	3	13	0001 | 0011		$G_3 = 0$	
SKMBZ	0	01	0000 | 0001		$RAM(B)_0 = 0$	Skip if RAM Bit is Zero
	1	11	0001 | 0001		$RAM(B)_1 = 0$	
	2	03	0000 | 0011		$RAM(B)_2 = 0$	
	3	13	0001 | 0011		$RAM(B)_3 = 0$	
INPUT/OUTPUT INSTRUCTIONS						
ING		33 2A	0011 | 0011 0010 | 1010	G → A	None	Input G Ports to A
INL		33 2E	0011 | 0011 0010 | 1110	$L_{7:4}$ → RAM(B) $L_{3:0}$ → A	None	Input L Ports to RAM, A
OBD		33 3E	0011 | 0011 0011 | 1110	Bd → D	None	Output Bd to D Outputs
OMG		33 3A	0011 | 0011 0011 | 1010	RAM(B) → G	None	Output RAM to G Ports
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO (Note 2)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicitly defined (e.g., Br and Bd are explicitly defined). Bits are numbered 0 to N where 0 signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit A register.

Note 2: For additional information on the operation of the XAS, JID, and LQID instructions, see below.

Note 3: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 4: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 5: The machine code for the lower 4 bits of the LBI instruction equals the binary value of the "d" data *minus 1*, e.g., to load the lower four bits of B (Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Note 6: Machine code for operand field y for LEI instruction should equal the binary value to be latched into EN, where a "1" or "0" in each bit of EN corresponds with the selection or deselection of a particular function associated with each bit. (See Functional Description, EN Register.)

Option List

The COP414L mask-programmable options are assigned numbers which correspond with the COP414L pins.

The following is a list of COP414L options. The options are programmed at the same time as the ROM pattern to provide the user with the hardware flexibility to interface to various I/O components using little or no external circuitry.

Option 1: L_4 Driver
= 0: Standard output
= 1: Open-drain output

Option 2: V_{CC} Pin
= 0: 4.5V to 6.3V operation
= 1: 4.5V to 9.5V operation

Option 3: L_3 Driver
same as Option 1

Option 4: L_2 Driver
same as Option 1

Option 5: L_1 Driver
same as Option 1

Option 6: L_6 Driver
same as Option 1

Option 7: SI Input
= 0: load device to V_{CC}
= 1: Hi-Z Output

Option 8: SO Driver
= 0: Standard output
= 1: Open-drain output
= 2: Push-pull output

Option 9: SK Driver
same as Option 8

Option 10:
= 0: Ground Pin—no options available

Option 11: G_0 I/O Port
= 0: Standard output
= 1: Open-drain output

Option 12: G_1 I/O Port
same as Option 11

Option 13: G_2 I/O Port
same as Option 11

Option 14: G_3 I/O Port
same as Option 11

Option 15: CKO Output
= 0: Clock output to ceramic resonator
= 1: No connection

Option 16: CKI Input
= 0: Ocillator input divided by 8 (500 kHz max)
= 1: Single pin RC controlled oscillator divided by 4
= 2: External Schmitt trigger level clock divided by 4

Option 17: $\overline{\text{RESET}}$ Input
= 0: Load device to V_{CC}
= 1: Hi-Z Input

Option 18: L_7 Driver
same as Option 3

Option 19: L_6 Driver
same as Option 1

Option 20: L_6 Driver
same as Option 2

Option 21: L Input Levels
= 0: Standard TTL input levels ("0" = 0.8V, "1" = 2.0V)
= 1: Higher voltage input levels ("0" = 1.2V, "1" = 3.6V)

Option 22: G Input Levels
same as Option 21

Option 23: SI Input Levels
same as Option 21

Option 24: Internal Initialization
= 0: Power on reset enabled
= 1: Disabled

Option 25: 15V Hi-Z Inputs/Open Drain Outputs
= 0: Regular operation
= 1: Must withstand 15V

TEST MODE (NON-STANDARD OPERATION)

The SO output has been configured to provide for standard test procedures for the custom-programmed COP414L. With SO forced to logic "1", two test modes are provided, depending upon the value of SI:

a. RAM and Internal Logic Test Mode (SI = 1)

b. ROM Test Mode (SI = 0)

These special test modes should not be employed by the user; they are intended for manufacturing tests only.

COP414L Option List

Please fill out the Option List and send it with the EPROM.

Option Data

OPTION 1 VALUE = _____ IS: L_4 DRIVER
OPTION 2 VALUE = _____ IS: V_{CC} PIN
OPTION 3 VALUE = _____ IS: L_3 DRIVER
OPTION 4 VALUE = _____ IS: L_2 DRIVER
OPTION 5 VALUE = _____ IS: L_1 DRIVER
OPTION 6 VALUE = _____ IS: L_6 DRIVER
OPTION 7 VALUE = _____ IS: SI INPUT
OPTION 8 VALUE = _____ IS: SO DRIVER
OPTION 9 VALUE = _____ IS: SK DRIVER
OPTION 10 VALUE = _____ IS: 0 GROUND PIN
OPTION 11 VALUE = _____ IS: G_0 I/O PORT
OPTION 12 VALUE = _____ IS: G_1 I/O PORT
OPTION 13 VALUE = _____ IS: G_2 I/O PORT
OPTION 14 VALUE = _____ IS: G_3 I/O PORT
OPTION 15 VALUE = _____ IS: CKO OUTPUT
OPTION 16 VALUE = _____ IS: CKI INPUT
OPTION 17 VALUE = _____ IS: RESET INPUT
OPTION 18 VALUE = _____ IS: L_7 DRIVER
OPTION 19 VALUE = _____ IS: L_6 DRIVER
OPTION 20 VALUE = _____ IS: L_6 DRIVER
OPTION 21 VALUE = _____ IS: L INPUT LEVELS
OPTION 22 VALUE = _____ IS: G INPUT LEVELS
OPTION 23 VALUE = _____ IS: SI INPUT LEVELS
OPTION 24 VALUE = _____ IS: INTERNAL INITIALIZATION
OPTION 25 VALUE = _____ IS: 15V Hi-Z INPUTS/OPEN DRAIN OUTPUTS

Development Support

The MOLE (Microcontroller On Line Emulator) is a low cost development system and real time emulator for COPS' products. They also include TMP, 8050 and the new 16 bit HPC microcontroller family. The MOLE provides effective support for the development of both software and hardware in the user's applicaton.

The purpose of the MOLE is to provide a tool to write and assemble code, emulate code for the target microcontroller and assist in debugging of the system.

The MOLE can be connected to various hosts, IBM PC, STARPLEX™, Kaypro, Apple and Intel systems, via RS-232 port. This link facilitates the up loading/down loading of code, supports host assembly and mass storage.

The MOLE consists of three parts; brain, personality and optional host software.

The brain board is the computing engine of the system. It is a self-contained computer with its own firmware which provides for all system operation, emulation control, communication, from programming and diagnostic operation. It has three serial ports which can be connected to a terminal, host system, printer, modem or to other MOLE's in a multi-MOLE environment.

The personality board contains the necessary hardware and firmware needed to emulate the target microcontroller. The emulation cable which replaces the target controller attaches to this board. The software contains a cross assembler and a communications program for up loading and down loading code from the MOLE.

MOLE Ordering Information

P/N	Description
MOLE-BRAIN	MOLE Computer Board
MOLE-COPS-PB1	COPS Personality Board
MOLE-XXX-YYY	Optional Software

Where XXX = COPS

YYY = Host System, IBM, Apple, KAY (Kaypro), CP/M

COPS Programming Manual

For detailed information on writing COPS programs, the COPS Programming Manual 424410284-001 provides an in-depth discussion of the COPS architecture, instruction set and general techniques of COPS programming. This manual is written with the programmer in mind.

National Semiconductor Corporation

ADVANCED INFORMATION

microCMOS

COP417C/COP418C/COP317C/COP318C/ COP217C/COP218C Single-Chip CMOS Microcontrollers

General Description

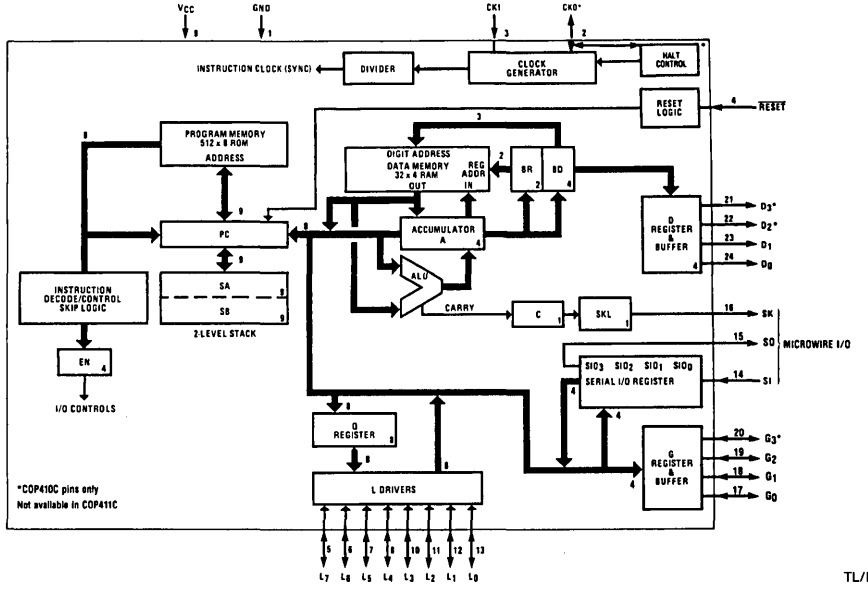
The COP417C, COP418C, COP317C, COP318C, COP217C and COP218C fully static, single-chip CMOS microcontrollers are members of the COPS™ family, fabricated using double-poly, silicon-gate CMOS technology. These controller-oriented processors are complete microcomputers containing all system timing, internal logic, ROM, RAM, and I/O necessary to implement dedicated control functions in a variety of applications. Features include single supply operation, a variety of output configuration options, with an instruction set, internal architecture, and I/O scheme designed to facilitate keyboard input, display output, and BCD data manipulation. The COP418C is identical to the COP417C but with 16 I/O lines instead of 20. They are an appropriate choice for use in numerous human interface control environments. Standard test procedures and reliable high-density fabrication techniques provide the medium to large volume customers with a customized controller-oriented processor at a low end-product cost.

The COP317C/COP318C/COP217C/COP218C is the extended temperature range version of the COP417C/ COP418C.

Features

- Lowest power dissipation (25 μW typical)
- Fully static (can turn off clock)
- Power-saving HALT mode
- 2 μs instruction cycle time, plus software selectable clocks
- 512 x 8 ROM, 32 x 4 RAM
- 18 I/O lines (COP417C)
- Two-level subroutine stack
- Single supply operation (2.4V to 6.0V)
- Internal binary counter register with MICROWIRE™ serial I/O capability
- General purpose and TRI-STATE® outputs
- Open-drain outputs and Hi-Z inputs operational at 12 volts
- Internal initialization on power-up optionally available
- External HALT mode wake-up optionally available
- LSTTL/CMOS compatible
- High-sink current optionally available
- Software/hardware compatible with COP400 family
- Extended temperature range devices COP317C/ COP318C (−40°C to +85°C)
- Military temperature range devices COP217C/COP218C (−55°C to +125°C)

Block Diagram



TL/DD/9130–1

FIGURE 1. COP417C/418C

National Semiconductor Corporation

COP420/COP421/COP422 and COP320/COP321/COP322 Single-Chip N-Channel Microcontrollers

General Description

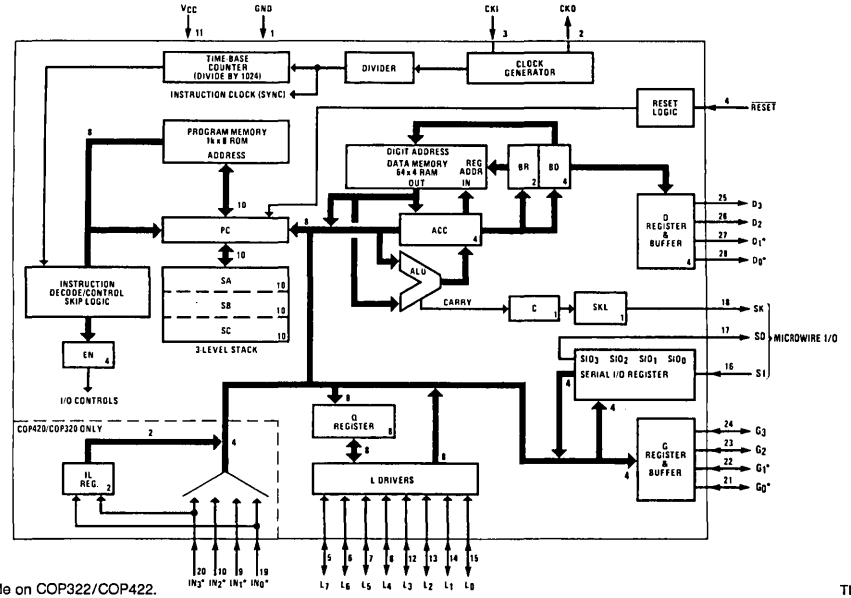
The COP420, COP421, COP422, COP320, COP321 and COP322 Single-Chip N-Channel Microcontrollers are members of the COPS™ family, fabricated using N-channel, silicon gate MOS technology. They are complete microcomputers containing all system timing, internal logic, ROM, RAM and I/O necessary to implement dedicated control functions in a variety of applications. Features include single supply operation, a variety of output configuration options, with an instruction set, internal architecture and I/O scheme designed to facilitate keyboard input, display output and BCD data manipulation. The COP421 is identical to the COP420, except with 19 I/O lines instead of 23; the COP422 has 15 I/O lines. They are an appropriate choice for use in numerous human interface control environments. Standard test procedures and reliable high-density fabrication techniques provide the medium to large volume customers with a customized Controller Oriented Processor at a low end-product cost.

The COP320 is the extended temperature range version of the COP420 (likewise the COP321 and COP322 are the extended temperature range versions of the COP421/COP422). The COP320/321/322 are exact functional equivalents of the COP420/421/422.

Features

- Low cost
- Powerful instruction set
- 1k x 8 ROM, 64 x 4 RAM
- 23 I/O lines (COP420, COP320)
- True vectored interrupt, plus restart
- Three-level subroutine stack
- 4.0 μs instruction time
- Single supply operation
- Internal time-base counter for real-time processing
- Internal binary counter register with MICROWIRE™ compatible serial I/O capacity
- General purpose and TRI-STATE® outputs
- TTL/CMOS compatible in and out
- LED direct drive outputs
- MICROBUS™ compatible
- Software/hardware compatible with other members of COP400 family
- Extended temperature range device COP320/COP321/COP322 (−40°C to +85°C)

Block Diagram



*Not available on COP322/COP422.

TL/DD/6921–1

FIGURE 1

COP420/COP421/COP422 and COP320/COP321/COP322 Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin	−0.3V to +7V
Operating Temperature Range	
COP420/COP421/COP422	0°C to 70°C
COP320/COP321/COP322	−40°C to +85°C
Storage Temperature Range	−65°C to +150°C
Total Sink Current	75 mA
Total Source Current	95 mA
Package Power Dissipation	750 mW at 25°C
24 and 28 pin	400 mW at 70°C
	250 mW at 85°C
Package Power Dissipation	650 mW at 25°C
20 pin	300 mW at 70°C
	200 mW at 85°C
Lead Temperature (soldering, 10 sec.)	300°C

Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.

COP420/COP421/COP422

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 6.3V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Operation Voltage		4.5	6.3	V
Power Supply Ripple	Peak to Peak (Note 3)		0.4	V
Supply Current	Outputs Open		38	mA
Supply Current	Outputs Open, $V_{CC} = 5V$, $T_A = 25°C$		30	mA
Input Voltage Levels				
CKI Input Levels				
Crystal Input				
Logic High	V_{CC} = Max.	3.0		
Logic High	$V_{CC} = 5V \pm 5\%$	2.0		V
Logic Low		−0.3	0.4	V
TTL Input	$V_{CC} = 5V \pm 5\%$			
Logic High		2.0		V
Logic Low		−0.3	0.8	V
Schmitt Trigger Inputs				
RESET, CKI (÷4)				
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.6	V
SO Input Level (Test Mode)	(Note 2)	2.0	3.0	V
All Other Inputs				
Logic High	V_{CC} = Max.	3.0		V
Logic High	$V_{CC} = 5V \pm 5\%$	2.0		V
Logic Low		−0.3	0.8	V
Input Levels High Trip Option				
Logic High		3.6		V
Logic Low		−0.3	1.2	V
Input Load Source Current	$V_{CC} = 5V$, $V_{IN} = 0V$			
CKO		−4	−800	μA
All Others		−100	−800	μA
Input Capacitance			7	pF
Hi-Z Input Leakage		−1	+1	μA
Output Voltage Levels				
Standard Outputs				
TTL Operation	$V_{CC} = 5V \pm 10\%$			
Logic High	$I_{OH} = -100\ \mu A$	2.4		V
Logic Low	$I_{OL} = 1.6$ mA	−0.3	0.4	V
CMOS Operation (Note 1)				
Logic High	$I_{OH} = -10\ \mu A$	$V_{CC} - 1$		V
Logic Low	$I_{OL} = +10\ \mu A$		0.2	V

Note 1: TRI-STATE and LED configurations are excluded.

Note 2: SO output "0" level must be less than 0.8V for normal operation.

COP420/COP421/COP422

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 6.3V$ unless otherwise noted (Continued)

Parameter	Conditions	Min	Max	Units
Output Current Levels				
LED Direct Drive Output	$V_{CC} = 6V$			
Logic High	$V_{OH} = 2.0V$	2.5	14	mA
CKI Sink Current (R/C Option)	$V_{IN} = 3.5V$	2		mA
CKO (RAM Supply Current)	$V_R = 3.3V$		3	mA
TRI-STATE or Open Drain Leakage Current	$V_{CC} = 5V$	−2.5	+2.5	μA
Output Current Levels				
Output Sink Current (I_{OL}) Output Source Current (I_{OH})	$V_{CC} = 4.5V$, $V_{OL} = 0.4V$	+1.6		mA
Standard Configuration				
All Outputs	$V_{CC} = 6.3V$, $V_{OH} = 3.0V$	−200	−900	μA
	$V_{CC} = 4.5V$, $V_{OH} = 2.0V$	−100	−500	μA
Push-Pull Configuration				
SO, SK Outputs	$V_{CC} = 6.3V$, $V_{OH} = 3.0V$	−1.0		mA
	$V_{CC} = 4.5V$, $V_{OH} = 2.0V$	−0.4		mA
TRI-STATE Configuration				
L_0–L_7 Outputs	$V_{CC} = 6.3V$, $V_{OH} = 3.2V$	−0.8		mA
	$V_{CC} = 4.5V$, $V_{OH} = 1.5V$	−0.9		mA
LED Configuration				
L_0–L_7 Outputs	$V_{CC} = 6.3V$, $V_{OH} = 3.0V$	−1.0		mA
	$V_{CC} = 4.5V$, $V_{OH} = 2.0V$	−0.5		mA
Allowable Sink Current				
Per Pin (L, D, G)			10	mA
Per Pin (All Others)			2	mA
Per Port (L)			16	mA
Per Port (D, G)			10	mA
Allowable Source Current				
Per Pin (L)			−15	mA
Per Pin (All Others)			−1.5	mA

COP320/COP321/COP322

DC Electrical Characteristics $-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 5.5V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Operation Voltage		4.5	5.5	V
Power Supply Ripple	Peak to Peak (Note 3)		0.4	V
Supply Current	$T_A = -40°C$, Outputs Open		40	mA
Input Voltage Levels				
CKI Input Levels				
Crystal Input				
Logic High		2.2		V
Logic Low		−0.3	0.3	V
TTL Input	$V_{CC} = 5V \pm 5\%$			
Logic High		2.2		V
Logic Low		−0.3	0.6	V
Schmitt Trigger Inputs				
RESET, CKI (÷4)				
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.4	V
SO Input Level (Test Mode)	(Note 2)	2.0	3.0	V
All Other Inputs				
Logic High	V_{CC} = Max.	3.0		V
Logic High	$V_{CC} = 5V \pm 5\%$	2.2		V
Logic Low		−0.3	0.6	V
Input Levels High Trip Option				
Logic High		3.6		V
Logic Low		−0.3	1.2	V
Input Load Source Current	$V_{CC} = 5V$, $V_{IN} = 0V$			
CKO		−4	−800	μA
All Others		−100	−800	μA
Input Capacitance			7	pF
Hi-Z Input Leakage		−2	+2	μA
Output Voltage Levels				
Standard Outputs				
TTL Operation	$V_{CC} = 5V \pm 10\%$			
Logic High	$I_{OH} = -75\ \mu A$	2.4		V
Logic Low	$I_{OL} = 1.6$ mA	−0.3	0.4	V
CMOS Operation (Note 1)				
Logic High	$I_{OH} = -10\ \mu A$	$V_{CC} - 1$		V
Logic Low	$I_{OL} = +10\ \mu A$	−0.3	0.2	V
Output Current Levels				
LED Direct Drive Output	$V_{CC} = 5V$ (Note 4)			
Logic High	$V_{OH} = 2.0V$	1.0	12	mA
CKI Sink Current (R/C Option)	$V_{IN} = 3.5V$	2		mA
CKO (RAM Supply Current)	$V_R = 3.3V$		4	mA
TRI-STATE or Open Drain Leakage Current		−5	+5	μA
Allowable Sink Current				
Per Pin (L, D, G)			10	mA
Per Pin (All Others)			2	mA
Per Port (L)			16	mA
Per Port (D, G)			10	mA
Allowable Source Current				
Per Pin (L)			−15	mA
Per Pin (All Others)			−1.5	mA

Note 1: TRI-STATE and LED configurations are excluded.

Note 2: SO output "0" level must be less than 0.6V for normal operation.

AC Electrical Characteristics

COP420/COP421/COP422 $0°C \le T_A \le 70°C$, $4.5V \le V_{CC} \le 6.3V$ unless otherwise noted

COP320/COP321/COP322 $-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 5.5V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time		4	10	μs
Operating CKI Frequency	÷16 mode	1.6	4.0	MHz
	÷8 mode	0.8	2.0	MHz
CKI Duty Cycle (Note 1)		40	60	%
Rise Time	Freq. = 4 MHz		60	ns
Fall Time	Freq. = 4 MHz		40	ns
CKI Using RC *(Figure 8c)*	÷4 mode			
Frequency	R = 15 kΩ ±5%, C = 100 pF	0.5	1.0	MHz
Instruction Cycle Time (Note 5)		4	8	μs
CKO as SYNC Input *(Figure 8d)*				
t_{SYNC}	*Figure 3a*	50		ns
Inputs:				
SI				
t_{SETUP}		0.3		μs
t_{HOLD}		250		ns
All Other Inputs				
t_{SETUP}		1.7		μs
t_{HOLD}		300		ns
Output Propagation Delay	Test Conditions:	300		ns
	R_L = 5 kΩ, C_L = 50 pF, V_{OUT} = 1.5V			
SO and SK				
t_{pd1}			1.0	μs
t_{pd0}			1.0	μs
CKO				
t_{pd1}			0.25	μs
t_{pd0}			0.25	μs
All Other Outputs				
t_{pd1}			1.4	μs
t_{pd0}			1.4	μs
MICROBUS™ Timing	C_L = 100 pF, V_{CC} = 5V ±5%			
Read Operation *(Figure 4)*				
Chip Select Stable before $\overline{RD}$—t_{CSR}		65		ns
Chip Select Hold Time for $\overline{RD}$—t_{RCS}		20		ns
$\overline{RD}$ Pulse Width—t_{RR}		400		ns
Data Delay from $\overline{RD}$—t_{RD}			375	ns
$\overline{RD}$ to Data Floating—t_{DF}			250	ns
Write Operation *(Figure 5)*				
Chip Select Stable before $\overline{WR}$—t_{CSW}		65		ns
Chip Select Hold Time for $\overline{WR}$—t_{WCS}		20		ns
$\overline{WR}$ Pulse Width—t_{WW}		400		ns
Data Set-Up Time for $\overline{WR}$—t_{DW}		320		ns
Data Hold Time for $\overline{WR}$—t_{WD}		100		ns
INTR Transition Time from $\overline{WR}$—t_{WI}			700	ns

Note 1: Duty cycle = $t_{W1}/(t_{W1} + t_{W0})$.

Note 2: See *Figure 9* for additional I/O characteristics.

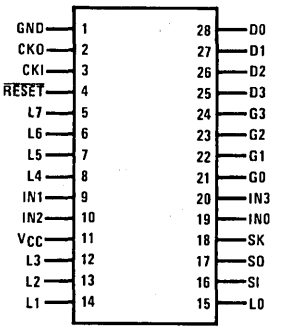
Note 3: Voltage change must be less than 0.5V in a 1 ms period.

Note 4: Exercise great care not to exceed maximum device power dissipation limits when direct driving LEDs (or sourcing similar loads) at high temperature.

Note 5: Variation due to the device included.

Connection Diagrams

COP420, COP320
Dual-In-Line Package

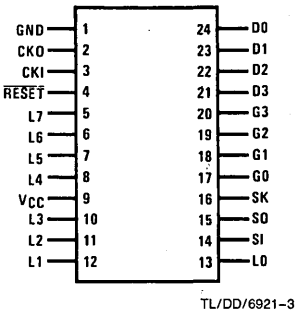


TL/DD/6921-2

Top View

Order Number COP420-XXX/N, COP320-XXX/N
See NS Package N28B

COP421, COP321
Dual-In-Line Package

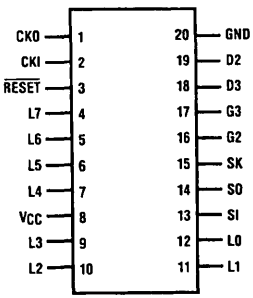


TL/DD/6921-3

Top View

Order Number COP421-XXX/M, COP321-XXX/M, COP421-XXX/N, COP321-XXX/N
See NS Package M24B or N24A

COP422, COP322
Dual-In-Line Package



TL/DD/6921-4

Top View

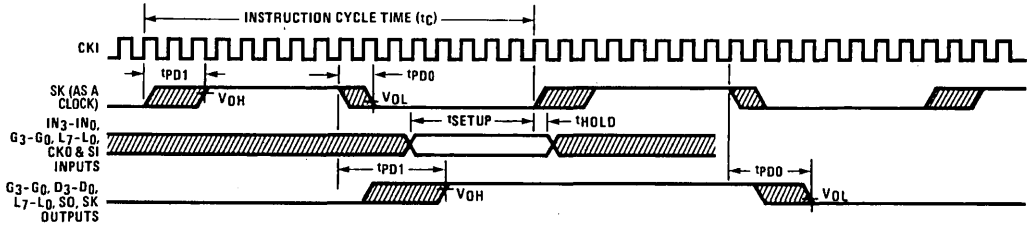
Order Number COP422-XXX/M, COP322-XXX/M, COP422-XXX/N, COP322-XXX/N
See NS Package M20B or N20A

FIGURE 2

Pin Descriptions

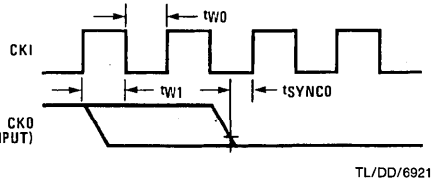
Pin	Description
L_7–L_0	8 bidirectional I/O ports with TRI-STATE
G_3–G_0	4 bidirectional I/O ports
D_3–D_0	4 general purpose outputs
IN_3–IN_0	4 general purpose inputs (COP420/320 only)
SI	Serial input (or counter input)
SO	Serial output (or general purpose output
SK	Logic-controlled clock (or general purpose output)
CKI	System oscillator input
CKO	System oscillator output (or general purpose input or RAM power supply)
$\overline{RESET}$	System reset input
V_{CC}	Power supply
GND	Ground

Timing Diagrams



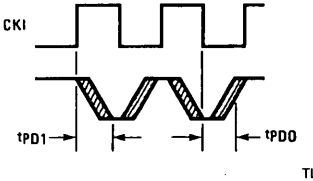
TL/DD/6921-5

FIGURE 3. Input/Output Timing Diagrams (Crystal Divide by 16 Mode)



TL/DD/6921-6

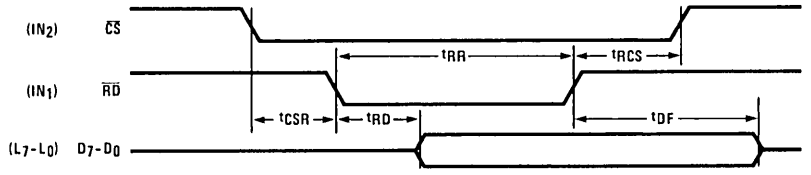
FIGURE 3A. Synchronization Timing



TL/DD/6921-7

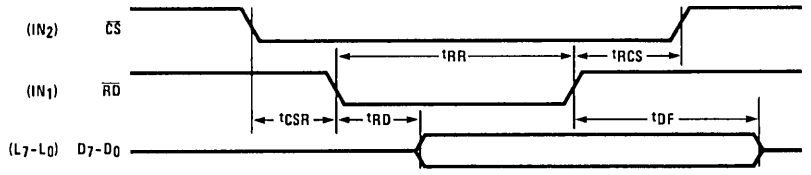
FIGURE 3B. CKO Output Timing

Timing Diagrams (Continued)



TL/DD/6921-8

FIGURE 4. MICROBUS Read Operation Timing



TL/DD/6921-9

FIGURE 5. MICROBUS Write Operation Timing

Functional Description COP420/COP421/COP422, COP320/COP321/COP322

For ease of reading this description, only COP420 and/or COP421 are referenced; however, all such references apply equally to the COP422, COP322, COP320 and/or COP321, respectively.

A block diagram of the COP420 is given in *Figure 1.* Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1" (greater than 2V). When a bit is reset, it is a logic "0" (less than 0.8V).

PROGRAM MEMORY

Program Memory consists of a 1,024 byte ROM. As can be seen by an examination of the COP420/421 instruction set, these words may be program instructions, program data or ROM addressing data. Because of the special characteristics associated with the JP, JSRP, JID and LQID instructions, ROM must often be thought of as being organized into 16 pages of 64 words each.

ROM addressing is accomplished by a 10-bit PC register. Its binary value selects one of the 1,024 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 10-bit binary count value. Three levels of subroutine nesting are implemented by the 10-bit subroutine save registers, SA, SB and SC, providing a last-in, first-out (LIFO) hardware subroutine stack.

ROM instruction words are fetched, decoded and executed by the Instruction Decode, Control and Skip Logic circuitry.

DATA MEMORY

Data memory consists of 256-bit RAM, organized as 4 data registers of 16 4-bit digits. RAM addressing is implemented by a 6-bit **B register** whose upper 2 bits (Br) select 1 of 4 data registers and lower 4 bits (Bd) select 1 of 16 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) is usually loaded into or from, or exchanged with, the A register (accumulator), it may also be loaded into or from the Q latches or loaded from the L ports.

RAM addressing may also be performed directly by the LDD and XAD instructions based upon the 6-bit contents of the operand field of these instructions. The Bd register also serves as a source register for 4-bit data sent directly to the D outputs.

INTERNAL LOGIC

The 4-bit **A register** (accumulator) is the source and destination register for most I/O, arithmetic, logic and data memory access operations. It can also be used to load the Br and Bd portions of the B register, to load the input 4 bits of the 8-bit Q latch data, to input 4 bits of the 8-bit L I/O port data and to perform data exchanges with the SIO register.

A **4-bit adder** performs the arithmetic and logic functions of the COP420/421, storing its results in A. It also outputs a carry bit to the 1-bit **C register,** most often employed to indicate arithmetic overflow. The C register, in conjunction with the XAS instruction and the EN register, also serves to control the SK output. C can be outputted directly to SK or can enable SK to be a sync clock each instruction cycle time. (See XAS instruction and EN register description, below.)

Four **general-purpose inputs, IN_3–IN_0,** are provided; IN_1, IN_2 and IN_3 may be selected, by a mask-programmable option, as Read Strobe, Chip Select and Write Strobe inputs, respectively, for use in MICROBUS applications.

The **D register** provides 4 general-purpose outputs and is used as the destination register for the 4-bit contents of Bd.

The **G register** contents are outputs to 4 general-purpose bidirectional I/O ports. G_0 may be mask-programmed as an output for MICROBUS applications.

The **Q register** is an internal, latched, 8-bit register, used to hold data loaded to or from M and A, as well as 8-bit data from ROM. Its contents are output to the L I/O ports when the L drivers are enabled under program control. (See LEI instruction.) With the MICROBUS option selected, Q can also be loaded with the 8-bit contents of the L I/O ports upon the occurrence of a write strobe from the host CPU.

Functional Description COP420/COP421/COP422, COP320/COP321/COP322 (Continued)

The **8 L drivers**, when enabled, output the contents of latched Q data to the L I/O ports. Also, the contents of L may be read directly into A and M. As explained above, the MICROBUS option allows L I/O port data to be latched into the Q register. L I/O ports can be directly connected to the segments of a multiplexed LED display (using the LED Direct Drive output configuration option) with Q data being outputted to the Sa–Sg and decimal point segments of the display.

The **SIO register** functions as a 4-bit serial-in/serial-out shift register or as a binary counter depending on the contents of the EN register. (See EN register description, below.) Its contents can be exchanged with A, allowing it to input or output a continuous serial data stream. SIO may also be used to provide additional parallel I/O by connecting SO to external serial-in/parallel-out shift registers. For example of additional parallel output capacity see **Application #2.**

The XAS instruction copies C into the **SKL latch.** In the counter mode, SK is the output of SKL; in the shift register mode, SK outputs SKL ANDed with the clock.

The **EN register** is an internal 4-bit register loaded under program control by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register (EN_3–EN_0).

1. The least significant bit of the enable register, EN_0 selects the SIO register as either a 4-bit shift register or a 4-bit binary counter. With EN_0 set, SIO is an asynchronous binary counter, *decrementing* its value by one upon each low-going pulse ("1" to "0" occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output is equal to the value of EN_3. With EN_0 reset, SIO is a serial shift register shifting let each instruction cycle time. The data present at DI goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. (See 4 below.) The SK output becomes a logic-controlled clock.
2. With the EN_1 set the IN_1 input is enabled as an interrupt input. Immediately following an interrupt, EN_1 is reset to disable further interrupts.
3. With EN_2 set, the L drivers are enabled to output the data in Q to the L I/O ports. Resetting EN_2 disables the L drivers, placing the L I/O ports in a high impedance input state.
4. EN_3, in conjunction with EN_0, affects the SO output. With EN_0 set (binary counter option selected) SO will output the value loaded into EN_3. With EN_0 reset (serial shift register option selected), setting EN enables SO as the output of the SIO shift register outputting serial shifted data each instruction time. Resetting EN_3 with the serial shift register option selected disables SO as the shift register output data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains reset to "0". The table below provides summary of the modes associated with EN_3 and EN_1.

OSCILLATOR

There are three basic clock oscillator configurations available as shown by *Figure 8.*

a. **Crystal Controlled Oscillator.** CKI and CKO are connected to an external crystal. The instruction cycle time equals the crystal frequency divided by 16 (optional by 8).

b. **External Oscillator.** CKI is an external clock input signal. The external frequency is divided by 16 (optional by 8) to give the instruction cycle time. CKO is now available to be used as the RAM power supply (V_R) of as a general purpose input.

c. **RC Controlled Oscillator.** CKI is configured as a single pin RC controlled Schmitt trigger oscillator. The instruction cycle equals the oscillation frequency divided by 4. CKO is available for non-timing functions.

CKO PIN OPTIONS

In a crystal controlled oscillator system, CKO is used as an output to the crystal network. As an option CKO can be a general purpose input, read into bit 2 of A (accumulator) upon execution of an INIL instruction. As another option, CKO can be a RAM power supply pin (V_R), allowing its connection to a standby/backup power supply to maintain the integrity of RAM data with minimum power drain when the main supply is inoperative or shut down to conserve power. Using either option is appropriate in applications where the COP420/421 system timing configuration does not require use of the CKO pin.

RAM KEEP-ALIVE OPTION (NOT AVAILABLE ON COP422)

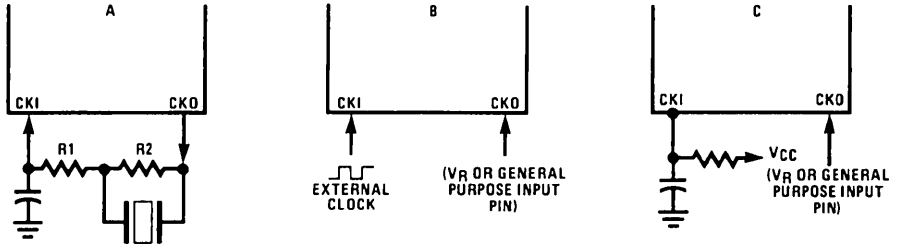
Selecting CKO as the RAM power supply (V_R) allows the user to shut off the chip power supply (V_{CC}) and maintain data in the RAM. To insure that RAM data integrity is maintained, the following conditions must be met:

1. $\overline{\text{RESET}}$ must go low before V_{CC} goes below spec during power off; V_{CC} must be within spec before $\overline{\text{RESET}}$ goes high on power up.
2. V_R must be within the operating range of the chip, and equal to $V_{CC} \pm 1V$ during normal operation.
3. V_R must be $\geq$ 3.3V with V_{CC} off.

Enable Register Modes—Bits EN_3 and EN_0

EN_3	EN_0	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL = 1, SK = CLOCK If SKL = 0, SK = 0
1	0	Shift Register	Input to Shift Register	Serial Out	If SKL = 1, SK = CLOCK If SKL = 0, SK = 0
0	1	Binary Counter	Input to Binary Counter	0	If SKL = 1, SK = 1 If SKL = 0, SK = 0
1	1	Binary Counter	Input to Binary Counter	1	If SKL = 1, SK = 1 If SKL = 0, SK = 0

Functional Description COP420/COP421/COP422, COP320/COP321/COP322 (Continued)



TL/DD/6921–10

Crystal Oscillator **External Oscillator** **RC Controlled Oscillator**

Crystal Oscillator

Crystal Value	Component Values		
	R1(Ω)	R2(Ω)	C(pF)
4 MHz	1k	1M	27
3.58 MHz	1k	1M	27
2.09 MHz	1k	1M	56

RC Controlled Oscillator

R(kΩ)	C(pF)	Instruction Cycle Time (μs)
12	100	5 ±20%
6.8	220	5.3 ±23%
8.2	300	8 ±29%
22	100	8.6 ±16%

Note: 50 kΩ ≥ R ≥ 5 kΩ
360 pF ≥ C ≥50 pF

FIGURE 8. COP420/421/COP320/321 Oscillator

INTERRUPT

The following features are associated with the IN_1 interrupt procedure and protocol and must be considered by the programmer when utilizing interrupts.

a. The interrupt, once acknowledged as explained below, pushes the next sequential program counter address (PC + 1) onto the stack, pushing in turn the contents of the other subroutine-save registers to the next lower level (PC + 1 → SA → SB → SC). Any previous contents of SC are lost. The program counter is set to hex address 0FF (the last word of page 3) and EN_1 is reset.

b. An interrupt will be acknowledged only after the following conditions are met:

1. EN_1 has been set.
2. A low-going pulse ("1" to "0") at least two instruction cycles wide occurs on the IN_1 input.
3. A currently executing instruction has been completed.
4. All successive transfer of control instructions and successive LBIs have been completed (e.g., if the main program is executing a JP instruction which transfers program control to another JP instruction, the interrupt will not be acknowledged until the second JP instruction has been executed.

c. Upon acknowledgement of an interrupt, the skip logic status is saved and later restored upon popping of the stack. For example, if an interrupt occurs during the execution of ASC (Add with Carry, Skip on Carry) instruction which results in carry, the skip logic status is saved and program control is transferred to the interrupt servicing routine at hex address 0FF. At the *end* of the interrupt routine, a RET instruction is executed to "pop" the stack and return program control to the instruction following the original ASC. *At this time,* the skip logic is enabled and skips this instruction because of the previous ASC carry. Subroutines and LQID instructions should not be nested within the interrupt service routine, since their popping the stack will enable any previously saved main program skips, interfering with the orderly execution of the interrupt routine.

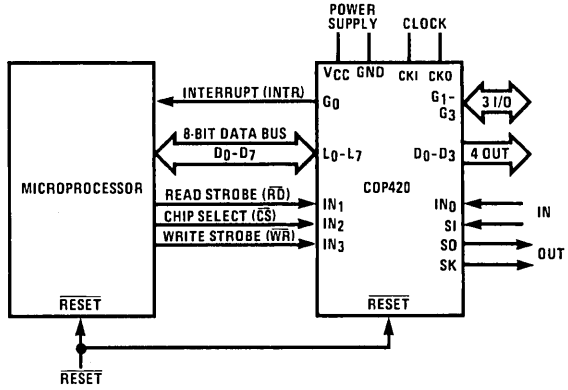
d. The first instruction of the interrupt routine at hex address 0FF must be a NOP.

e. A LEI instruction can be put immediately before the RET to re-enable interrupts.

MICROBUS™ INTERFACE

The COP420 has an option which allows it to be used as a peripheral microprocessor device, inputting and outputting data from and to a host microprocessor (μP). IN_1, IN_2 and IN_3 general purpose inputs become **MICROBUS compatible** read-strobe, chip-select, and write-strobe lines, respectively. IN_1 becomes $\overline{RD}$—a logic "0" on this input will cause Q latch data to be enabled to the L ports for input to the μP. IN_2 becomes $\overline{CS}$—a logic "0" on this line selects the COP420 as the μP peripheral device by enabling the operation of the $\overline{RD}$ and $\overline{WR}$ lines and allows for the selection of one of several peripheral components. IN_3 becomes $\overline{WR}$—a logic "0" on this line will write bus data from the L ports to the Q latches for input to the COP420. G_0 becomes INTR a "ready" output, reset by a write pulse from the μP on the $\overline{WR}$ line, providing the "handshaking" capability necessary for asynchronous data transfer between the host CPU and the COP420.

Functional Description COP420/COP421/COP422, COP320/COP321/COP322 (Continued)



TL/DD/6921-12

FIGURE 6. MICROBUS Option Interconnect

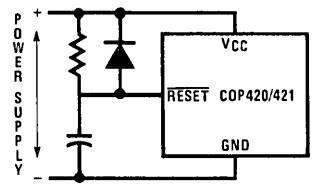
This option has been designed for compatibility with National's MICROBUS—a standard interconnect system for 8-bit parallel data transfer between MOS/LSI CPUs and interfacing devices. (See MICROBUS National Publication.) The functioning and timing relationships between the COP420 signal lines affected by this option are as specified for the MICROBUS interface, and are given in the AC electrical characteristics and shown in the timing diagrams (*Figures 4* and *5*). Connection of the COP420 to the MICROBUS is shown in *Figure 6.*

Note: TRI-STATE outputs must be used on L-port.

INITIALIZATION

The Reset Logic, internal to the COP420/421, will initialize (clear) the device upon power-up if the power supply rise time is less than 1 ms and greater than 1 μs. If the power supply rise time is greater than 1 ms, the user must provide an external RC network and diode to the $\overline{\text{RESET}}$ pin as shown below. The $\overline{\text{RESET}}$ pin is configured as a Schmitt trigger input. If not used it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the RESET input, provided it stays low for at least three instruction cycle times.

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN, and G registers are cleared. The SK output is enabled as a SYNC output, providing a pulse each instruction cycle time. *Data Memory (RAM) is not cleared upon initialization.* The first instruction at address 0 must be a CLRA.



TL/DD/6921-13

FIGURE 7. Power-Up Clear Circuit

I/O OPTIONS

COP420/421 outputs have the following optional configurations, illustrated in *Figure 9a*:

a. Standard—an enhancement mode device to ground in conjunction with a depletion-mode device to V_{CC}, compatible with TTL and CMOS input requirements. Available on SO, SK, and all D and G outputs.

b. Open-Drain—an enhancement-mode device to ground only, allowing external pull-up as required by the user's application. Available on SO, SK, and all D and G outputs.

c. Push-Pull—An enhancement-mode device to ground in conjunction with a depletion-mode device paralleled by an enhancement-mode device to V_{CC}. This configuration has been provided to allow for fast rise and fall times when driving capacitive loads. Available on SO and SK outputs only.

d. Standard L—same as **a.**, but may be disabled. Available on L outputs only.

e. Open Drain L—same as **b.**, but may be disabled. Available on L outputs only.

f. LED Direct Drive—an enhancement-mode device to ground and to V_{CC}, meeting the typical current sourcing requirements of the segments of an LED display. The sourcing device is clamped to limit current flow. These devices may be turned off under program control (See Functional Description, EN Register), placing the outputs in a high-impedance state to provide required LED segment blanking for a multiplexed display.

g. TRI-STATE Push-Pull—an enhancement-mode device to ground and V_{CC}. These outputs are TRI-STATE outputs, allowing for connection of these outputs to a data bus shared by other bus drivers.

COP420/COP421 inputs have the following optional configurations:

h. An on-chip depletion load device to V_{CC}.

i. A Hi-Z input which must be driven to a "1" or "0" by external components.

The above input and output configurations share common enhancement-mode and depletion-mode devices. Specifically, all configurations use one or more of six devices (numbered 1–6, respectively). Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figure 9b* for each

Functional Description COP420/COP421/COP422, COP320/COP321/COP322 (Continued)

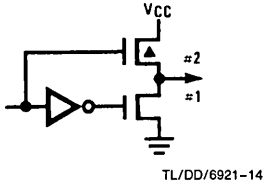
of these devices to allow the designer to effectively use these I/O configurations in designing a COP420/421 system.

The SO, SK outputs can be configured as shown in **a.**, **b.**, or **c.** The D and G outputs can be configured as shown in **a.** or **b.** Note that when inputting data to the G ports, the G outputs should be set to "1." The L outputs can be configured as in **d.**, **e.**, **f.** or **g.**

An important point to remember if using configuration **d.** or **f.** with the L drivers is that even when the L drivers are disabled, the depletion load device will source a small amount of current (see *Figure 9b,* device 2); however, when the L lines are used as input, the disabled depletion device can *not* be relied on to source sufficient current to pull an input to logic "1".

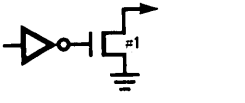
COP421

If the COP420 is bonded as a 24-pin device, it becomes the COP421, illustrated in *Figure 2,* COP420/421 Connection Diagrams. Note that the COP421 does not contain the four general purpose IN inputs (IN_3–IN_0). Use of this option precludes, of course, use of the IN options, interrupt feature, and the MICROBUS option which uses IN_1–IN_3. All other options are available for the COP421.



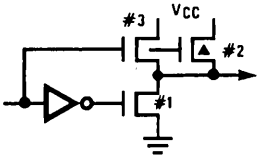
TL/DD/6921-14

a. Standard Output



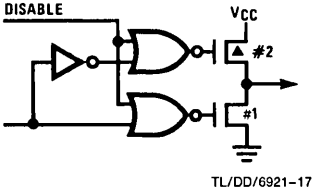
TL/DD/6921-15

b. Open-Drain Output



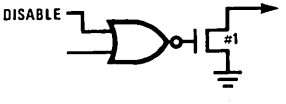
TL/DD/6921-16

c. Push-Pull Output



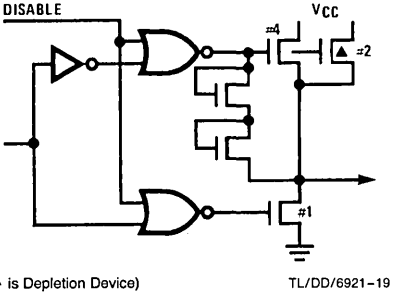
TL/DD/6921-17

d. Standard L Output



TL/DD/6921-18

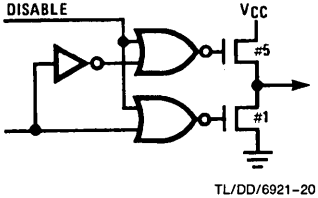
e. Open-Drain L Output



(▲ is Depletion Device)

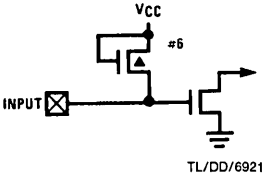
TL/DD/6921-19

f. LED (L Output)



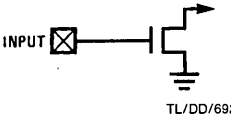
TL/DD/6921-20

g. TRI-STATE Push-Pull (L Output)



TL/DD/6921-21

h. Input with Load

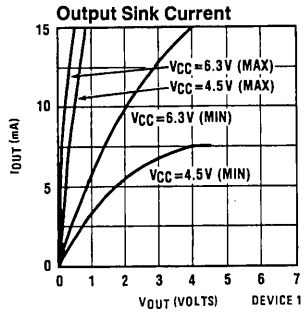


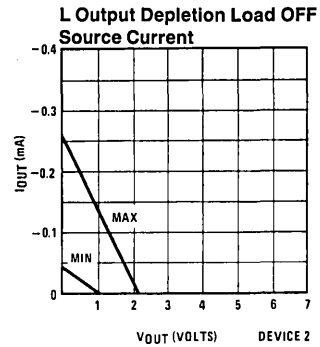
TL/DD/6921-22

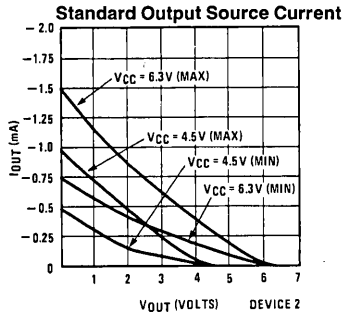
i. Hi-Z Input

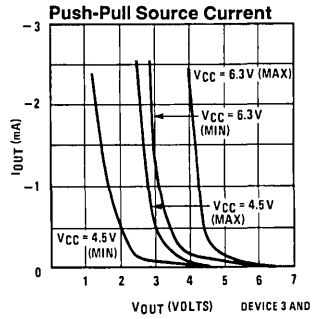
FIGURE 9a. Input/Output Configurations

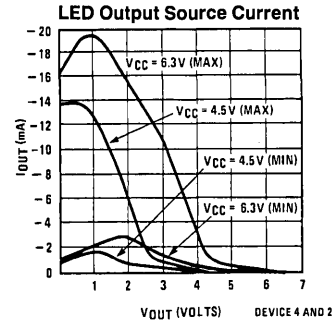
Typical Performance Characteristics

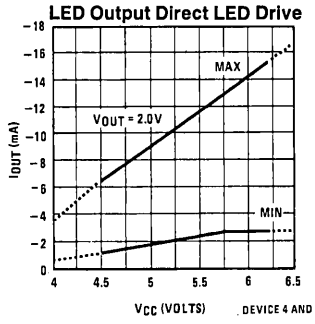


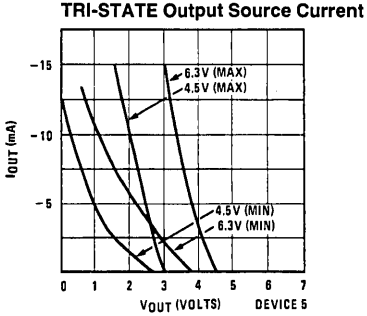


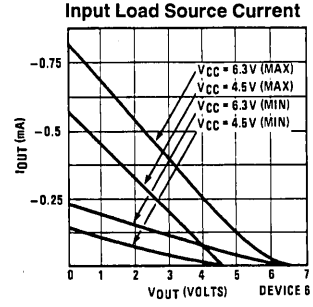










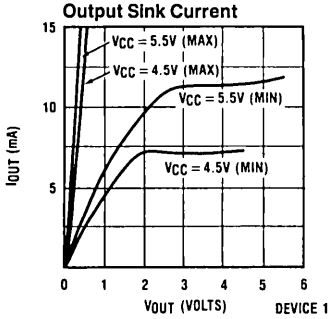


TL/DD/6921-23

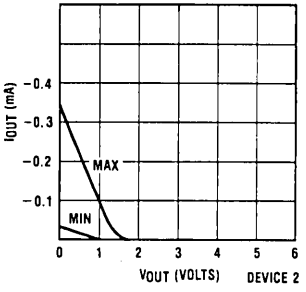
FIGURE 9b. COP420/COP421 Input/Output Characteristics

Typical Performance Characteristics (Continued)

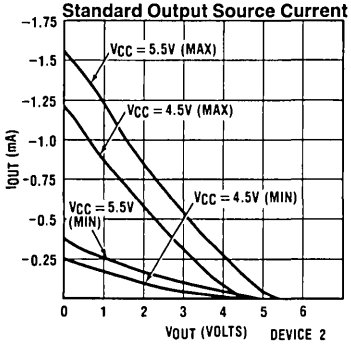
Output Sink Current



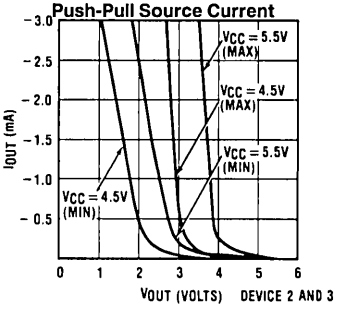
L Output Depletion Load OFF Source Current



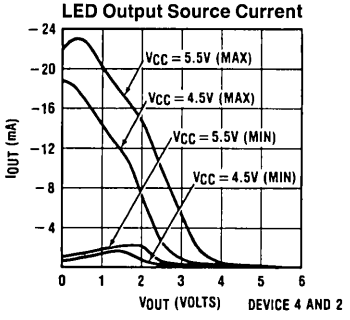
Standard Output Source Current



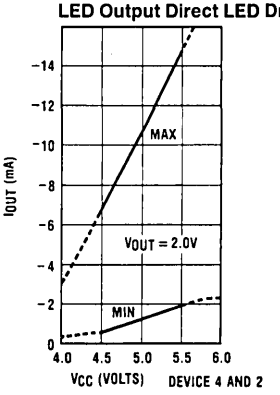
Push-Pull Source Current



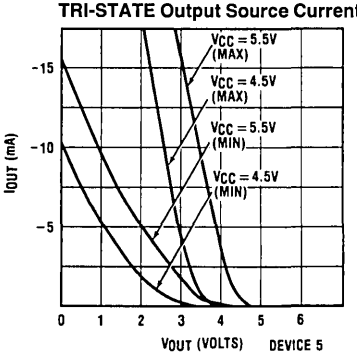
LED Output Source Current



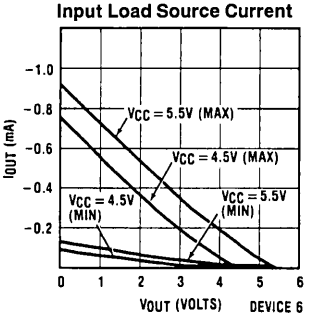
LED Output Direct LED Drive



TRI-STATE Output Source Current



Input Load Source Current



TL/DD/6921–24

FIGURE 9c. COP320/COP321 Input/Output Characteristics

Instruction Set

Table I is a symbol table providing internal architecture, instruction operand and operational symbols used in the instruction set table.

Table II provides the mnemonic, operand, machine code, data flow, skip conditions and description associated with each instruction in the COP420/COP421/COP422 instruction set.

TABLE I. COP420/421/422/320/321/322 Instruction Set Table Symbols

Symbol	Definition
INTERNAL ARCHITECTURE SYMBOLS	
A	4-bit Accumulator
B	6-bit RAM Address Register
Br	Upper 2 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry Register
D	4-bit Data Output Port
EN	4-bit Enable Register
G	4-bit Register to latch data for G I/O Port
IL	Two 1-bit latches associated with the IN_3 or IN_0 inputs
IN	4-bit Input Port
L	8-bit TRI-STATE I/O Port
M	4-bit contents of RAM Memory pointed to by B Register
PC	9-bit ROM Address Register (program counter)
Q	8-bit Register to latch data for L I/O Port
SA	10-bit Subroutine Save Register A
SB	10-bit Subroutine Save Register B
SC	10 Subroutine Save Register A
SIO	4-bit Shift Register and Counter
SK	Logic-Controlled Clock Output

Symbol	Definition
INSTRUCTION OPERAND SYMBOLS	
d	4-bit Operand Field, 0-15 binary (RAM Digit Select)
r	2-bit Operand Field, 0-3 binary (RAM Register Select)
a	10-bit Operand Field, 0-1023 binary (ROM Address)
y	4-bit Operand Field, 0-15 binary (Immediate Data)
RAM(s)	Contents of RAM location addressed by s
ROM(t)	Contents of ROM location addressed by t
OPERATIONAL SYMBOLS	
+	Plus
−	Minus
→	Replaces
↔	Is exchanged with
=	Is equal to
$\overline{A}$	The one's complement of A
⊕	Exclusive-OR
:	Range of values

TABLE II. COP420/421/422/320/321/322 Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	0011 | 0000	A + C + RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	0011 | 0001	A + RAM(B) → A	None	Add RAM to A
ADT		4A	0100 | 1010	A + 10_{10} → A	None	Add Ten to A
AISC	y	5−	0101 | y	A + y → A	Carry	Add immediate, Skip on Carry (y ≠ 0)
CASC		10	0001 | 0000	$\overline{A}$ + RAM(B) + C → A Carry → C	Carry	Complement and Add with Carry, Skip on Carry
CLRA		00	0000 | 0000	0 → A	None	Clear A
COMP		40	0100 | 0000	$\overline{A}$ → A	None	One's complement of A to A
NOP		44	0100 | 0100	None	None	No Operation
RC		32	0011 | 0010	"0" → C	None	Reset C
SC		22	0010 | 0010	"1" → C	None	Set C
XOR		02	0000 | 0010	A ⊕ RAM(B) → A	None	Exclusive-OR RAM with A

Instruction Set (Continued)

TABLE II. COP420/421/422/320/321/322 Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
TRANSFER OF CONTROL INSTRUCTIONS						
JID		FF	1111 | 1111	ROM (PC_8, A,M) → $PC_{7:0}$	None	Jump Indirect (Note 3)
JMP	a	6–	0110 | 00 | a_8	a → PC	None	Jump
		--	$a_{7:0}$			
JP	a	--	1 | $a_{6:0}$ (pages 2,3 only)	a → $PC_{6:0}$	None	Jump within Page (Note 4)
			or			
		--	11 | $a_{5:0}$ (all other pages)	a → $PC_{5:0}$		
JSRP	a	--	10 | $a_{5:0}$	PC + 1 → SA → SB → SC 010 → $PC_{8:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 5)
JSR	a	6–	0110 | 10 | $a_{9:8}$	PC + 1 → SA → SB → SC a → PC	None	Jump to Subroutine
		--	$a_{7:0}$			
RET		48	0100 | 1000	SC → SB → SA → PC	None	Return from Subroutine
RETSK		49	0100 | 1001	SC → SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
MEMORY REFERENCE INSTRUCTIONS						
CAMQ		33	0011 | 0011	A → $Q_{7:4}$ RAM(B) → $Q_{3:0}$	None	Copy A, RAM to Q
		3C	0011 | 1100			
CQMA		33	0011 | 0011	$Q_{7:4}$ → RAM(B) $Q_{3:0}$ → A	None	Copy Q to RAM, A
		2C	0010 | 1100			
LD	r	–5	00 | r | 0101	RAM(B) → A Br ⊕ r → Br	None	Load RAM into A Exclusive-OR Br with r
LDD	r,d	23	0010 | 0011	RAM(r,d) → A	None	Load A with RAM pointed to directly by r,d
		--	00 | r | d			
LQID		BF	1011 | 1111	ROM($PC_{9:8}$,A,M) → Q SB → SC	None	Load Q Indirect (Note 3)
RMB	0	4C	0100 | 1100	0 → $RAM(B)_0$	None	Reset RAM Bit
	1	45	0100 | 0101	0 → $RAM(B)_1$		
	2	42	0100 | 0010	0 → $RAM(B)_2$		
	3	43	0100 | 0011	0 → $RAM(B)_3$		
SMB	0	4D	0100 | 1101	1 → $RAM(B)_0$	None	Set RAM Bit
	1	47	0100 | 1101	1 → $RAM(B)_1$		
	2	46	0100 | 0110	1 → $RAM(B)_2$		
	3	4B	0100 | 1011	1 → $RAM(B)_3$		
STII	y	7–	0111 | y	y → RAM(B) Bd + 1 → Bd	None	Store Memory Immediate and Increment Bd
X	r	–6	00 | r | 0110	RAM(B) ⟷ A Br ⊕ r → Br	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	r,d	23	0010 | 0011	RAM(r,d) ⟷ A	None	Exchange A with RAM pointed to directly by r,d
		--	10 | r | d			

Instruction Set (Continued)

TABLE II. COP420/421/422/320/321/322 Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
MEMORY REFERENCE INSTRUCTIONS (Continued)						
XDS	r	−7	|00|r|0111|	RAM(B) ⟷ A Bd − 1 → Bd Br ⊕ r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd, Exclusive-OR Br with r
XIS	r	−4	|00|r|0100|	RAM(B) ⟷ A Bd + 1 → Bd Br ⊕ r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd, Exclusive-OR Br with r
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	|0101|0000|	A → Bd	None	Copy A to Bd
CBA		4E	|0100|1110|	Bd → A	None	Copy Bd to A
LBI	r,d	– –	|00|r|(d - 1)| (d = 0,9:15) or 33 |0011|0011| – – |10|r| d | (any d)	r,d → B	Skip until not a LBI	Load B Immediate with r,d (Note 6)
LEI	y	33 6–	|0011|0011| |0010| y |	y → EN	None	Load EN Immediate (Note 7)
XABR		12	|0001|0010|	A ⟷ Br (0,0 → A_3,A_2)	None	Exchange A with Br
TEST INSTRUCTIONS						
SKC		20	|0010|0000|		C = "1"	Skip if C is True
SKE		21	|0010|0001|		A = RAM(B)	Skip if A Equals RAM
SKGZ		33 21	|0011|0011| |0010|0001|		$G_{3:0}$ = 0	Skip if G is Zero (all 4 bits)
SKGBZ		33	|0011|0011|	1st byte		Skip if G Bit is Zero
	0	01	|0000|0001|	2nd byte	G_0 = 0	
	1	11	|0001|0001|	2nd byte	G_1 = 0	
	2	03	|0000|0011|	2nd byte	G_2 = 0	
	3	13	|0010|0011|	2nd byte	G_3 = 0	
SKMBZ	0	01	|0000|0001|		$RAM(B)_0$ = 0	Skip if RAM Bit is Zero
	1	11	|0001|0001|		$RAM(B)_1$ = 0	
	2	03	|0000|0011|		$RAM(B)_2$ = 0	
	3	13	|0001|0011|		$RAM(B)_3$ = 0	
SKT		41	|0100|0001|		A time-base counter carry has occurred since last test	Skip on Timer (Note 3)

Instruction Set (Continued)

TABLE II. COP420/421/422/320/321/322 Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
INPUT/OUTPUT INSTRUCTIONS						
ING		33 2A	0011 | 0011 0010 | 1010	G → A	None	Input G Ports to A
ININ		33 28	0011 | 0011 0010 | 1000	IN → A	None	Input IN Inputs to A (Note 2)
INIL		33 29	0011 | 0011 0010 | 1001	IL_3, CKO, "0", IL_0 → A	None	Input IL Latches to A (Note 3)
INL		33 2E	0011 | 0011 0010 | 1110	$L_{7:4}$ → RAM(B) $L_{3:0}$ → A	None	Input L Ports to RAM, A
OBD		33 3E	0011 | 0011 0011 | 1110	Bd → D	None	Output Bd to D Outputs
OGI	y	33 5–	0011 | 0011 0101 | y	y → G	None	Output to G Ports Immediate
OMG		33 3A	0011 | 0011 0011 | 1010	RAM(B) → G	None	Output RAM to G Ports
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO (Note 3)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicitly defined (e.g., Br and Bd are explicitly defined). Bits are numbered 0 to N where 0 signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit register.

Note 2: The ININ instruction is not available on the COP421/COP321 and COP422/COP322 since these devices do not contain the IN inputs.

Note 3: For additional information on the operation of the XAS, JID, LQID, INIL, and SKT instructions, see below.

Note 4: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 5: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 6: LBI is a single-byte instruction if d = 0, 9, 10, 11, 12, 13, 14, or 15. The machine code for the lower 4 bits equals the binary value of the "d" data *minus 1*, e.g., to load the lower four bits of B (Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Note 7: Machine code for operand field y for LEI instruction should equal the binary value to be latched into EN, where a "1" or "0" in each bit of EN corresponds with the selection or deselection of a particular function associated with each bit. (See Functional Description, EN Register.)

Description of Selected Instructions

The following information is provided to assist the user in understanding the operation of several unique instructions and to provide notes useful to programmers in writing COP420/421 programs.

XAS INSTRUCTION

XAS (Exchange A with SIO) exchanges the 4-bit contents of the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. An XAS instruction will also affect the SK output. (See Functional Description, EN Register, above.) If SIO is selected as a shift register, an XAS instruction must be performed once every 4 instruction cycles to effect a continuous data stream.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower 8 bits of the ROM address register PC with the *contents* of ROM addressed by the 10-bit word, $PC_{9:8}$, A, M. PC_9 and PC_8 are not affected by this instruction.

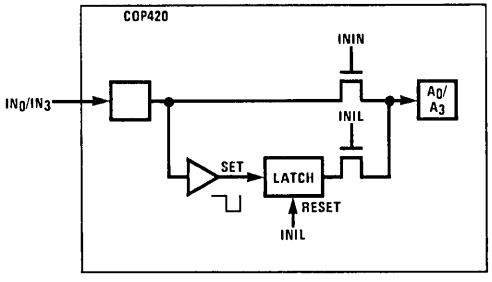
Note that JID requires 2 instruction cycles to execute.

Description of Selected Instructions (Continued)

INIL INSTRUCTION

INIL (Input IL Latches to A) inputs 2 latches, IL_3 and IL_0 (see *Figure 10*) and CKO into A. The IL_3 and IL_0 latches are set if a low-going pulse ("1" to "0") has occurred on the IN_3 and IN_0 inputs since the last INIL instruction, provided the input pulse stays low for at least two instruction times. Execution of an INIL inputs IL_3 and IL_0 into A3 and A0 respectively, and resets these latches to allow them to respond to subsequent low-going pulses on the IN_3 and IN_0 lines. If CKO is mask programmed as a general purpose input, an INIL will input the state of CKO into A2. If CKO has not been so programmed, a "1" will be placed in A2. A "0" is always placed in A1 upon the execution of an INIL. The general purpose inputs IN_3–IN_0 are input to A upon execution of an ININ instruction. (See Table II, ININ instruction.) INIL is useful in recognizing pulses of short duration or pulses which occur too often to be read conveniently by an ININ instruction.

Note: IL latches are not cleared on reset.



TL/DD/6921–25

FIGURE 10

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 10-bit word PC_9, PC_8, A, M. LQID can be used for table lookup or code conversion such as BCD to seven-segment. The LQID instruction "pushes" the stack (PC + 1 → SA → SB → SC) and replaces the least significant 8 bits of PC as follows: A → $PC_{7:4}$, RAM(B) → $PC_{3:0}$, leaving PC_9 and PC_8 unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SC → SB → SA → PC), restoring the saved value of PC to continue sequential program execution. Since LQID pushes SB → SC, the previous contents of SC are lost. Also, when LQID pops the stack, the previously pushed contents of SB are left in SC. The net result is that the content of SB are placed in SC (SB → SC). Note that LQID takes two instruction cycle times to execute.

SKT INSTRUCTION

The SKT (Skip On Timer) instruction tests the state of an internal 10-bit time-base counter. This counter divides the instruction cycle clock frequency by 1024 and provides a latched indication of counter overflow. The SKT instruction tests this latch, executing the next program instruction if the latch is not set. If the latch has been set since the previous test, the next program instruction is skipped and the latch is reset. The features associated with this instruction, therefore, allow the COP420/421 to generate its own time-base for real-time processing rather than relying on an external input signal.

For example, using a 2.097 MHz crystal as the time-base to the clock generator, the instruction cycle clock frequency will be 131 kHz (crystal frequency ÷ 16) and the binary counter output pulse frequency will be 128 Hz. For time-of-day or similar real-time processing, the SKT instruction can call a routine which increments a "seconds" counter every 128 ticks.

INSTRUCTION SET NOTES

a. The first word of a COP420/421 program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instruction are not executed, one instruction cycle time is devoted to skipping each byte of the skipped instruction. Thus all program paths take the same number of cycle times whether instructions are skipped or executed except JID and LQID. LQID and JID take two cycle times if executed and one if skipped.

c. The ROM is organized into 16 pages of 64 words each. The Program Counter is a 10-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID or LQID instruction is located in the last word of a page, the instruction operates as if it were in the next page. For example: a JP located in the last word of a page will jump to a location in the next page. Also, a LQID or JID located in the last word of page 3, 7, 11 or 15 will access data in the next group of four pages.

Option List

The COP420/421/422 mask-programmable options are assigned numbers which correspond with the COP*420* pins.

The following is a list of COP420 options. When specifying a COP421 or COP422 chip, Options 9, 10, 19, 20 and 29 must all be set to zero. When specifying a COP422 chip, Options 21, 22, 27 and 28 must also be zero, and Option 2 must not be a 1. The options are programmed at the same time as the ROM pattern to provide the user with the hardware flexibility to interface to various I/O components using little or no external circuitry.

Option 1 = 0: Ground—no options available

Option 2: CKO Pin
- = 0: clock generator output to crystal
  0 not available if option 3 = 4 or 5)
- = 1: Pin is RAM power supply (V_R) input
  (Not available on COP422/COP322)
- = 2: general purpose input with load device
- = 4: general purpose Hi Z input

Option 3: CKI Input
- = 0: crystal input devided by 16
- = 1: crystal input divided by 8
- = 2: TTL external clock input divided by 16
- = 3: TTL external clock input divided by 8
- = 4: single-pin RC controlled oscillator (÷4)
- = 5: Schmitt trigger clock input (÷4)

Option 4: RESET Pin
- = 0: load devices to V_{CC}
- = 1: Hi-Z input

Option 5: L_7 Driver
- = 0: Standard output *(Figure 9D)*
- = 1: Open-Drain output (E)
- = 2: LED direct drive output (F)
- = 3: TRI-STATE push-pull output (G)

Option 6: L_6 Driver
same as Option 5

Option 7: L_5 Driver
same as Option 5

Option 8: L_4 Driver
same as Option 5

Option 9: IN_1 Input
- = 0: load devices to V_{CC} (H)
- = 1: Hi-Z input (I)

Option 10: IN_2 Input
same as Option 9

Option 11 = 0: V_{CC} Pin—no options available

Option 12: L_3 Driver
same as Option 5

Option 13: L_2 Driver
same as Option 5

Option 14: L_1 Driver
same as Option 5

Option 15: L_0 Driver
same as Option 5

Option 16: SI Input
same as Option 9

Option 17: SO Driver
- = 0: standard output (A)
- = 1: open-drain output (B)
- = 2: push-pull output (C)

Option 18: SK Driver
same as Option 17

Option 19: IN_0 Input
same as Option 9

Option 20: IN_3 Input
same as Option 9

Option 21: G_0 I/O Port
- = 0: Standard output (A)
- = 1: Open-Drain output (B)

Option 22: G_1 I/O Port
same as Option 21

Option 23: G_2 I/O Port
same as Option 21

Option 24: G_3 I/O Port
same as Option 21

Option 25: D_3 Output
- = 0: Standard output (A)
- = 1: Open-Drain output (B)

Option 26: D_2 Output
same as Option 25

Option 27: D_1 Output
same as Option 25

Option 28: D_0 Output
same as Option 25

Option 29: COP Function
- = 0: normal operation
- = 1: MICROBUS option

Option 30: COB Bonding
- = 0: COP420 (28-pin device)
- = 1: COP421 (24-pin device)
- = 2: 28- and 24-pin device
- = 3: COP422 (20-pin device)
- = 4: 28- and 20-pin device
- = 5: 24- and 20-pin device
- = 6: 28-, 24- and 20-pin device

Option 31: In Input Levels
- = 0: normal input levels
- = 1: Higher voltage input levels
  ("0" = 1.2V, "1" = 3.6V)

Option 32: G Input Levels
same as Option 31

Option 33: L Input Levels
same as Option 31

Option 34: CKO Input Levels
same as Option 31

Option 35: SI Input Levels
same as Option 31

Option List (Continued)

COP OPTION LIST

The following option information is to be sent to National along with the EPROM.

OPTION DATA

OPTION 1 VALUE = _____ IS: GROUND PIN
OPTION 2 VALUE = _____ IS: CKO PIN
OPTION 3 VALUE = _____ IS: CKI INPUT
OPTION 4 VALUE = _____ IS: RESET INPUT
OPTION 5 VALUE = _____ IS: L_7 DRIVER
OPTION 6 VALUE = _____ IS: L_6 DRIVER
OPTION 7 VALUE = _____ IS: L_5 DRIVER
OPTION 8 VALUE = _____ IS: L_4 DRIVER
OPTION 9 VALUE = _____ IS: IN1 INPUT
OPTION 10 VALUE = _____ IS: IN2 INPUT
OPTION 11 VALUE = _____ IS: VCC PIN
OPTION 12 VALUE = _____ IS: L_3 DRIVER
OPTION 13 VALUE = _____ IS: L_2 DRIVER
OPTION 14 VALUE = _____ IS: L_1 DRIVER
OPTION 15 VALUE = _____ IS: L_0 DRIVER
OPTION 16 VALUE = _____ IS: SI INPUT
OPTION 17 VALUE = _____ IS: SO DRIVER
OPTION 18 VALUE = _____ IS: SK DRIVER
OPTION 19 VALUE = _____ IS: IN_0 INPUT
OPTION 20 VALUE = _____ IS: IN_3 INPUT
OPTION 21 VALUE = _____ IS: G_0 I/O PORT
OPTION 22 VALUE = _____ IS: G_1 I/O PORT
OPTION 23 VALUE = _____ IS: G_2 I/O PORT
OPTION 24 VALUE = _____ IS: G_3 I/O PORT
OPTION 25 VALUE = _____ IS: D_3 OUTPUT
OPTION 26 VALUE = _____ IS: D_2 OUTPUT
OPTION 27 VALUE = _____ IS: D_1 OUTPUT
OPTION 28 VALUE = _____ IS: D_0 OUTPUT
OPTION 29 VALUE = _____ IS: COP FUNCTION
OPTION 30 VALUE = _____ IS: COP BONDING
OPTION 31 VALUE = _____ IS: IN INPUT LEVELS
OPTION 32 VALUE = _____ IS: G INPUT LEVELS
OPTION 33 VALUE = _____ IS: L INPUT LEVELS
OPTION 34 VALUE = _____ IS: CKO INPUT LEVELS
OPTION 35 VALUE = _____ IS: SI INPUT LEVELS

TEST MODE (Non-Standard Operation)

The SO output has been configured to provide for standard test procedures for the custom-programmed COP420. With SO forced to logic "1", two test modes are provided, depending upon the value of SI:

a. RAM and Internal Logic Test Mode (SI = 1)

b. ROM Test Mode (SI = 0)

These special test modes should not be employed by the user; they are intended for manufacturing test only.

APPLICATION # 1: COP420 General Controller

Figure 8 shows an interconnect diagram for a COP420 used as a general controller. Operation of the system is as follows:

1. The L_7–L_0 outputs are configured as LED Direct Drive outputs, allowing direct connection to the segments of the display.
2. The D_3–D_0 outputs drive the digits of the mulitplexed display directly and scan the columns of the 4 x 4 keyboard matrix.
3. The IN_3–IN_0 inputs are used to input the 4 rows of the keyboard matrix. Reading the IN lines in conjunction with the current value of the D outputs allows detection, debouncing, and decoding of any one of the 16 keyswitches.
4. CKI is configured as a single-pin oscillator input allowing system timing to be controlled by a single-pin RC network. CKO is therefore available for use as a V_R RAM power supply pin. RAM data integrity is thereby assured when the main power supply is shut down (see RAM Keep-Alive option description).
5. SI is selected as the input to a binary counter input. With SIO used as a binary counter, SO and SK can be used as general purpose outputs.
6. The 4 bidirectional G I/O ports (G_3–G_0) are available for use as required by the user's application.

APPLICATION # 2: MUSICAL ORGAN AND MUSIC BOX

Play Mode: Twenty-five musical keys and 25 LEDs are provided to denote F to F with half notes in between. All the keys and LEDs are directly detected and driven by the microprocessor. Depression of the key will give the corresponding musical note and light up the corresponding LED.

Clear: Memory is provided to store a played tune. Depression of the CLEAR key erases the memory and the microprocessor is ready to store new musical notes. A maximum of 28 notes can be stored where each note can be of one to eight musical beats. (Two bytes of memory are required to store one musical note. Any note longer than eight musical beats will require additional memory space for storage.)

Playback: Depression of this button will playback the tune stored in the memory since last "clear."

Preprogrammed Tunes: There are ten preprogrammed tunes (each has an average of 55 notes) masked in the chip. Any tune can be recalled by depressing the "Tune Button" followed by the corresponding "Sharp Key."

Learn Mode: This mode is for the player to learn the ten preprogrammed tunes. By pressing the "Learn Button" followed by the corresponding "Sharp Key," the LEDs will be lighted up one by one to indicate the notes of the selected tune. The LED will remain "on" until the player presses the correct musical key; the LED for the next note will then be lighted up.

Pause: In addition to the 25 musical keys, there is a special pause key. The depression of this key generates a blank note to the memory.

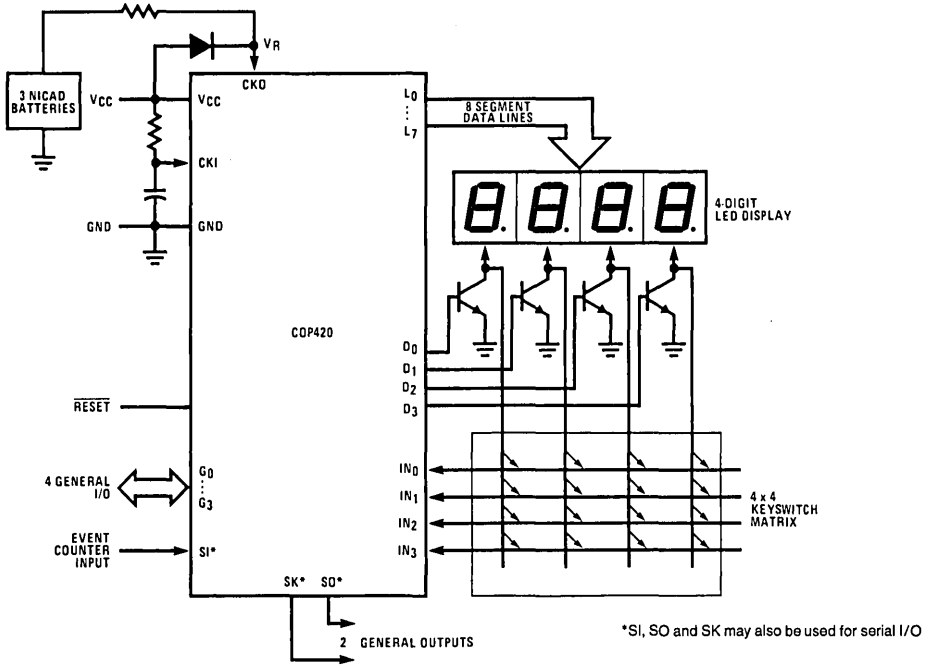
Note: In the Learn Mode when playing "Oh Susanna," the pause key must be used.

Tempo: This is a control input to the musical beat time oscillator for varying the speed of the musical tunes.

Vibrato: This is a switch control to vary the frequency vibration of the note.

Tunes Listing: The following is a listing of the ten preprogrammed tunes: 1) Jingle Bells, 2) Twinkle, Twinkle Little Star, 3) Happy Birthday, 4) Yankee Doodle, 5) Silent Night, 6) This Old Man, 7) London Bridge Is Falling Down, 8) Auld Lang Syne, 9) Oh Susanna, 10) Clementine.

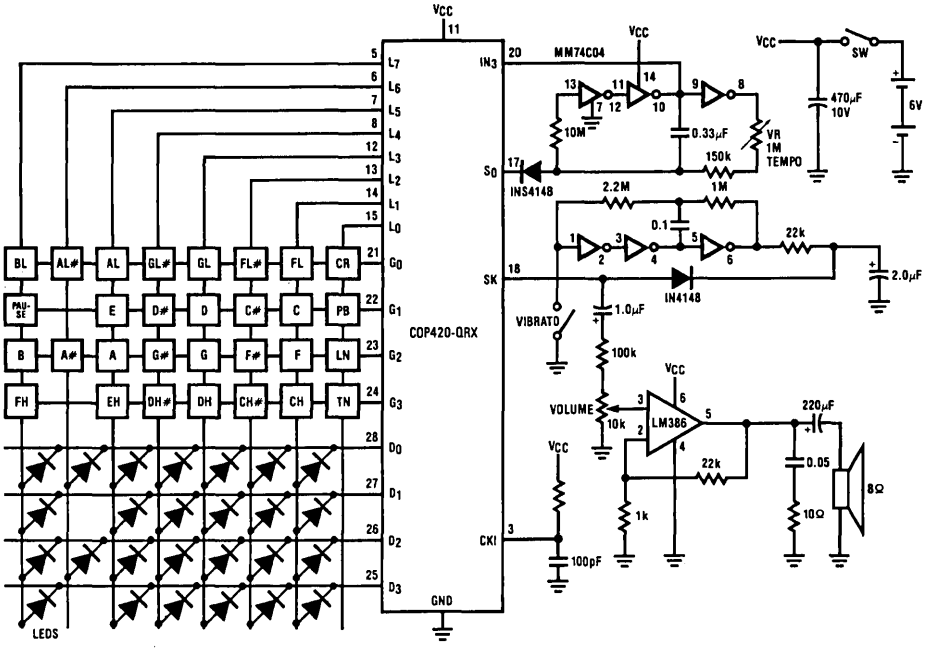
Typical Applications



TL/DD/6921-26

FIGURE 11. COP420 Keyboard Display Interface

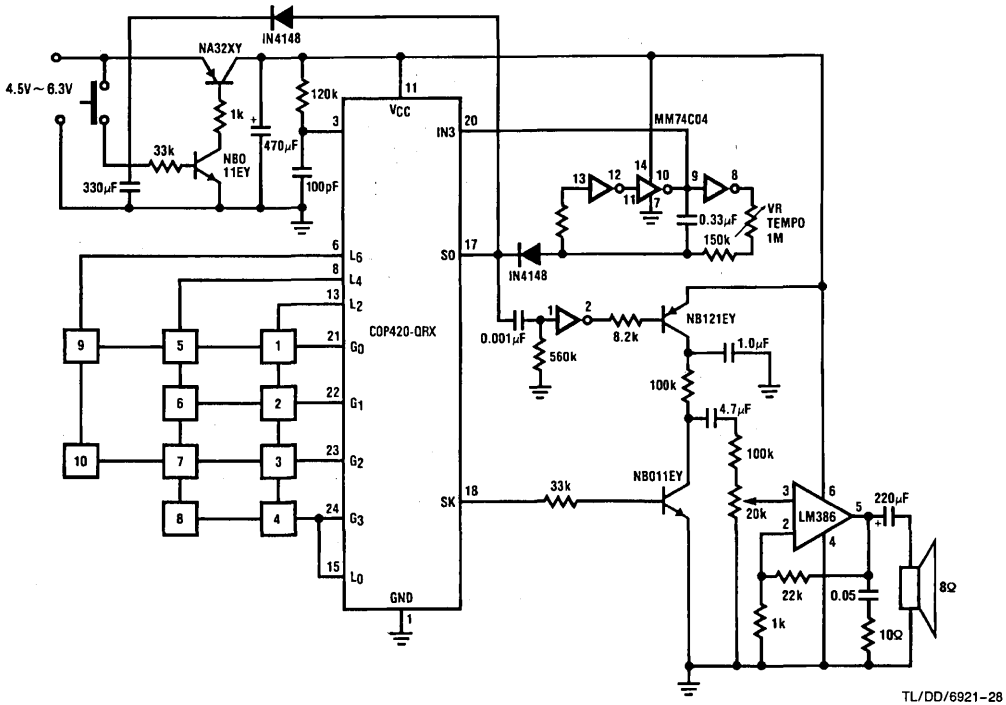
Circuit Diagram of COP420 Musical Organ



TL/DD/6921-27

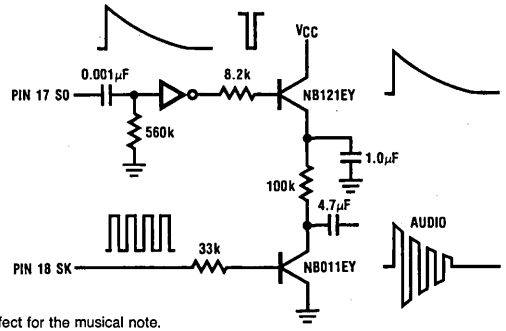
Typical Applications (Continued)

Music Box Application with Direct Key Access



TL/DD/6921-28

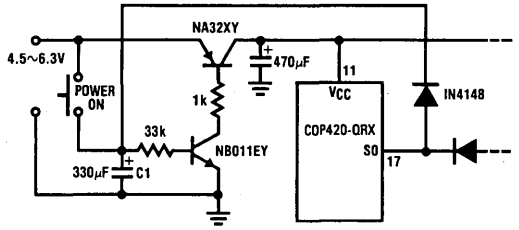
Bell Sound Circuit



This additional circuit provides tinkling effect for the musical note.

TL/DD/6921-29

Auto Power Shut-Off Circuit



TL/DD/6921-30

This circuit automatically turns off the musical organ if none of the keys are pressed within approximately 30 seconds.

National Semiconductor Corporation

COP420L/COP421L/COP422L/COP320L/COP321L/ COP322L Single-Chip N-Channel Microcontrollers

General Description

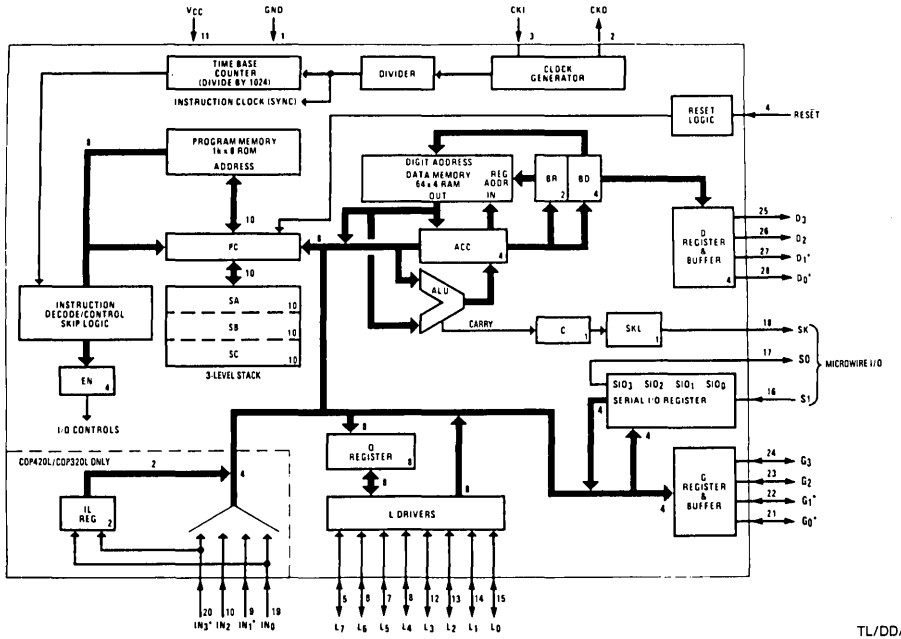
The COP420L, COP421L, COP422L, COP320L, COP321L, and COP322L Single-Chip N-Channel Microcontrollers are members of the COPS™ family, fabricated using N-channel, silicon gate MOS technology. These controller oriented processors are complete microcomputers containing all system timing, internal logic, ROM, RAM, and I/O necessary to implement dedicated control functions in a variety of applications. Features include single supply operation, a variety of output configuration options, with an instruction set, internal architecture, and I/O scheme designed to facilitate keyboard input, display output, and BCD data manipulation. The COP421L and COP422L are identical to the COP420L, but with 19 and 15 I/O lines, respectively, instead of 23. They are an appropriate choice for use in numerous human interface control environments. Standard test procedures and reliable high-density fabrication techniques provide the medium to large volume customers with a customized controller oriented processor at a low end-product cost.

The COP320L, COP321L, and COP322L are exact functional equivalents, but extended temperature range versions, of the COP420L, COP421L, and COP422L respectively.

Features

- Low cost
- Powerful instruction set
- 1k x 8 ROM, 64 x 4 RAM
- 23 I/O lines (COP420L)
- True vectored interrupt, plus restart
- Three-level subroutine stack
- 16 μs instruction time
- Single supply operation (4.5V–6.3V)
- Low current drain (8 mA max)
- Internal time-base counter for real-time processing
- Internal binary counter register with MICROWIRE™ compatible serial I/O
- General purpose and TRI-STATE® outputs
- LSTTL/CMOS compatible in and out
- Direct drive of LED digit and segment lines
- Software/hardware compatible with other members of COP400 family
- Extended temperature range device— COP320L/COP321L/COP322L (−40°C to +85°C)
- Wider supply range (4.5V–9.5V) optionally available

Block Diagram



TL/DD/8825–1

*Not available on COP422L/COP322L

FIGURE 1

COP420L/COP421L/COP422L

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin Relative to GND	−0.5V to +10V
Ambient Operating Temperature	0°C to +70°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C

Power Dissipation	
COP420L/COP421L	0.75W at 25°C 0.4W at 70°C
COP422L	0.65W at 25°C 0.3W at 70°C
Total Source Current	120 mA
Total Sink Current	120 mA

Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 9.5V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Standard Operating Voltage (V_{CC})	(Note 1)	4.5	6.3	V
Optional Operating Voltage (V_{CC})		4.5	9.5	V
Power Supply Ripple	Peak to Peak		0.5	V
Operating Supply Current	All Inputs and Outputs Open		9	mA
Input Voltage Levels				
CKI Input Levels				
Crystal Input (÷32, ÷16, ÷8)				
Logic High (V_{IH}) V_{CC} = Max		3.0		V
Logic High (V_{IH})				
V_{CC} = 5V ±5%		2.0		V
Logic Low (V_{IL})		−0.3	0.4	V
Schmitt Trigger Input (÷4)				
Logic High (V_{IH})		0.7 V_{CC}		V
Logic Low (V_{IL})		−0.3	0.6	V
$\overline{\text{RESET}}$ Input Levels	Schmitt Trigger Input			
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.6	V
SO Input Level (Test Mode)	(Note 3)	2.0	2.5	V
All Other Inputs				
Logic High	V_{CC} = Max	3.0		V
Logic High	with TTL Trip Level Options	2.0		V
Logic Low	Selected, V_{CC} = 5V ±5%	−0.3	0.8	V
Logic High	with High Trip Level Options	3.6		V
Logic Low	Selected	−0.3	1.2	V
Input Capacitance			7	pF
Hi-Z Input Leakage		−1	+1	μA
Output Voltage Levels				
LSTTL Operation	V_{CC} = 5V ±10%			
Logic High (V_{OH})	I_{OH} = −25 μA	2.7		V
Logic Low (V_{OL})	I_{OL} = 0.36 ma		0.4	V
CMOS Operation (Note 2)	V_{CC} = 4.5V			
Logic High	I_{OH} = −10 μA	V_{CC} −1		V
Logic Low	I_{OL} = +10 μA		0.2	V

Note 1: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Note 2: TRI-STATE and LED configurations are excluded.

Note 3: SO output "0" level must be less than 0.8V for normal operation.

COP420L/COP421L/COP422L

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 9.5V$ unless otherwise noted (Continued)

Parameter	Conditions	Min	Max	Units
Output Current Levels				
Output Sink Current				
SO and SK Outputs (I_{OL})	$V_{CC} = 9.5V, V_{OL} = 0.4V$	1.8		mA
	$V_{CC} = 6.3V, V_{OL} = 0.4V$	1.2		mA
	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.9		mA
L_0–L_7 Outputs and Standard	$V_{CC} = 9.5V, V_{OL} = 0.4V$	0.4		mA
G_0–G_3, D_0–D_3 Outputs (I_{OL})	$V_{CC} = 6.3V, V_{OL} = 0.4V$	0.4		mA
	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.4		mA
G_0–G_3 and D_0–D_3 Outputs with	$V_{CC} = 9.5V, V_{OL} = 1.0V$	15		mA
High Current Options (I_{OL})	$V_{CC} = 6.3V, V_{OL} = 1.0V$	11		mA
	$V_{CC} = 4.5V, V_{OL} = 1.0V$	7.5		mA
G_0–G_3 and D_0–D_3 Outputs with	$V_{CC} = 9.5V, V_{OL} = 1.0V$	30		mA
Very High Current Options (I_{OL})	$V_{CC} = 6.3V, V_{OL} = 1.0V$	22		mA
	$V_{CC} = 4.5V, V_{OL} = 1.0V$	15		mA
CKI (Single-Pin RC Oscillator)	$V_{CC} = 4.5V, V_{IH} = 3.5V$	2		mA
CKO	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.2		mA
Output Source Current				
Standard Configuration,	$V_{CC} = 9.5V, V_{OH} = 2.0V$	−140	−800	μA
All Outputs (I_{OH})	$V_{CC} = 6.3V, V_{OH} = 2.0V$	−75	−480	μA
	$V_{CC} = 4.5V, V_{OH} = 2.0V$	−30	−250	μA
Push-Pull Configuration	$V_{CC} = 9.5V, V_{OH} = 4.75V$	−1.4		mA
SO and SK Outputs (I_{OH})	$V_{CC} = 6.3V, V_{OH} = 2.4V$	−1.4		mA
	$V_{CC} = 4.5V, V_{OH} = 1.0V$	−1.2		mA
LED Configuration, L_0–L_7				
Outputs, Low Current	$V_{CC} = 9.5V, V_{OH} = 2.0V$	−1.5	−18	mA
Driver Option (I_{OH})	$V_{CC} = 6.0V, V_{OH} = 2.0V$	−1.5	−13	mA
LED Configuration, L_0–L_7				
Outputs, High Current	$V_{CC} = 9.5V, V_{OH} = 2.0V$	−3.0	−35	mA
Driver Option (I_{OH})	$V_{CC} = 6.0V, V_{OH} = 2.0V$	−3.0	−25	mA
TRI-STATE Configuration,	$V_{CC} = 9.5V, V_{OH} = 5.5V$	−0.75		mA
L_0–L_7 Outputs, Low	$V_{CC} = 6.3V, V_{OH} = 3.2V$	−0.8		mA
Current Driver Option (I_{OH})	$V_{CC} = 4.5V, V_{OH} = 1.5V$	−0.9		mA
TRI-STATE Configuration,	$V_{CC} = 9.5V, V_{OH} = 5.5V$	−1.5		mA
L_0–L_7 Outputs, High	$V_{CC} = 6.3V, V_{OH} = 3.2V$	−1.6		mA
Current Driver Option (I_{OH})	$V_{CC} = 4.5V, V_{OH} = 1.5V$	−1.8		mA
Input Load Source Current	$V_{CC} = 5.0V, V_{IL} = 0V$	−10	−140	μA
CKO Output				
RAM Power Supply Option	$V_R = 3.3V$		3.0	mA
Power Requirement				
TRI-STATE Output Leakage Current		−2.5	+2.5	μA
Total Sink Current Allowed				
All Outputs Combined			120	mA
D, G Ports			120	mA
L_7–L_4			4	mA
L_3–L_0			4	mA
All Other Pins			1.5	mA
Total Source Current Allowed				
All I/O Combined			120	mA
L_7–L_4			60	mA
L_3–L_0			60	mA
Each L Pin			30	mA
All Other Pins			1.5	mA

COP320L/COP321L/COP322L

Absolute Maximum Ratings

Voltage at Any Pin Relative to GND	−0.5V to +10V
Ambient Operating Temperature	−40°C to +85°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C
Power Dissipation	
COP320L/COP321L	0.75W at 25°C
	0.4W at 70°C
	0.25W at 85°C
COP322L	0.65W at 25°C
	0.20W at 70°C
Total Source Current	120 mA
Total Sink Current	120 mA

Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.

DC Electrical Characteristics $-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 7.5V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Standard Operating Voltage (V_{CC})	(Note 1)	4.5	5.5	V
Optional Operating Voltage (V_{CC})		4.5	7.5	V
Power Supply Ripple	Peak to Peak		0.5	V
Operating Supply Current	All Inputs and Outputs Open		11	mA
Input Voltage Levels				
CKI Input Levels				
Crystal Input				
Logic High (V_{IH}) V_{CC} = Max		3.0		V
Logic High (V_{IH})				
V_{CC} = 5V ±5%		2.2		V
Logic Low (V_{IL})		−0.3	0.3	V
Schmitt Trigger Input				
Logic High (V_{IH})		0.7 V_{CC}		V
Logic Low (V_{IL})		−0.3	0.4	V
$\overline{\text{RESET}}$ Input Levels	Schmitt Trigger Input			
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.4	V
SO Input Level (Test Mode)	(Note 3)	2.2	2.5	V
All Other Inputs				
Logic High	V_{CC} = Max	3.0		V
Logic High	with TTL Trip Level Options	2.2		V
Logic Low	Selected, V_{CC} = 5V ±5%	−0.3	0.6	V
Logic High	with High Trip Level Options	3.6		V
Logic Low	Selected	−0.3	1.2	V
Input Capacitance			7	pF
Hi-Z Input Leakage		−2	+2	μA
Output Voltage Levels				
LSTTL Operation	V_{CC} = 5V ±10%			
Logic High (V_{OH})	I_{OH} = −20 μA	2.7		V
Logic Low (V_{OL})	I_{OL} = 0.36 mA		0.4	V
CMOS Operation (Note 2)	V_{CC} = 4.5V			
Logic High	I_{OH} = −10 μA	V_{CC} −1		V
Logic Low	I_{OL} = +10 μA		0.2	V

Note 1: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Note 2: TRI-STATE and LED configurations are excluded.

Note 3: SO output "0" level must be less than 0.6V for normal operation.

COP320L/COP321L/COP322L

DC Electrical Characteristics

$-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 7.5V$ unless otherwise noted (Continued)

Parameter	Conditions	Min	Max	Units
Output Current Levels				
Output Sink Current				
SO and SK Outputs (I_{OL})	$V_{CC} = 7.5V, V_{OL} = 0.4V$	1.4		mA
	$V_{CC} = 5.5V, V_{OL} = 0.4V$	1.0		mA
	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.8		mA
L_0–L_7 Outputs and Standard	$V_{CC} = 7.5V, V_{OL} = 0.4V$	0.4		mA
G_0–G_3 and D_0–D_3 Outputs (I_{OL})	$V_{CC} = 5.5V, V_{OL} = 0.4V$	0.4		mA
	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.4		mA
G_0–G_3 and D_0–D_3 Outputs with	$V_{CC} = 7.5V, V_{OL} = 1.0V$	12		mA
High Current Options (I_{OL})	$V_{CC} = 5.5V, V_{OL} = 1.0V$	9		mA
	$V_{CC} = 4.5V, V_{OL} = 1.0V$	7		mA
G_0–G_3 and D_0–D_3 Outputs with	$V_{CC} = 7.5V, V_{OL} = 1.0V$	24		mA
Very High Current Options (I_{OL})	$V_{CC} = 5.5V, V_{OL} = 1.0V$	18		mA
	$V_{CC} = 4.5V, V_{OL} = 1.0V$	14		mA
CKI (Single-Pin RC Oscillator)	$V_{CC} = 4.5V, V_{IH} = 3.5V$	2		mA
CKO	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.2		mA
Output Source Current				
Standard Configuration,	$V_{CC} = 7.5V, V_{OH} = 2.0V$	−100	−900	μA
All Outputs (I_{OH})	$V_{CC} = 5.5V, V_{OH} = 2.0V$	−55	−600	μA
	$V_{CC} = 4.5V, V_{OH} = 2.0V$	−28	−350	μA
Push-Pull Configuration	$V_{CC} = 7.5V, V_{OH} = 3.75V$	−0.85		mA
SO and SK Outputs (I_{OH})	$V_{CC} = 5.5V, V_{OH} = 2.0V$	−1.1		mA
	$V_{CC} = 4.5V, V_{OH} = 1.0V$	−1.2		mA
LED Configuration, L_0–L_7	$V_{CC} = 7.5V, V_{OH} = 2.0V$	−1.4	−27	mA
Outputs, Low Current	$V_{CC} = 6.0V, V_{OH} = 2.0V$	−1.4	−17	mA
Driver Option (I_{OH})	$V_{CC} = 5.5V, V_{OH} = 2.0V$	−0.7	−15	mA
LED Configuration, L_0–L_7	$V_{CC} = 7.5V, V_{OH} = 2.0V$	−2.7	−54	mA
Outputs, High Current	$V_{CC} = 6.0V, V_{OH} = 2.0V$	−2.7	−34	mA
Driver Option (I_{OH})	$V_{CC} = 5.5V, V_{OH} = 2.0V$	−1.4	−30	mA
TRI-STATE Configuration,	$V_{CC} = 7.5V, V_{OH} = 4.0V$	−0.7		mA
L_0–L_7 Outputs, Low	$V_{CC} = 5.5V, V_{OH} = 2.7V$	−0.6		mA
Current Driver Option (I_{OH})	$V_{CC} = 4.5V, V_{OH} = 1.5V$	−0.9		mA
TRI-STATE Configuration,	$V_{CC} = 7.5V, V_{OH} = 4.0V$	−1.4		mA
L_0–L_7 Outputs, High	$V_{CC} = 5.5V, V_{OH} = 2.7V$	−1.2		mA
Current Driver Option (I_{OH})	$V_{CC} = 4.5V, V_{OH} = 1.5V$	−1.8		mA
Input Load Source Current	$V_{CC} = 5.0V, V_{IL} = 0V$	−10	−200	μA
CKO Output				
RAM Power Supply Option	$V_R = 3.3V$		4.0	mA
Power Requirement				
TRI-STATE Output Leakage Current		−5	+5	μA
Total Sink Current Allowed				
All Outputs Combined			120	mA
D, G Ports			120	mA
L_7–L_4			4	mA
L_3–L_0			4	mA
All Other Pins			1.5	mA
Total Source Current Allowed				
All I/O Combined			120	mA
L_7–L_4			60	mA
L_3–L_0			60	mA
Each L Pin			30	mA
All Other Pins			1.5	mA

AC Electrical Characteristics

COP420L/COP421L/COP422L: $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 9.5V$ unless otherwise noted

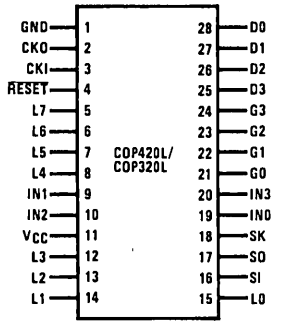
COP320L/COP321L/COP322L: $-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 7.5V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time—t_C		16	40	μs
CKI				
Input Frequency—f_I	÷32 Mode	0.8	2.0	MHz
	÷16 Mode	0.4	1.0	MHz
	÷8 Mode	0.2	0.5	MHz
	÷4 Mode	0.1	0.25	MHz
Duty Cycle		30	60	%
Rise Time	f_I = 2 MHz		120	ns
Fall Time			80	ns
CKI Using RC (÷4)	R = 56 kΩ ±5% C = 100 pF ±10%			
Instruction Cycle Time (Note 1)		16	28	μs
CKO as SYNC Input				
t_{SYNC}		400		ns
INPUTS:				
IN_3–IN_0, G_3–G_0, L_7–L_0				
t_{SETUP}		8.0		μs
t_{HOLD}		1.3		μs
SI				
t_{SETUP}		2.0		μs
t_{HOLD}		1.0		μs
OUTPUT PROPAGATION DELAY	Test Condition: C_L = 50 pF, R_L = 20 kΩ, V_{OUT} = 1.5V			
SO, SK Outputs				
t_{pd1}, t_{pd0}			4.0	μs
All Other Outputs				
t_{pd1}, t_{pd0}			5.6	μs

Note 1: Variation due to the device included.

Connection Diagrams

Dual-In Line Package

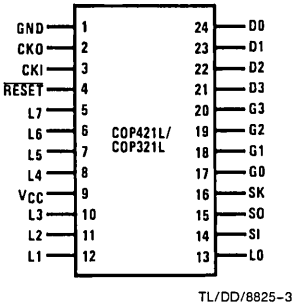


TL/DD/8825-2

Top View

Order Number COP420L-XXX/N or COP320L-XXX/N
See NS Package Number N28B

Dual-In Line Package

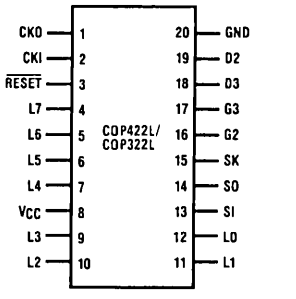


TL/DD/8825-3

Top View

Order Number COP421L-XXX/M, COP421L-XXX/N, COP321LXXX-/M or COP321L-XXX/N,
See NS Package Number M20B or N20A

Dual-In Line Package



TL/DD/8825-4

Top View

Order Number COP422L-XXX/M, COP422L-XXX/N, COP322L-XXX/M or COP322L-XXX/N
See NS Package Number M24B or N24A

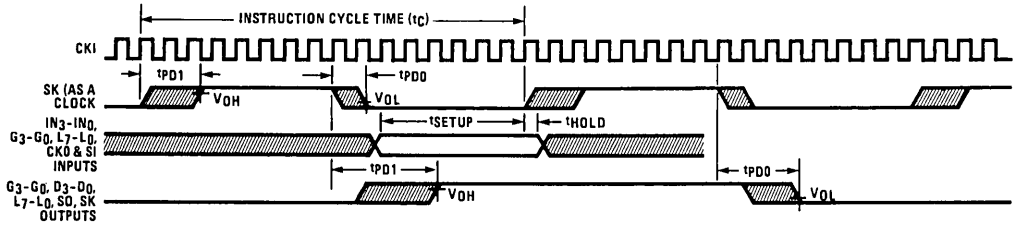
FIGURE 2

Pin Descriptions

Pin	Description
L_7–L_0	8 bidirectional I/O ports with TRI-STATE
G_3–G_0	4 bidirectional I/O ports
D_3–D_0	4 general purpose outputs
IN_3–IN_0	4 general purpose inputs (COP420L only)
SI	Serial input (or counter input)
SO	Serial output (or general purpose output)

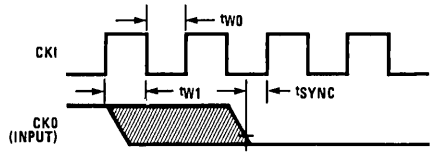
Pin	Description
SK	Logic-controlled clock (or general purpose output)
CKI	System oscillator input
CKO	System oscillator output (or general purpose input, RAM power supply or SYNC input)
$\overline{RESET}$	System reset input
V_{CC}	Power supply
GND	Ground

Timing Diagrams



TL/DD/8825-5

FIGURE 3. Input/Output Timing Diagrams (Crystal Divide-by-16 Mode)



TL/DD/8825-6

FIGURE 3a. Synchronization Timing

Functional Description

For ease of reading this description, only COP420L and/or COP421L are referenced; however, all such references apply also to COP320L, COP321L, COP322L, or COP422L.

A block diagram of the COP420L is given in *Figure 1.* Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1" (greater than 2V). When a bit is reset, it is a logic "0" (less than 0.8V).

PROGRAM MEMORY

Program Memory consists of a 1,024 byte ROM. As can be seen by an examination of the COP420L/421L instruction set, these words may be program instructions, program data or ROM addressing data. Because of the special characteristics associated with the JP, JSRP, JID and LQID instructions, ROM must often be thought of as being organized into 16 pages of 64 words each.

ROM addressing is accomplished by a 10-bit PC register. Its binary value selects one of the 1,024 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 10-bit binary count value. Three levels of subroutine nesting are implemented by the 10-bit subroutine save registers, SA, SB and SC, providing a last-in, first-out (LIFO) hardware subroutine stack.

ROM instruction words are fetched, decoded and executed by the Instruction Decode, Control and Skip Logic circuitry.

DATA MEMORY

Data memory consists of a 256-bit RAM, organized as 4 data registers of 16 4-bit digits. RAM addressing is implemented by a 6-bit B register whose upper 2 bits (Br) select 1 of 4 data registers and lower 4 bits (Bd) select 1 of 16 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) is usually loaded into or from, or exchanged with, the A register (accumulator), it may also be loaded into or from the Q latches or loaded from the L ports. RAM addressing may also be performed directly by the LDD and XAD instructions is based upon the 6-bit contents of the operand field of these instructions. The Bd register also serves as a source register for 4-bit data sent directly to the D outputs.

INTERNAL LOGIC

The 4-bit A register (accumulator) is the source and destination register for most I/O, arithmetic, logic and data memory access operations. It can also be used to load the Br and Bd portions of the B register, to load and input 4 bits of the 8-bit Q latch data, to input 4 bits of the 8-bit L I/O port data and to perform data exchanges with the SIO register.

A 4-bit adder performs the arithmetic and logic functions of the COP420/421L, storing its results in A. It also outputs a carry bit to the 1-bit C register, most often employed to indicate arithmetic overflow. The C register, in conjunctions with the XAS instruction and the EN register, also serves to control the SK output. C can be outputted directly to SK or can enable SK to be a sync clock each instruction cycle time. (See XAS instruction and EN register description, below.)

Four general-purpose inputs, IN_3–IN_0, are provided.

The D register provides 4 general-purpose outputs and is used as the destination register for the 4-bit contents of Bd. The D outputs can be directly connected to the digits of a multiplexed LED display.

The G register contents are outputs to 4 general-purpose bidirectional I/O ports. G I/O ports can be directly connected to the digits of a multiplexed LED display.

The Q register is an internal, latched, 8-bit register, used to hold data loaded to or from M and A, as well as 8-bit data from ROM. Its contents are outputted to the L I/O ports when the L drivers are enabled under program control. (See LEI instruction.)

The 8 L drivers, when enabled, output the contents of latched Q data to the L I/O ports. Also, the contents of L may be read directly into A and M. L I/O ports can be directly connected to the segments of a multiplexed LED display (using the LED Direct Drive output configuration option) with Q data being outputted to the Sa–Sg and decimal point segments of the display.

The SIO register functions as a 4-bit serial-in/serial-out shift register or as a binary counter depending on the contents of the EN register. (See EN register description, below.) Its contents can be exchanged with A, allowing it to input or output a continuous serial data stream. SIO may also be used to provide additional parallel I/O by connecting SO to external serial-in/parallel-out shift registers. For example of additional parallel output capacity see Application #2.

The XAS instruction copies C into the SKL latch. In the counter mode, SK is the output of SKL; in the shift register mode, SK outputs SKL ANDed with the clock.

The EN register is an internal 4-bit register loaded under program control by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register (EN_3–EN_0).

1. The least significant bit of the enable register, EN_0, selects the SIO register as either a 4-bit shift register or a 4-bit binary counter. With EN_0 set, SIO is an asynchronous binary counter, *decrementing* its value by one upon each low-going pulse ("1" to "0") occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output is equal to the value of EN_3. With EN_0 reset, SIO is a serial shift register shifting left each instruction cycle time. The data present at SI goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. (See 4 below.) The SK output becomes a logic-controlled clock.
2. With EN_1 set the IN_1 input is enabled as an interrupt input. Immediately following an interrupt, EN_1 is reset to disable further interrupts.
3. With EN_2 set, the L drivers are enabled to output the data in Q to the L I/O ports. Resetting EN_2 disables

Functional Description (Continued)

the L drivers, placing the L I/O ports in a high-impedance input state.

4. EN_3, in conjunction with EN_0, affects the SO output. With EN_0 set (binary counter option selected) SO will output the value loaded into EN_3. With EN_0 reset (serial shift register option selected), setting EN_3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN_3 with the serial shift register option selected disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains reset to "0". The table below provides a summary of the modes associated with EN_3 and EN_0.

Enable Register Modes—Bits EN_3 and EN_0

EN_3	EN_0	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL = 1, SK = Clock If SKL = 0, SK = 0
1	0	Shift Register	Input to Shift Register	Serial Out	If SKL = 1, SK = Clock If SKL = 0, SK = 0
0	1	Binary Counter	Input to Binary Counter	0	If SKL = 1, SK = 1 If SKL = 0, SK = 0
1	1	Binary Counter	Input to Binary Counter	1	If SKL = 1, SK = 1 If SKL = 0, SK = 0

INTERRUPT

The following features are associated with the IN_1 interrupt procedure and protocol and must be considered by the programmer when utilizing interrupts.

a. The interrupt, once aknowledged as explained below, pushes the next sequential program counter address (PC + 1) onto the stack, pushing in turn the contents of the other subroutine-save registers to the next lower level (PC + 1 → SA → SB → SC). Any previous contents of SC are lost. The program counter is set to hex address 0FF (the last word of page 3) and EN_1 is reset.

b. An interrupt will be acknowledged only after the following conditions are met:

   1. EN_1 has been set.
   2. A low-going pulse ("1" to "0") at least two instruction cycles wide occurs on the IN_1 input.
   3. A currently executing instruction has been completed.
   4. All successive transfer of control instructions and successive LBIs have been completed (e.g., if the main program is executing a JP instruction which transfers program control to another JP instruction, the interrupt will not be acknowledged until the second JP instruction has been executed).

c. Upon acknowledgement of an interrupt, the skip logic status is saved and later restored upon popping of the stack. For example if an interrupt occurs during the execution of ASC (Add with Carry, Skip on Carry) instruction which results in carry, the skip logic status is saved and program control is transferred to the interrupt servicing routine at address 0FF. At the *end* of the interrupt routine, a RET instruction is executed to "pop" the stack and return program control to the instruction following the original ASC. *At this time,* the skip logic is enabled and skips this instruction because of the previous ASC carry. Subroutines and LQID instructions should not be nested within the interrupt servicing routine since their popping the stack will enable any previously saved main program skips, interfering with the orderly execution of the interrupt routine.

d. The first instruction of the interrupt routine at hex address 0FF must be a NOP.

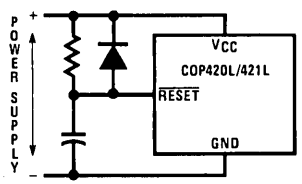
e. A LEI instruction can be put immediately before the RET to re-enable interrupts.

INITIALIZATION

The Reset Logic will initialize (clear) the device upon power-up if the power supply rise time is less than 1 ms and greater than 1 μs. If the power supply rise time is greater than 1 ms, the user must provide an external RC network and diode to the $\overline{\text{RESET}}$ pin as shown below. The $\overline{\text{RESET}}$ pin is configured as a Schmitt trigger input. If not used it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the $\overline{\text{RESET}}$ input, provided it stays low for at least three instruction cycle times.

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN, and G registers are cleared. The SK output is enabled as a SYNC output, providing a pulse each instruction cycle time. *Data Memory (RAM) is not cleared upon initialization.* The first instruction at address 0 must be a CLRA.

Power-Up Clear Circuit



TL/DD/8825-7

RC ≥ 5 × Power Supply Rise Time

Functional Description (Continued)

OSCILLATOR

There are three basic clock oscillator configurations available as shown by *Figure 4.*

a. **Crystal Controlled Oscillator.** CKI and CKO are connected to an external crystal. The instruction cycle time equals the crystal frequency divided by 32 (optional by 16 or 8).

b. **External Oscillator.** CKI is an external clock input signal. The external frequency is divided by 32 (optional by 16 or 8) to give the instruction cycle time. CKO is now available to be used as the RAM power supply (V_R) or as a general purpose input.

c. **RC Controlled Oscillator.** CKI is configured as a single pin RC controlled Schmitt trigger oscillator. The instruction cycle equals the oscillation frequency divided by 4. CKO is available as the RAM power supply (V_R) or as a general purpose input.

CKO PIN OPTIONS

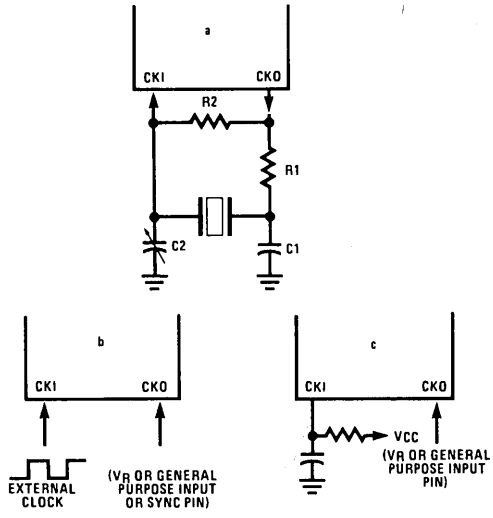
In a crystal controlled oscillator system, CKO is used as an output to the crystal network. As an option CKO can be a general purpose input, read into bit 2 of A (accumulator) upon execution of an INIL instruction. As another option, CKO can be a RAM power supply pin (V_R), allowing its connection to a standby/backup power supply to maintain the integrity of RAM data with minimum power drain when the main supply is inoperative or shut down to conserve power. Using either option is appropriate in applications where the COP420L/421L system timing configuration does not require use of the CKO pin.

RAM KEEP-ALIVE OPTION (Not available on COP422L)

Selecting CKO as the RAM power supply (V_R) allows the user to shut off the chip power supply (V_{CC}) and maintain data in the RAM. To insure that RAM data integrity is maintained, the following conditions must be met:

1. $\overline{\text{RESET}}$ must go low before V_{CC} goes below spec during power-off; V_{CC} must be within spec before $\overline{\text{RESET}}$ goes high on power-up.
2. During normal operation V_R must be within the operating range of the chip, with $(V_{CC} - 1) \leq V_R \leq V_{CC}$.
3. V_R must be $\geq$ 3.3V with V_{CC} off.

Crystal Oscillator



TL/DD/8825–8

Crystal Value	Component Values			
	R1 (Ω)	R2 (Ω)	C1 (pF)	C2 (pF)
455 kHz	4.7k	1M	220	220
2.097 MHz	1k	1M	30	6–36

RC Controlled Oscillator

R (kΩ)	C (pF)	Instruction Cycle Time (μs)
51	100	19 ±15%
82	56	19 ± 13%

Note: 200k ≥ R ≥ 25k
360 pF ≥ C ≤ 50 pF

FIGURE 4. COP420L/421L Oscillator

Functional Description (Continued)

I/O OPTIONS

COP420L/421L outputs have the following optional configurations, illustrated in *Figure 5*:

a. **Standard**—an enhancement mode device to ground in conjunction with a depletion-mode device to V_{CC}, compatible with LSTTL and CMOS input requirements. Available on SO, SK, and all D and G outputs.

b. **Open-Drain**—an enhancement-mode device to ground only, allowing external pull-up as required by the user's application. Available on SO, SK, and all D and G outputs.

c. **Push-Pull**—An enhancement-mode device to ground in conjunction with a depletion-mode device paralleled by an enhancement-mode device to V_{CC}. This configuration has been provided to allow for fast rise and fall times when driving capacitive loads. Available on SO and SK outputs only.

d. **Standard L**—same as **a.**, but may be disabled. Available on L outputs only.

e. **Open Drain L**—same as **b.**, but may be disabled. Available on L outputs only.

f. **LED Direct Drive**—an enhancement-mode device to ground and to V_{CC}, meeting the typical current sourcing requirements of the segments of an LED display. The sourcing device is clamped to limit current flow. These devices may be turned off under program control (see Functional Description, EN Register), placing the outputs in a high-impedance state to provide required LED segment blanking for a multiplexed display. Available on L outputs only.

g. **TRI-STATE Push-Pull**—an enhancement-mode device to ground and V_{CC}. These outputs are TRI-STATE outputs, allowing for connection of these outputs to a data bus shared by other bus drivers. Available on L outputs only.

COP420L/COP421L inputs have the following optional configurations:

h. An on-chip depletion load device to V_{CC}.

i. A Hi-Z input which must be driven to a "1" or "0" by external components.

The above input and output configurations share common enhancement-mode and depletion-mode devices. Specifically, all configurations use one or more of six devices (numbered 1–6, respectively). Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figure 6* for each of these devices to allow the designer to effectively use these I/O configurations in designing a COP420L/421L system.

The SO, SK outputs can be configured as shown in **a.**, **b.**, or **c.** The D and G outputs can be configured as shown in **a.** or **b.** Note that when inputting data to the G ports, the G outputs should be set to "1". The L outputs can be configured as in **d.**, **e.**, **f.** or **g.**

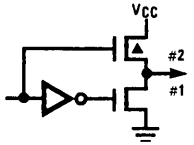
An important point to remember if using configuration **d.** or **f.** with the L drivers is that even when the L drivers are disabled, the depletion load device will source a small amount of current (see *Figure 6*, device 2); however, when the L lines are used as inputs, the disabled depletion device *cannot* be relied on to source sufficient current to pull an input to a logic 1.

COP421L

If the COP420L is bonded as a 24-pin device, it becomes the COP421L, illustrated in *Figure 2*, COP420L/421L Connection Diagrams. Note that the COP421L does not contain the four general purpose IN inputs (IN_3–IN_0). Use of this option precludes, of course, use of the IN options and the interrupt feature. All other options are available for the COP421L.

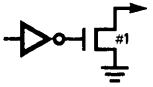
COP422L

If the COP421L is bonded as a 20-pin device, it becomes the COP422L, as illustrated in *Figure 2*. Note that the COP422L contains all the COP421L pins except D_0, D_1, G_0, and G_1. COP422L also does not allow RAM power supply input as a valid CKO pin option.



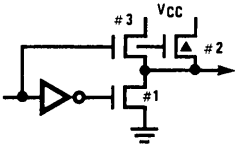
TL/DD/8825-9

a. Standard Output



TL/DD/8825-10

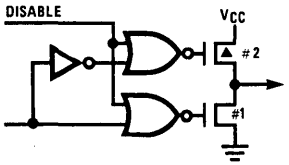
b. Open-Drain Output



TL/DD/8825-11

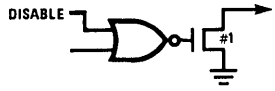
c. Push-Pull Output

Functional Description (Continued)



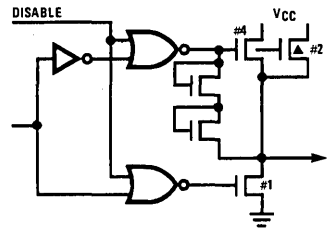
TL/DD/8825-12

d. Standard L Output



TL/DD/8825-13

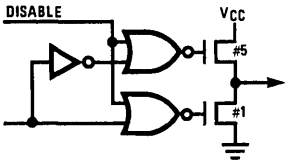
e. Open-Drain L Output



TL/DD/8825-14

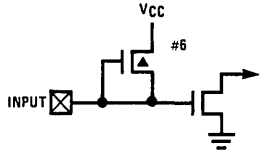
(▲ is Depletion Device)

f. LED (L Output)



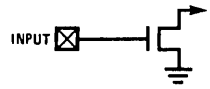
TL/DD/8825-15

g. TRI-STATE Push-Pull (L Output)



TL/DD/8825-16

h. Input with Load



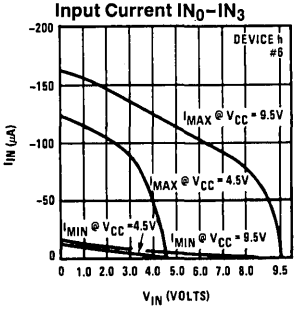
TL/DD/8825-17

I. HI-Z Input

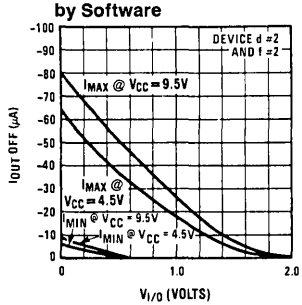
FIGURE 5. Output Configurations

Typical Performance Characteristics

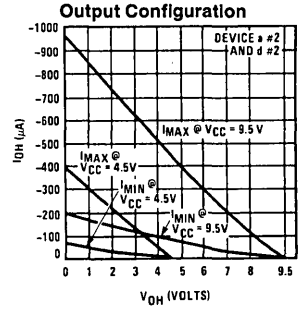
Input Current IN_0–IN_3



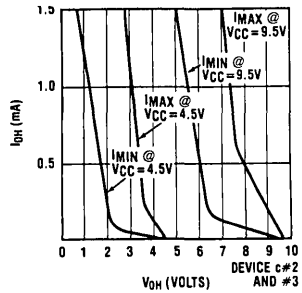
Input Current for L_0–L_7 when Output Programmed OFF by Software



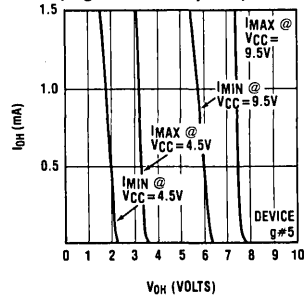
Source Current for Standard Output Configuration



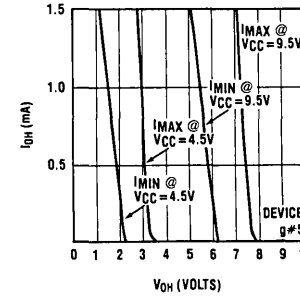
Source Current for SO and SK in Push-Pull Configuration



Source Current for L_0–L_7 in TRI-STATE Configuration (High Current Option)



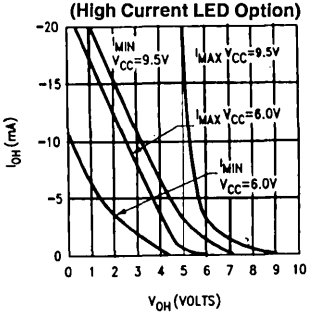
Source Current for L_0–L_7 in TRI-STATE Configuration (Low Current Option)



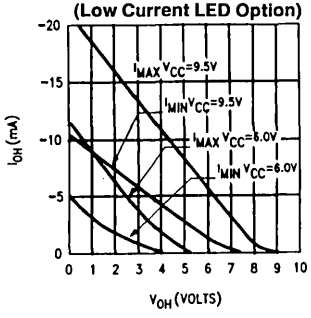
TL/DD/8825-18

Typical Performance Characteristics (Continued)

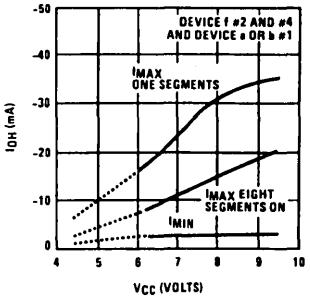
LED Output Source Current (High Current LED Option)



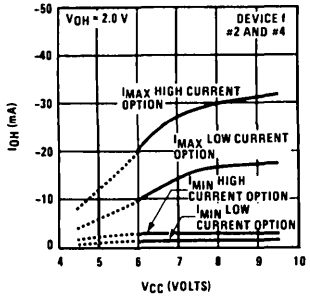
LED Output Source Current (Low Current LED Option)



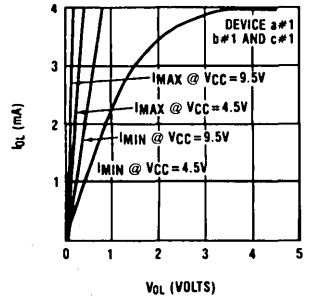
LED Output Direct Segment and Digit Drive (High Current Options on L_0–L_7; Very High Current Options on D_0–D_3 or G_0–G_3)



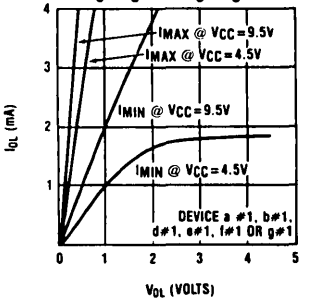
LED Output Direct Segment Drive



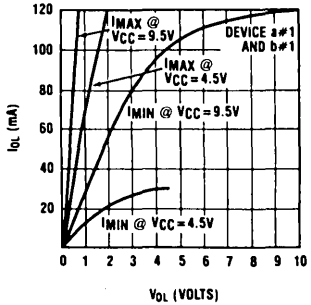
Output Sink Current for SO and SK



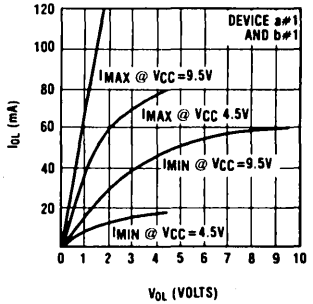
Output Sink Current for L_0–L_7 and Standard Drive Option for D_0–D_3 and G_0–G_3



Output Sink Current for G_0–G_3 and D_0–D_3 with Very High Current Option



Output Sink Current for G_0–G_3 and D_0–D_3 (High Current Option)



TL/DD/8825–19

FIGURE 6. COP420L/COP421L/COP422L Input/Output Characteristics

Typical Performance Characteristics (Continued)

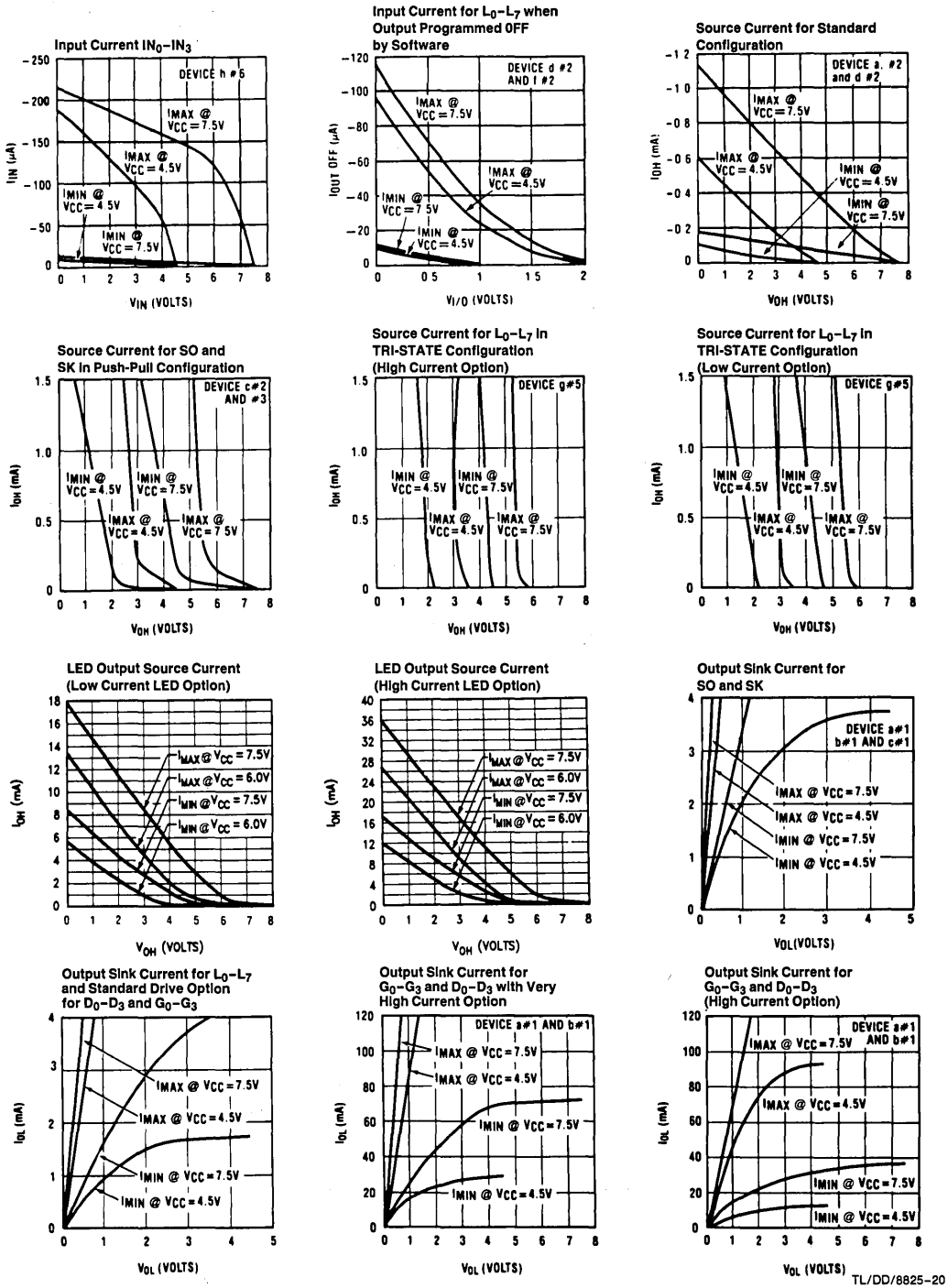


FIGURE 7. COP320L/DOP321L/COP322L Input/Output Characteristics

COP420L/COP421L Instruction Set

Table I is a symbol table providing internal architecture, instruction operand and operational symbols used in the instruction set table.

Table II provides the mnemonic, operand, machine code, data flow, skip conditions and description associated with each instruction in the COP410L/411L instruction set.

TABLE I. COP420L/421L Instruction Set Table Symbols

Symbol	Definition
INTERNAL ARCHITECTURE SYMBOLS	
A	4-bit Accumulator
B	6-bit RAM Address Register
Br	Upper 2 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry Register
D	4-bit Data Output Port
EN	4-bit Enable Register
G	4-bit Register to latch data for G I/O Port
IL	Two 1-bit Latches associated with the IN_3 or IN_0 inputs
IN	4-bit Input Port
L	8-bit TRI-STATE I/O Port
M	4-bit contents of RAM Memory pointed to by B Register
PC	10-bit ROM Address Register (program counter)
Q	8-bit Register to latch data for L I/O Port
SA	10-bit Subroutine Save Register A
SB	10-bit Subroutine Save Register B
SC	10-bit Subroutine Save Register C
SIO	4-bit Shift Register and Counter
SK	Logic-Controlled Clock Output

Symbol	Definition
INSTRUCTION OPERAND SYMBOLS	
d	4-bit Operand Field, 0–15 binary (RAM Digit Select)
r	2-bit Operand Field, 0–3 binary (RAM Register Select)
a	10-bit Operand Field, 0–1023 binary (ROM Address)
y	4-bit Operand Field, 0–15 binary (Immediate Data)
RAM(s)	Contents of RAM location addressed by s
ROM(t)	Contents of ROM location addressed by t
OPERATIONAL SYMBOLS	
+	Plus
−	Minus
→	Replaces
⟷	Is exchanged with
=	Is equal to
$\overline{A}$	The ones complement of A
⊕	Exclusive-OR
:	Range of values

Instruction Set (Continued)

TABLE II. COP420L/421L Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	0011 | 0000	A + C + RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	0011 | 0001	A + RAM(B) → A	None	Add RAM to A
ADT		4A	0100 | 1010	A + 10_{10} → A	None	Add Ten to A
AISC	y	5–	0101 | y	A + y → A	Carry	Add Immediate, Skip on Carry (y ≠ 0)
CASC		10	0001 | 0000	$\overline{A}$ + RAM(B) + C → A Carry → C	Carry	Compliment and Add with Carry, Skip on Carry
CLRA		00	0000 | 0000	0 → A	None	Clear A
COMP		40	0100 | 0000	$\overline{A}$ → A	None	Ones complement of A to A
NOP		44	0100 | 0100	None	None	No Operation
RC		32	0011 | 0010	"0" → C	None	Reset C
SC		22	0010 | 0010	"1" → C	None	Set C
XOR		02	0000 | 0010	A ⊕ RAM(B) → A	None	Exclusive-OR RAM with A
TRANSFER OF CONTROL INSTRUCTIONS						
JID		FF	1111 | 1111	ROM ($PC_{9:8}$, A,M) → $PC_{7:0}$	None	Jump Indirect (Note 3)
JMP	a	6– – –	0110 | 00 | $a_{9:8}$ $a_{7:0}$	a → PC	None	Jump
JP	a	– –	1 | $a_{6:0}$ (pages 2,3 only) or	a → $PC_{6:0}$	None	Jump within Page (Note 4)
		– –	11 | $a_{5:0}$ (all other pages)	a → $PC_{5:0}$		
JSRP	a	– –	10 | $a_{5:0}$	PC + 1 → SA → SB → SC 0010 → $PC_{9:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 5)
JSR	a	6– – –	0110 | 10 | $a_{9:8}$ $a_{7:0}$	PC + 1 → SA → SB → SC a → PC	None	Jump to Subroutine
RET		48	0100 | 1000	SC → SB → SA → PC	None	Return from Subroutine
RETSK		49	0100 | 1001	SC → SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip

Instruction Set (Continued)

TABLE II. COP420L/421L Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
MEMORY REFERENCE INSTRUCTIONS						
CAMQ		33 3C	|0011|0011| |0011|1100|	$A \rightarrow Q_{7:4}$ $RAM(B) \rightarrow Q_{3:0}$	None	Copy A, RAM to Q
CQMA		33 2C	|0011|0011| |0010|1100|	$Q_{7:4} \rightarrow RAM(B)$ $Q_{3:0} \rightarrow A$	None	Copy Q to RAM, A
LD	r	-5	|00|r|0101|	$RAM(B) \rightarrow A$ $Br \oplus r \rightarrow Br$	None	Load RAM into A, Exclusive-OR Br with r
LDD	r,d	23 --	|0010|0011| |00|r|d|	$RAM(r,d) \rightarrow A$	None	Load A with RAM pointed to directly by r,d
LQID		BF	|1011|1111|	$ROM(PC_{9:8},A,M) \rightarrow Q$ $SB \rightarrow SC$	None	Load Q Indirect (Note 3)
RMB	0 1 2 3	4C 45 42 43	|0100|1100| |0100|0101| |0100|0010| |0100|0011|	$0 \rightarrow RAM(B)_0$ $0 \rightarrow RAM(B)_1$ $0 \rightarrow RAM(B)_2$ $0 \rightarrow RAM(B)_3$	None	Reset RAM Bit
SMB	0 1 2 3	4D 47 46 4B	|0100|1101| |0100|1101| |0100|0110| |0100|1011|	$1 \rightarrow RAM(B)_0$ $1 \rightarrow RAM(B)_1$ $1 \rightarrow RAM(B)_2$ $1 \rightarrow RAM(B)_3$	None	Set RAM Bit
STII	y	7-	|0111|y|	$y \rightarrow RAM(B)$ $Bd + 1 \rightarrow Bd$	None	Store Memory Immediate and Increment Bd
X	r	-6	|00|r|0110|	$RAM(B) \longleftrightarrow A$ $Br \oplus r \rightarrow Br$	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	r,d	23 --	|0010|0011| |10|r|d|	$RAM(r,d) \longleftrightarrow A$	None	Exchange A with RAM pointed to directly by (r,d)
XDS	r	-7	|00|r|0111|	$RAM(B) \longleftrightarrow A$ $Bd - 1 \rightarrow Bd$ $Br \oplus r \rightarrow Br$	Bd decrements past 0	Exchange RAM with A and Decrement Bd, Exclusive-OR Br with r
XIS	r	-4	|00|r|0100|	$RAM(B) \longleftrightarrow A$ $Bd + 1 \rightarrow Bd$ $Br \oplus r \rightarrow Br$	Bd increments past 15	Exchange RAM with A and Increment Bd, Exclusive-OR Br with r

Instruction Set (Continued)

TABLE II. COP420L/421L Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	0101 | 0000	A → Bd	None	Copy A to Bd
CBA		4E	0100 | 1110	Bd → A	None	Copy Bd to A
LBI	r,d	– –	00 | r | (d−1) (d=0,9:15) or	r,d → B	Skip until not an LBI	Load B Immediate with r,d (Note 6)
		33	0011 | 0011			
		– –	10 | r | d (any d)			
LEI	y	33	0011 | 0011	y → EN	None	Load EN Immediate (Note 7)
		6–	0110 | y			
XABR		12	0001 | 0010	A ⟷ Br (0,0 → A_3,A_2)	None	Exchange A with Br
TEST INSTRUCTIONS						
SKC		20	0010 | 0000		C = "1"	Skip if C is True
SKE		21	0010 | 0001		A = RAM(B)	Skip if A Equals RAM
SKGZ		33	0011 | 0011		$G_{3:0}$ = 0	Skip if G is Zero (all 4 bits)
		21	0010 | 0001			
SKGBZ		33	0011 | 0011	1st byte		Skip if G Bit is Zero
	0	01	0000 | 0001	2nd byte	G_0 = 0	
	1	11	0001 | 0001		G_1 = 0	
	2	03	0000 | 0011		G_2 = 0	
	3	13	0001 | 0011		G_3 = 0	
SKMBZ	0	01	0000 | 0001		$RAM(B)_0$ = 0	Skip if RAM Bit is Zero
	1	11	0001 | 0001		$RAM(B)_1$ = 0	
	2	03	0000 | 0011		$RAM(B)_2$ = 0	
	3	13	0001 | 0011		$RAM(B)_3$ = 0	
SKT		41	0100 | 0001		A time-base counter carry has occurred since last test	Skip on Timer (Note 3)

Instruction Set (Continued)

TABLE II. COP420L/421L Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
INPUT/OUTPUT INSTRUCTIONS						
ING		33 2A	0011 | 0011 0010 | 1010	G → A	None	Input G Ports to A
ININ		33 28	0011 | 0011 0010 | 1000	IN → A	None	Input IN Inputs to A (Note 2)
INIL		33 29	0011 | 0011 0010 | 1001	IL_3, CKO, "0", IL_0 → A	None	Input IL Latches to A (Note 3)
INL		33 2E	0011 | 0011 0010 | 1110	$L_{7:4}$ → RAM(B) $L_{3:0}$ → A	None	Input L Ports to RAM, A
OBD		33 3E	0011 | 0011 0011 | 1110	Bd → D	None	Output Bd to D Outputs
OGI	y	33 5–	0011 | 0011 0101 | y	y → G	None	Output to G Ports Immediate
OMG		33 3A	0011 | 0011 0011 | 1010	RAM(B) → G	None	Output RAM to G Ports
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO (Note 3)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicitly defined (e.g., Br and Bd are explicitly defined). Bits are numbered 0 to N where 0 signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit A register.

Note 2: The ININ instruction is only available on the 28-pin COP420L as the other devices do not contain the IN inputs.

Note 3: For additional information on the operation of the XAS, JID, LQID, INIL, and SKT instructions, see below.

Note 4: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 5: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 6: LBI is a single-byte instruction if d = 0, 9, 10, 11, 12, 13, 14, or 15. The machine code for the lower 4 bits equals the binary value of the "d" data *minus 1*, e.g., to load the lower four bits of B (Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Note 7: Machine code for operand field y for LEI instruction should equal the binary value to be latched into EN, where a "1" or "0" in each bit of EN corresponds with the selection or deselection of a particular function associated with each bit. (See Functional Description, EN Register.)

Description of Selected Instructions

The following information is provided to assist the user in understanding the operation of several unique instructions and to provide notes useful to programmers in writing COP420L/421L programs.

XAS INSTRUCTION

XAS (Exchange A with SIO) exchanges the 4-bit contents of the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. An XAS instruction will also affect the SK output. (See Functional Description, EN Register, above.) If SIO is selected as a shift register, an XAS instruction must be performed once every 4 instruction cycles to effect a continuous data stream.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower 8 bits of the ROM address register PC with the *contents* of ROM addressed by the 10-bit word, $PC_{9:8}$, A, M. PC_9 and PC_8 are not affected by this instruction.

Note that JID requires 2 instruction cycles to execute.

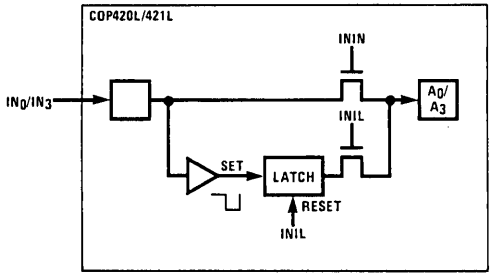
Description of Selected Instructions (Continued)

INIL INSTRUCTION

INIL (Input IL Latches to A) inputs 2 latches, IL_3 and IL_0 (see *Figure 8*) and CKO into A. The IL_3 and IL_0 latches are set if a low-going pulse ("1" to "0") has occurred on the IN_3 and IN_0 inputs since the last INIL instruction, provided the input pulse stays low for at least two instruction times. Execution of an INIL inputs IL_3 and IL_0 into A3 and A0 respectively, and resets these latches to allow them to respond to subsequent low-going pulses on the IN_3 and IN_0 lines. If CKO is mask programmed as a general purpose input, an INIL will input the state of CKO into A2. If CKO has not been so programmed, a "1" will be placed in A2. A "0" is always placed in A1 upon the execution of an INIL. The general purpose inputs IN_3–IN_0 are input to A upon execution of an ININ instruction. (See Table II, ININ instruction.) INIL is useful in recognizing pulses of short duration or pulses which occur too often to be read conveniently by an ININ instruction. IL latches are *not cleared* on reset.

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 10-bit word PC_9, PC_8, A, M. LQID can be used for table lookup or code conversion such as BCD to seven-segment. The LQID instruction "pushes" the stack (PC + 1 → SA → SB → SC) and replaces the least significant 8 bits of PC as follows: A → $PC_{7:4}$, RAM(B) → $PC_{3:0}$, leaving PC_9 and PC_8 unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SC → SB → SA → PC), restoring the saved value of PC to continue sequential program execution. Since LQID pushes SB → SC, the previous contents of SC are lost. Also, when LQID pops the stack, the previously pushed contents of SB are left in SC. The net result is that the contents of SB are placed in SC (SB → SC). Note the LQID takes two instruction cycle times to execute.



TL/DD/8825–21

FIGURE 8. INIL Hardware Implementation

SKT INSTRUCTION

The SKT (Skip On Timer) instruction tests the state of an internal 10-bit time-base counter. This counter divides the instruction cycle clock frequency by 1024 and provides a latched indication of counter overflow. The SKT instruction tests this latch, executing the next program instruction if the latch is not set. If the latch has been set since the previous test, the next program instruction is skipped and the latch is reset. The features associated with this instruction, therefore, allow the COP420L/421L to generate its own time-base for real-time processing rather than relying on an external input signal.

For example, using a 2.097 MHz crystal as the time-base to the clock generator, the instruction cycle clock frequency will be 65 kHz (crystal frequency ÷ 32) and the binary counter output pulse frequency will be 64 Hz. For time-of-day or similar real-time processing, the SKT instruction can call a routine which increments a "seconds" counter every 64 ticks.

INSTRUCTION SET NOTES

a. The first word of a COP420L/421L program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, one instruction cycle time is devoted to skipping each byte of the skipped instruction. Thus all program paths except JID and LQID take the same number of cycle times whether instructions are skipped or executed. JID and LQID instructions take 2 cycles if executed and 1 cycle if skipped.

c. The ROM is organized into 16 pages of 64 words each. The Program Counter is a 10-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID or LQID instruction is located in the last word of a page, the instruction operates as if it were in the next page. For example: a JP located in the last word of a page will jump to a location in the next page. Also, a LQID or JID located in the last word of page 3, 7, 11, or 15 will access data in the next group of four pages.

Option List

The COP420L/421L mask-programmable options are assigned numbers which correspond with the COP*420L* pins.

The following is a list of COP420L options. When specifying a COP421L chip, Options 9, 10, 19, and 20 must all be set to zero. When specifying a COP422L chip, options 9, 10, 19, and 20 must all be set to zero; options 21 and 22 may not be set to one, three or five; and option 2 may not be set to one. The options are programmed at the same time as the ROM pattern to provide the user with the hardware flexibility to interface to various I/O components using little or no external circuitry.

The Option Table should be copied and sent in with your EPROM or disc.

Option 1 = 0: Ground Pin—no options available

Option 2: CKO Output
- = 0: clock generator output to crystal/resonator (0 not allowable value if Option 3 = 3)
- = 1: pin is RAM power supply (V_R) input (not available on the COP422L)
- = 2: general purpose input with load device to V_{CC}
- = 3: general purpose input, Hi-Z

Option 3: CKI Input
- = 0: oscillator input divided by 32 (2 MHz max.)
- = 1: oscillator input divided by 16 (1 MHz max.)
- = 2: oscillator input divided by 8 (500 kHz max.)
- = 3: single-pin RC controlled oscillator (÷4)
- = 4: Schmitt trigger clock input (÷4)

Option 4: RESET Input
- = 0: load device to V_{CC}
- = 1: Hi-Z Input

Option 5: L_7 Driver
- = 0: Standard output
- = 1: Open-drain output
- = 2: High current LED direct segment drive output
- = 3: High current TRI-STATE push-pull output
- = 4: Low-current LED direct segment drive output
- = 5: Low-current TRI-STATE push-pull output

Option 6: L_6 Driver
same as Option 5

Option 7: L_5 Driver
same as Option 5

Option 8: L_4 Driver
same as Option 5

Option 9: IN_1 Input
- = 0: load device to V_{CC}
- = 1: Hi-Z input

Option 10: IN_2 Input
same as Option 9

Option 11: V_{CC} pin
- = 0: Standard V_{CC}
- = 1: Optional higher voltage V_{CC}

Option 12: L_3 Driver
same as Option 5

Option 13: L_2 Driver
same as Option 5

Option 14: L_1 Driver
same as Option 5

Option 15: L_0 Driver
same as Option 5

Option 16: SI Input
same as Option 9

Option 17: SO Driver
- = 0: standard output
- = 1: open-drain output
- = 2: push-pull output

Option 18: SK Driver
same as Option 17

Option 19: IN_0 Input
same as Option 9

Option 20: IN_3 Input
same as Option 9

Option 21: G_0 I/O Port
- = 0: very-high current standard output
- = 1: very-high current open-drain output
- = 2: high current open-drain output
- = 3: high current open-drain output
- = 4: standard LSTTL output (fanout = 1)
- = 5: open-drain LSTTL output (fanout = 1)

Option 22: G_1 I/O Port
same as Option 21

Option 23: G_2 I/O Port
same as Option 21

Option 24: G_3 I/O Port
same as Option 21

Option 25: D_3 Output
same as Option 21

Option 26: D_2 Output
same as Option 21

Option 27: D_1 Output
same as Option 21

Option 28: D_0 Output
same as Option 21

Option 29: L Input Levels
- = 0: standard TTL input levels ("0" = 0.8V, "1" = 2.0V)
- = 1: higher voltage input levels ("0" = 1.2V, "1" = 3.6V)

Option 30: IN Input Levels
same as Option 29

Option 31: G Input Levels
same as Option 29

Option 32: SI Input Levels
same as Option 29

Option 33: $\overline{\text{RESET}}$ Input
- = 0: Schmitt trigger input
- = 1: standard TTL input levels
- = 2: higher voltage input levels

Option 34: CKO Input Levels
(CKO = input; Option 2 = 2,3)
same as Option 29

Option 35: COP Bonding
- = 0: COP420L (28-pin device)
- = 1: COP421L (24-pin device)
- = 2: 28- and 24-pin versions
- = 3: COP422L (20-pin device)
- = 4: 28- and 20-pin versions
- = 5: 24- and 20-pin versions
- = 5: 28-, 24-, and 20-pin versions

Option 36: Internal Initialization Logic
- = 0: normal operation
- = 1: no internal initialization logic

Option Table

The following EPROM option information is to be sent to National along with the EPROM.

OPTION DATA

OPTION 1 VALUE = _____ IS: GROUND PIN
OPTION 2 VALUE = _____ IS: CKO OUTPUT
OPTION 3 VALUE = _____ IS: CKI INPUT
OPTION 4 VALUE = _____ IS: RESET INPUT
OPTION 5 VALUE = _____ IS: L_7 DRIVER
OPTION 6 VALUE = _____ IS: L_6 DRIVER
OPTION 7 VALUE = _____ IS: L_5 DRIVER
OPTION 8 VALUE = _____ IS: L_4 DRIVER
OPTION 9 VALUE = _____ IS: IN1 INPUT
OPTION 10 VALUE = _____ IS: IN2 INPUT
OPTION 11 VALUE = _____ IS: VCC PIN
OPTION 12 VALUE = _____ IS: L_3 DRIVER
OPTION 13 VALUE = _____ IS: L_2 DRIVER
OPTION 14 VALUE = _____ IS: L_1 DRIVER
OPTION 15 VALUE = _____ IS: L_0 DRIVER
OPTION 16 VALUE = _____ IS: SI INPUT
OPTION 17 VALUE = _____ IS: SO DRIVER
OPTION 18 VALUE = _____ IS: SK DRIVER

OPTION DATA

OPTION 19 VALUE = _____ IS: IN_0 INPUT
OPTION 20 VALUE = _____ IS: IN_3 INPUT
OPTION 21 VALUE = _____ IS: G_0 I/O PORT
OPTION 22 VALUE = _____ IS: G_1 I/O PORT
OPTION 23 VALUE = _____ IS: G_2 I/O PORT
OPTION 24 VALUE = _____ IS: G_3 I/O PORT
OPTION 25 VALUE = _____ IS: D_3 OUTPUT
OPTION 26 VALUE = _____ IS: D_2 OUTPUT
OPTION 27 VALUE = _____ IS: D_1 OUTPUT
OPTION 28 VALUE = _____ IS: D_0 OUTPUT
OPTION 29 VALUE = _____ IS: L INPUT LEVELS
OPTION 30 VALUE = _____ IS: IN INPUT LEVELS
OPTION 31 VALUE = _____ IS: G INPUT LEVELS
OPTION 32 VALUE = _____ IS: SI INPUT LEVELS
OPTION 33 VALUE = _____ IS: $\overline{\text{RESET}}$ INPUT
OPTION 34 VALUE = _____ IS: CKO INPUT LEVELS
OPTION 35 VALUE = _____ IS: COP BONDING
OPTION 36 VALUE = _____ IS: INTERNAL INITIALIZATION LOGIC

TEST MODE (Non-Standard Operation)

The SO output has been configured to provide for standard test procedures for the customer-programmed COP420L. With SO forced to logic "1", two test modes are provided, depending upon the value of SI:

a. RAM and Internal Logic Test Mode (SI = 1)

b. ROM Test Mode (SI = 0)

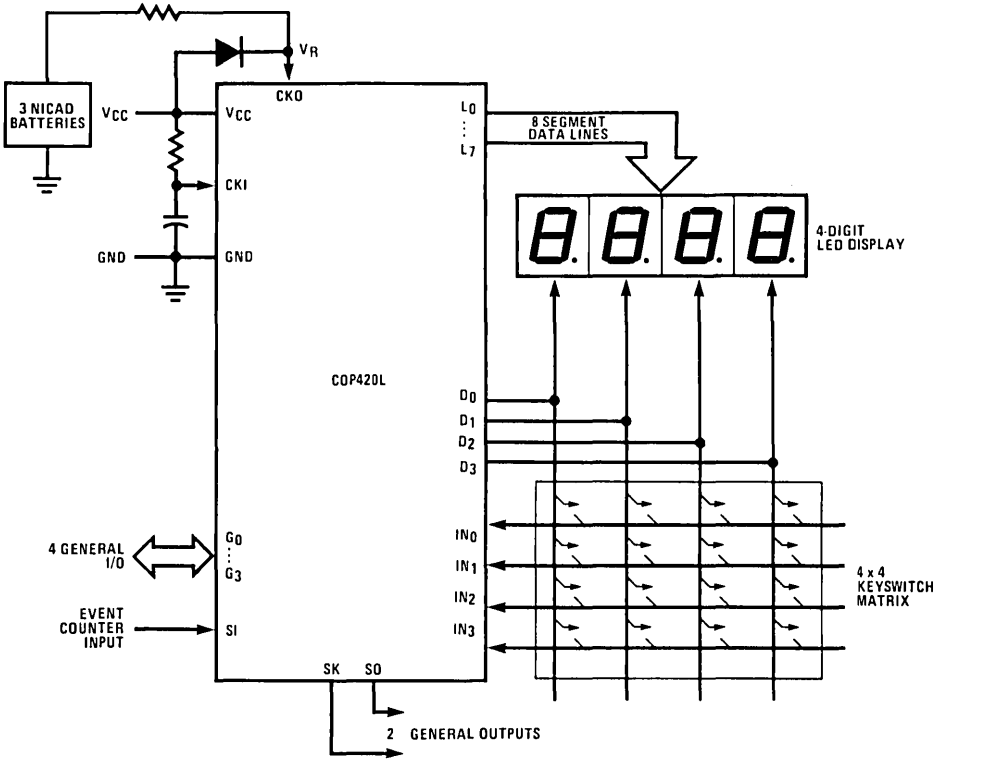
These special test modes should not be employed by the user; they are intended for manufacturing test only.

APPLICATIONS # 1: COP420L General Controller

Figure 9 shows an interconnect diagram for a COP420L used as a general controller. Operation of the system is as follows:

1. The L_7–L_0 outputs are configured as LED Direct Drive outputs, allowing direct connection to the segments of the display.
2. The D_3–D_0 outputs drive the digits of the multiplexed display directly and scan the columns of the 4 x 4 keyboard matrix.
3. The IN_3–IN_0 inputs are used to input the 4 rows of the keyboard matrix. Reading the IN lines in conjunction with the current value of the D outputs allows detection, debouncing, and decoding of any one of the 16 keyswitches.
4. CKI is configured as a single-pin oscillator input allowing system timing to be controlled by a single-pin RC network. CKO is therefore available for use as a V_R RAM power supply pin. RAM data integrity is thereby assured when the main power supply is shut down (see RAM Keep-Alive option description).
5. SI is selected as the input to a binary counter input. With SIO used as a binary counter, SO and SK can be used as general purpose outputs.
6. The 4 bidirectional G I/O ports (G_3–G_0) are available for use as required by the user's application.

Typical Applications



TL/DD/8825-22

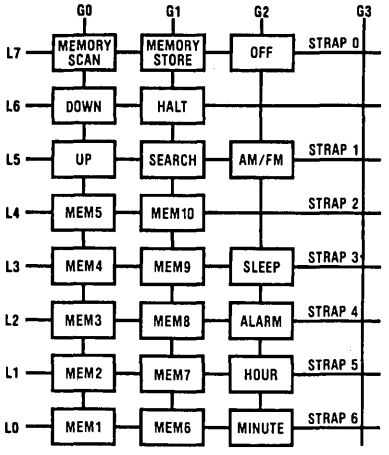
*SO, SI, SK may also be used for Serial I/O

FIGURE 9. COP420L Keyboard/Display Interface

APPLICATION #2:

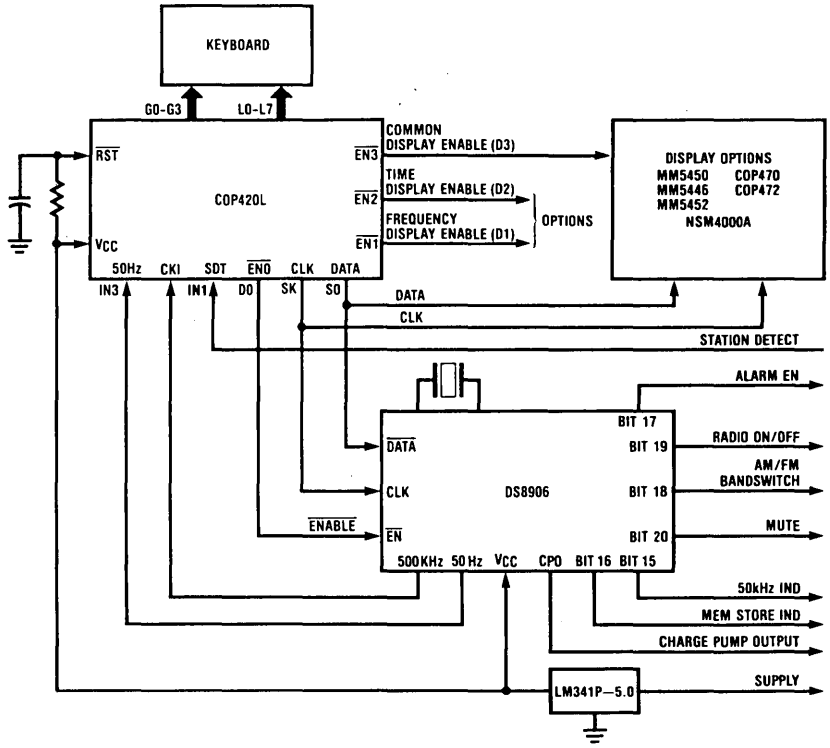
Digitally Tuned Radio Controller and Clock

Keyboard Matrix Configuration



TL/DD/8825-23

Typical Applications (Continued)



TL/DD/8825-24

FIGURE 10. Digital Tuning System Block

Functional Description

LOGIC I/Os

CKI Input: This input accepts an external 500 kHz signal, divides it by eight and outputs the quotient at the CLK output as the system clock.

$\overline{\text{RST}}$ Input: Schmitt trigger input to clear device upon initialization.

SDT Input: Interrupt input for station detection. The SDT signal is generated by the radio's station detector and used by the COP420L to determine if there is a valid station on the active frequency. The status of the SDT input is only relevant during station searching mode. A high on SDT will temporarily terminate the search mode for eight seconds.

ALM Input: A high on ALM will activate alarm output via slave device at alarm time. A low on the input will disable alarm function.

DATA Output: Push-pull output providing serial data to external devices.

CLK Output: Push-pull output providing system clock at data transmitting time.

50 Hz Input: A normally high input to accept a 50 Hz external time base for real-time calculation.

MOMENTARY KEYS DESCRIPTION

MEM 1–MEM 10: Each memory represents data of a favorite station in a certain band. Depression of one of these keys will recall the previous stored data and transmit it to the PLL. The PLL will in turn change the radio's receiving frequency as well as the band if necessary. Memory recall keys can also turn on the radio.

UP: This key will manually increment receiving frequency. The first four steps of increment will be for fine tuning a station, after which will be fast slewing meant for manual receive frequency changing.

DOWN: Has the same function as UP key except that frequency is decremented.

MEMORY SCAN: This will start the radio scanning through all ten memories automatically at eight seconds per memory starting from Memory 1. This will also turn on the radio if it was off.

MEMORY STORE: Enables the memory store mode which lasts for three seconds. Depression of any memory key will store the active frequency and band in that memory and disable the store mode. Any function key will also disable the mode to prevent memory data being accidentally destroyed.

HALT: Depression of the HALT key will stop the search and scan functions at current frequency or memory. HALT also turns on the radio during off time and recall frequency display in signal display mode.

SEARCH: Activates station searching in the current band. Search speed is 50 ms per frequency step with wrapping

Functional Description (Continued)

around at end of band. An 8-second stop will take place on reaching a valid station. The HALT key or any function key will terminate the search. Search direction will normally be upwards unless the DOWN key has been depressed prior to the SEARCH key or during the search function in which case search direction will be downwards.

OFF: Turns off the radio or alarm when active.

AM/FM: Radio band switch.

SLEEP: Activates sleep mode, turns on radio on depression and off radio at the end of sleep period. Setting of sleep period is done by depressing the SLEEP and MINUTE key simultaneously.

ALARM: Enables alarm time setting. Depressing the HOUR or MINUTE key and ALARM key simultaneously will set the alarm hour and minute respectively.

HOUR: Sets the hour digits of time-related functions.

MINUTE: Sets the minute digits of time-related functions.

DIODE STRAPS CONNECTIONS

STRAP 0: Controls the on and off of radio. In applications where a toggle type ON/OFF switch is used, momentary OFF key can be omitted; connecting the strap will turn on the radio and vice versa. Must be connected to use momentary OFF key.

STRAP 1, 2: Selects the AM IF options.

STRAP 3: 12/24-hour clock select.

STRAP 4: 3/5 kHz AM step size select.

STRAP 5, 6: FM IF offsets select.

	STRAP 0	STRAP 3	STRAP 4
Connected	Radio ON	12 hour	5 kHz step
Open	Radio OFF	24 hour	3 kHz step

AM/FM IF OPTIONS

AM	STRAP 1	STRAP 2
455 kHz	X	X
460 kHz	X	✓
450 kHz	✓	X
260 kHz	✓	✓
FM	**STRAP 5**	**STRAP 6**
10.7 MHz	X	X
10.75 MHz	X	✓
10.65 MHz	✓	X
10.8 MHz	✓	✓

X = No connection.

✓ = Diode inserted.

INDIRECT FEATURES AND OPTIONS

As indicated in *Figure 10,* there are a few options and indirect features provided via the help of a slave device, namely the Phase Lock Loop, DS8906N.

DISPLAY OPTIONS

As mentioned above, the COP420L-HSB is MICROWIRE compatible. Internal circuitry enables it to directly interface with all of National's serial input MICROWIRE compatible display drivers whether they are of a direct drive or multiplex drive format. On *Figure 10* is a list of drivers available for the system. EN1 and EN2 are optional enable outputs meant for a dual display system in which EN3 will not be used. By dual display, it means that one display will be constantly showing time information and the other showing frequency information. Whereas in conventional single display systems, the display shows both time and frequency information in a time-sharing method. The National system provides a time-prioritized display-sharing method. That is, whenever a tuning function is completed, the frequency information will stay on the display for eight seconds then time display will take over. This is achieved by using EN3 for the driver's enable logic.

CONTROL OUTPUTS

Six open collector outputs controlled by the COP420L are provided from DS8906N, the phase lock loop for controlling radio switching circuits.

Radio ON/OFF: A high from this output indicates that the radio should be switched on and vice versa.

AM/FM: Output for controlling the AM/FM bandswitch. A high level output indicates FM and a low indicates the AM band.

MUTE: For muting the audio output when performing any frequency related function. The output will go high prior to the frequency change except when doing fine tuning.

ALARM ENABLE: Active high output for turning on the alarm circuit at alarm time.

50 kHz IND: For driving the 50 kHz indicator in FM band or the LSB in a 5-digit display. Output is active high.

MEM STORE IND: For driving the memory store mode indicator. Output is active high.

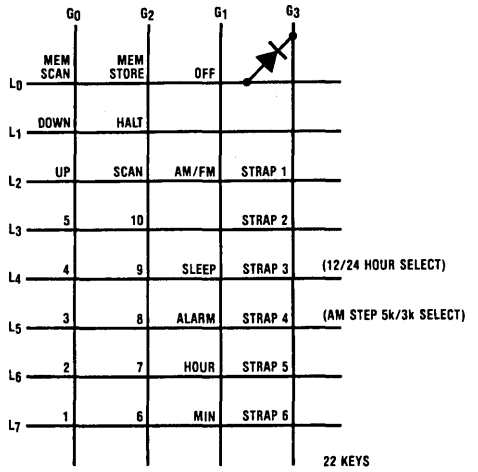
TYPICAL IMPLEMENTATION ALTERNATIVES

A full keyboard or any portion of it can be implemented with various applications for features/functions vs. cost/size.

Figure 11 shows two keyboard configurations with 22-key and 11-key keyboards for a desk-top/tuner system or auto-radio system, respectively.

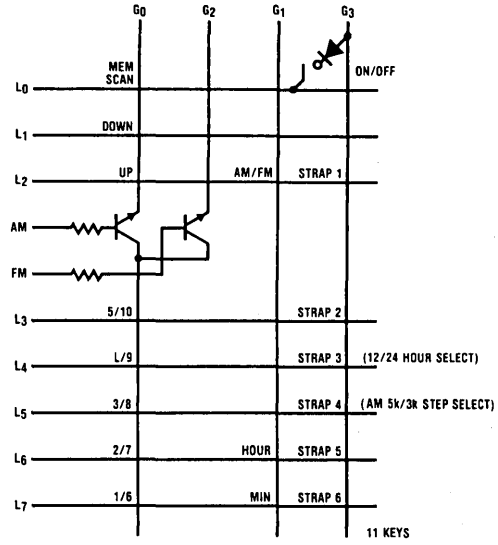
Functional Description (Continued)

Desk Top DTR Keyboard



TL/DD/8825–25

Car DTR Keyboard



TL/DD/8825–26

FIGURE 11

National Semiconductor Corporation

PRELIMINARY

COP220H/COP221H/COP222H/COP320H/ COP321H/COP322H/COP420H/COP421H/COP422H Single-Chip N-Channel Microcontroller

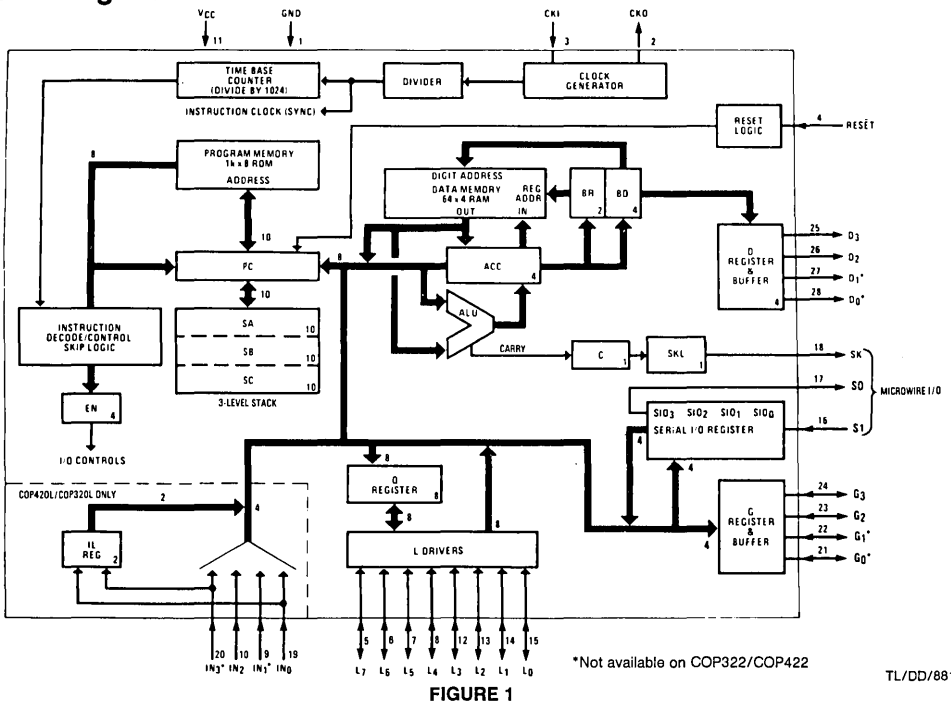
General Description

The COP420H is an enhanced version of the COP420L. It is fabricated in N-channel MOS. It combines the low power consumption of the COP420L and the faster cycle time of the COP420. It has all the software and hardware features of the COP400 series with some added options, and the LED/TRI-STATE® options replaced by an on chip current limiting resistor design. Also it has been designed for late programming of the ROM and hardware options. This allows for a very fast turnaround time for prototype parts. To meet the higher speed requirements the pull-ups and pull-downs had to be made stronger. They have been selected to meet most of the DC specs of the COP420. Therefore some DC parameters will exceed the maximum of the COP420L.

Features

- Low cost
- Low current drain (5 mA max.)
- Powerful instruction set of the COP420L
- 1k x 8 ROM, 64 x 4 RAM
- 4 μs instruction cycle time
- Single supply operation 4.5V–6.3V
- 23 I/O lines (COP420H)
- True vectored interrupt, plus restart
- Three-level subroutine stack
- Internal binary counter register with MICROWIRE™ compatible serial I/O
- Internal time-base counter for real-time processing
- Direct drive of LED digit and segment lines
- General purpose and TRI-STATE outputs
- LSTTL/CMOS compatible in and out
- Software/hardware compatible with COP400 family of devices
- Wider supply range 4.5–9.5V optionally available
- Extended temperature range devices
  COP320H/COP321H/COP322H (−40°C to +85°C)
  COP220H/COP221H/COP222H (−40°C to +110°C)
- Optional internal initialization
- Inputs and open drain outputs able to withstand 15V

Block Diagram



*Not available on COP322/COP422

TL/DD/8815–1

FIGURE 1

COP420H/COP421H/COP422H

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage on Any Pin Relative to GND	−0.5V to +10V
Ambient Operating Temperature	0°C to +70°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C
ESD Susceptibility (Note 4)	2000V

Power Dissipation	
COP420H/COP421H	0.75W at 25°C
	0.4W at 70°C
COP422H	0.65W at 25°C
	0.3W at 70°C
Total Source Current	120 mA
Total Sink Current	120 mA

Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 9.5V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Standard Operating Voltage (V_{CC})	(Note 1)	4.5	6.3	V
Optional Operating Voltage (V_{CC})		4.5	9.5	V
Power Supply Ripple	Peak to Peak		0.5	V
Operating Supply Current	All Inputs and Outputs Open		5	mA
Input Voltage Levels				
CKI Input Levels				
Crystal Input (÷32, ÷16, ÷8)				
Logic High (V_{IH})		2.0		V
Logic Low (V_{IL})		−0.3	0.4	V
Schmitt Trigger Input (÷4)				
Logic High (V_{IH})		0.7 V_{CC}		V
Logic Low (V_{IL})		−0.3	0.6	V
$\overline{\text{RESET}}$ Input Levels	Schmitt Trigger Input			
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.6	V
SO Input Level (Test Mode)	(Note 3)	2.0	2.5	V
All Other Inputs				
Logic High	V_{CC} = Max	3.0		V
Logic High	with TTL Trip Level Options	2.0		V
Logic Low	Selected, V_{CC} = 5V ±5%	−0.3	0.8	V
Logic High	with High Trip Level Options	3.6		V
Logic Low	Selected	−0.3	1.2	V
Input Capacitance			7	pF
Hi-Z Input Leakage	$V_{IN} < 9.5V$	−1	+1	μA
	$9.5 \le V_{IN} \le 15V$	−10	+10	μA
Output Voltage Levels				
LSTTL Operation	V_{CC} = 5V ±10%			
Logic High (V_{OH})	I_{OH} = −25 μA	2.7		V
Logic Low (V_{OL})	I_{OL} = 0.36 ma		0.4	V
CMOS Operation (Note 2)	V_{CC} = 4.5V			
Logic High	I_{OH} = −10 μA	V_{CC} −1		V
Logic Low	I_{OL} = +10 μA		0.2	V

Note 1: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Note 2: TRI-STATE and LED configurations are excluded.

Note 3: SO output "0" level must be less than 0.8V for normal operation.

Note 4: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

COP420H/COP421H/COP422H

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 9.5V$ unless otherwise noted (Continued)

Parameter	Conditions	Min	Max	Units
Output Current Levels				
Output Sink Current				
L_0–L_7 Outputs and Standard	$V_{CC} = 9.5V, V_{OL} = 0.4V$	3.2		mA
G_0–G_3, D_0–D_3 Outputs (I_{OL})	$V_{CC} = 6.3V, V_{OL} = 0.4V$	2.3		mA
SO & SK	$V_{CC} = 4.5V, V_{OL} = 0.4V$	1.7		mA
G_0–G_3 and D_0–D_3 Outputs with	$V_{CC} = 9.5V, V_{OL} = 1.0V$	15		mA
High Current Options (I_{OL})	$V_{CC} = 6.3V, V_{OL} = 1.0V$	11		mA
	$V_{CC} = 4.5V, V_{OL} = 1.0V$	7.5		mA
G_0–G_3 and D_0–D_3 Outputs with	$V_{CC} = 9.5V, V_{OL} = 1.0V$	30		mA
Very High Current Options (I_{OL})	$V_{CC} = 6.3V, V_{OL} = 1.0V$	22		mA
	$V_{CC} = 4.5V, V_{OL} = 1.0V$	15		mA
CKI (Single-Pin RC Oscillator)	$V_{CC} = 4.5V, V_{IH} = 3.5V$	2.2		mA
CKO	$V_{CC} = 4.5V, V_{OL} = 0.4V$	1.8		mA
Output Source Current				
Standard Configuration,	$V_{CC} = 9.5V, V_{OH} = 2.0V$	−470	−2350	μA
All Outputs (I_{OH})	$V_{CC} = 6.3V, V_{OH} = 2.0V$	−250	−1250	μA
	$V_{CC} = 4.5V, V_{OH} = 2.0V$	−100	−500	μA
Push-Pull Configuration	$V_{CC} = 9.5V, V_{OH} = 4.75V$	−1.4		mA
SO and SK Outputs (I_{OH})	$V_{CC} = 6.3V, V_{OH} = 2.4V$	−1.4		mA
	$V_{CC} = 4.5V, V_{OH} = 1.0V$	−1.2		mA
LED Configuration, L_0–L_7				
Outputs, Low Current	$V_{CC} = 9.5V, V_{OH} = 2.0V$	−6	−18	mA
Driver Option (I_{OH})	$V_{CC} = 6.0V, V_{OH} = 2.0V$	−1.5	−7	mA
LED Configuration, L_0–L_7				
Outputs, High Current	$V_{CC} = 9.5V, V_{OH} = 2.0V$	−12	−36	mA
Driver Option (I_{OH})	$V_{CC} = 6.0V, V_{OH} = 2.0V$	−3.0	−14	mA
TRI-STATE Configuration,	$V_{CC} = 9.5V, V_{OH} = 5.5V$	−0.75		mA
L_0–L_7 Outputs, Low	$V_{CC} = 6.3V, V_{OH} = 3.2V$	−0.8		mA
Current Driver Option (I_{OH})	$V_{CC} = 4.5V, V_{OH} = 1.5V$	−0.9		mA
TRI-STATE Configuration,	$V_{CC} = 9.5V, V_{OH} = 5.5V$	−1.5		mA
L_0–L_7 Outputs, High	$V_{CC} = 6.3V, V_{OH} = 3.2V$	−1.6		mA
Current Driver Option (I_{OH})	$V_{CC} = 4.5V, V_{OH} = 1.5V$	−100	−800	mA
Input Load Source Current	$V_{CC} = 5.0V, V_{IL} = 0V$	−10	−140	μA
CKO Output				
RAM Power Supply Option	$V_R = 3.3V$		3.0	mA
Power Requirement				
TRI-STATE Output Leakage Current		−2.5	+2.5	μA
Open Drain Output Leakage	$V_{OUT} < 9.5V$	−2.5	+2.5	μA
	$9.5V \le V_{OUT} < 15V$	−2.5	+2.5	μA
Total Sink Current Allowed				
All Outputs Combined			120	mA
D, G Ports			120	mA
L_7–L_4			4	mA
L_3–L_0			4	mA
All Other Pins			1.5	mA
Total Source Current Allowed				
All I/O Combined			120	mA
L_7–L_4			60	mA
L_3–L_0			60	mA
Each L Pin			30	mA
All Other Pins			1.5	mA

COP320H/COP321H/COP322H

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage on Any Pin Relative to GND	−0.5V to +10V
Ambient Operating Temperature	−40°C to +85°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C
ESD Susceptibility (Note 4)	3000V

Power Dissipation	
COP320H/COP321H	0.75W at 25°C
COP220H/COP221H	0.4W at 70°C
COP322H/COP222H	0.65W at 25°C
	0.20W at 70°C
Total Source Current	120 mA
Total Sink Current	120 mA

Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.

DC Electrical Characteristics

COP220H/221H/222H $-40°C \le T_A \le +110°C$, $4.5V \le V_{CC} \le 7.5V$; COP320H/321H/322H $-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 7.5V$ unless otherwise noted.

Parameter	Conditions	Min	Max	Units
Standard Operating Voltage (V_{CC})	(Note 1)	4.5	5.5	V
Optional Operating Voltage (V_{CC})		4.5	7.5	V
Power Supply Ripple	Peak to Peak		0.5	V
Operating Supply Current	All Inputs and Outputs Open		8	mA
Input Voltage Levels				
CKI Input Levels				
Crystal Input				
Logic High (V_{IH}) V_{CC} = Max		3.0		V
Logic High (V_{IH})				
V_{CC} = 5V ±5%		2.2		V
Logic Low (V_{IL})		−0.3	0.3	V
Schmitt Trigger Input				
Logic High (V_{IH})		0.7 V_{CC}		V
Logic Low (V_{IL})		−0.3	0.4	V
$\overline{\text{RESET}}$ Input Levels	Schmitt Trigger Input			
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.4	V
SO Input Level (Test Mode)	(Note 3)	2.2	2.5	V
All Other Inputs				
Logic High	V_{CC} = Max	3.0		V
Logic High	with TTL Trip Level Options	2.2		V
Logic Low	Selected, V_{CC} = 5V ±5%	−0.3	0.6	V
Logic High	with High Trip Level Options	3.6		V
Logic Low	Selected	−0.3	1.2	V
Input Capacitance			7	pF
Hi-Z Input Leakage	$V_{IN} < 7.5$	−2	+2	μA
	$7.5 \le V_{IN} \le 15V$	−20	+20	μA
Output Voltage Levels				
LSTTL Operation	V_{CC} = 5V ±10%			
Logic High (V_{OH})	I_{OH} = −20 μA	2.7		V
Logic Low (V_{OL})	I_{OL} = 0.36 mA		0.4	V
CMOS Operation (Note 2)	V_{CC} = 4.5V			
Logic High	I_{OH} = −10 μA	V_{CC} −1		V
Logic Low	I_{OL} = +10 μA		0.2	V

Note 1: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Note 2: TRI-STATE and LED configurations are excluded.

Note 3: SO output "0" level must be less than 0.6V for normal operation.

Note 4: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

COP320H/COP321H/COP322H

DC Electrical Characteristics

$-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 7.5V$ unless otherwise noted (Continued)

Parameter	Conditions	Min	Max	Units
Output Current Levels				
Output Sink Current				
L_0–L_7 Outputs and Standard	$V_{CC} = 7.5V$, $V_{OL} = 0.4V$	2.5		mA
G_0–G_3 and D_0–D_3 Outputs (I_{OL})	$V_{CC} = 5.5V$, $V_{OL} = 0.4V$	1.9		mA
	$V_{CC} = 4.5V$, $V_{OL} = 0.4V$	1.6		mA
G_0–G_3 and D_0–D_3 Outputs with	$V_{CC} = 7.5V$, $V_{OL} = 1.0V$	12		mA
High Current Options (I_{OL})	$V_{CC} = 5.5V$, $V_{OL} = 1.0V$	9		mA
	$V_{CC} = 4.5V$, $V_{OL} = 1.0V$	7		mA
G_0–G_3 and D_0–D_3 Outputs with	$V_{CC} = 7.5V$, $V_{OL} = 1.0V$	24		mA
Very High Current Options (I_{OL})	$V_{CC} = 5.5V$, $V_{OL} = 1.0V$	18		mA
	$V_{CC} = 4.5V$, $V_{OL} = 1.0V$	14		mA
CKI (Single-Pin RC Oscillator)	$V_{CC} = 4.5V$, $V_{IH} = 3.5V$	2		mA
CKO	$V_{CC} = 4.5V$, $V_{OL} = 0.4V$	1.6		mA
Output Source Current				
Standard Configuration,	$V_{CC} = 7.5V$, $V_{OH} = 2.0V$	−330	−900	μA
All Outputs (I_{OH})	$V_{CC} = 5.5V$, $V_{OH} = 2.0V$	−110	−600	μA
	$V_{CC} = 4.5V$, $V_{OH} = 2.0V$	−92	−350	μA
Push-Pull Configuration	$V_{CC} = 7.5V$, $V_{OH} = 3.75V$	−0.85		mA
SO and SK Outputs (I_{OH})	$V_{CC} = 5.5V$, $V_{OH} = 2.0V$	−1.1		mA
	$V_{CC} = 4.5V$, $V_{OH} = 1.0V$	−1.2		mA
LED Configuration, L_0–L_7	$V_{CC} = 7.5V$, $V_{OH} = 2.0V$	−2.0	−130	mA
Outputs, Low Current	$V_{CC} = 6.0V$, $V_{OH} = 2.0V$	−1.5	−8.0	mA
Driver Option (I_{OH})				
LED Configuration, L_0–L_7	$V_{CC} = 7.5V$, $V_{OH} = 2.0V$	−2.7	−30	mA
Outputs, High Current	$V_{CC} = 6.0V$, $V_{OH} = 2.0V$	−2.7	−18	mA
Driver Option (I_{OH})				
TRI-STATE Configuration,	$V_{CC} = 7.5V$, $V_{OH} = 4.0V$	−0.7		mA
L_0–L_7 Outputs, Low	$V_{CC} = 5.5V$, $V_{OH} = 2.7V$	−0.6		mA
Current Driver Option (I_{OH})	$V_{CC} = 4.5V$, $V_{OH} = 1.5V$	−0.9		mA
TRI-STATE Configuration,	$V_{CC} = 7.5V$, $V_{OH} = 4.0V$	−1.4		mA
L_0–L_7 Outputs, High	$V_{CC} = 5.5V$, $V_{OH} = 2.7V$	−1.2		mA
Current Driver Option (I_{OH})	$V_{CC} = 4.5V$, $V_{OH} = 1.5V$	−1.8		mA
Input Load Source Current	$V_{CC} = 5.0V$, $V_{IL} = 0V$	−100	−1150	μA
CKO Output				
RAM Power Supply Option	$V_R = 3.3V$		4.0	mA
Power Requirement				
TRI-STATE Output Leakage Current		−5	+5	μA
Open Drain Output Leakage	$V_{OUT} < 7.5V$	−5	+5	μA
	$7.5V \le V_{OUT} \le 15V$	−50	+50	μA
Total Sink Current Allowed				
All Outputs Combined			120	mA
D, G Ports			120	mA
L_7–L_4			4	mA
L_3–L_0			4	mA
All Other Pins			1.5	mA
Total Source Current Allowed				
All I/O Combined			120	mA
L_7–L_4			60	mA
L_3–L_0			60	mA
Each L Pin			30	mA
All Other Pins			1.5	mA

AC Electrical Characteristics

COP420H/COP421H/COP422H: $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 9.5V$ unless otherwise noted
COP320H/COP321H/COP322H: $-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 7.5V$ unless otherwise noted
COP220H/COP221H/COP222H: $-40°C \le T_A \le +110°C$, $4.5V \le V_{CC} \le 7.5V$

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time—t_C	(Note 4)	4.0	40	μs
CKI				
Input Frequency—f_I	÷32 Mode (Note 5)	0.8	4.0	MHz
	÷16 Mode	0.4	4.0	MHz
	÷8 Mode	0.2	2.0	MHz
	÷4 Mode	0.1	1.0	MHz
Duty Cycle		40	60	%
Rise Time	f_I = 4 MHz		60	ns
Fall Time			40	ns
CKI Using RC (÷4)	R = 15 kΩ ±5% C = 100 pF ±10%			
Instruction Cycle Time (Note 6)		16	28	μs
INPUTS:				
IN_3–IN_0, G_3–G_0, L_7–L_0				
t_{SETUP}		1.7		μs
t_{HOLD}		0.3		μs
SI				
t_{SETUP}		0.3		μs
t_{HOLD}		0.25		μs
OUTPUT PROPAGATION DELAY	Test Condition: C_L = 50 pF, R_L = 20 kΩ, V_{OUT} = 1.5V			
SO, SK Outputs				
t_{pd1}, t_{pd0}			0.25	μs
All Other Outputs				
t_{pd1}, t_{pd0}			1.4	μs

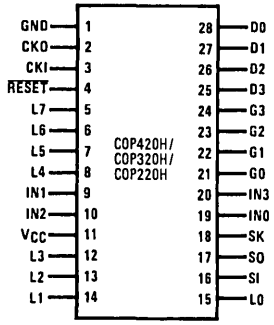
Note 4: COP420H Instruction cycle time is 4 μs ≤ T_{cycle} ≤ 40 μs.

Note 5: T_{cycle} = 4 μs cannot be achieved using CKI % 32.

Note 6: RC tolerances and the variation due to the device included.

Connection Diagrams

Dual-In Line Package

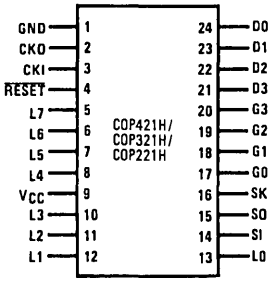


TL/DD/8815-2

Top View

Order Number COP420H-XXX/N, COP320H-XXX/N or COP220H-XXX/N
See NS Package Number N28B

Dual-In Line Package

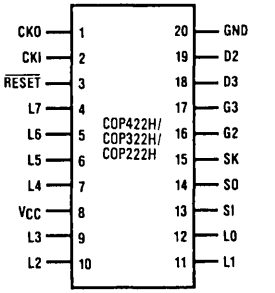


TL/DD/8815-3

Top View

Order Number COP421H-XXX/M or COP421H-XXX/N COP321H-XXX/M or COP321H-XXX/N, or COP221H-XXX/M or COP221H-XXX/N
See NS Package Number M24B or N24A

Dual-In Line Package



TL/DD/8815-4

Top View

Order Number COP422H-XXX/M or COP422H-XXX/N COP322H-XXX/M or COP322H-XXX/N, or COP222H-XXX/M or COP222H-XXX/N
See NS Package Number M20B or N20A

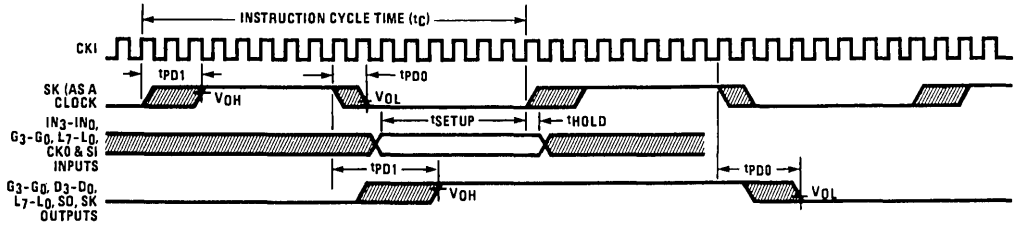
FIGURE 2

Pin Descriptions

Pin	Description
L_7–L_0	8 bidirectional I/O ports with TRI-STATE
G_3–G_0	4 bidirectional I/O ports
D_3–D_0	4 general purpose outputs
IN_3–IN_0	4 general purpose inputs (COP420L only)
SI	Serial input (or counter input)
SO	Serial output (or general purpose output)

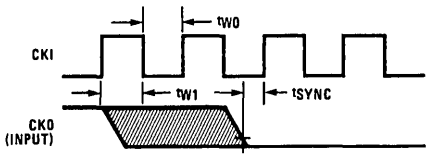
Pin	Description
SK	Logic-controlled clock (or general purpose output)
CKI	System oscillator input
CKO	System oscillator output (or general purpose input, RAM power supply)
$\overline{RESET}$	System reset input
V_{CC}	Power supply
GND	Ground

Timing Diagrams



TL/DD/8815-5

FIGURE 3. Input/Output Timing Diagrams (Crystal Divide-by-16 Mode)



TL/DD/8815-6

FIGURE 3a. Synchronization Timing

Functional Description

For ease of reading this description, only COP420H and/or COP421H are referenced; however, all such references apply also to COP220H, COP221H, COP222H, COP320H, COP321H, COP322H, or COP422H.

A block diagram of the COP420H is given in *Figure 1.* Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1" (greater than 2V). When a bit is reset, it is a logic "0" (less than 0.8V).

PROGRAM MEMORY

Program Memory consists of a 1,024 byte ROM. As can be seen by an examination of the COP420H/421H instruction set, these words may be program instructions, program data or ROM addressing data. Because of the special characteristics associated with the JP, JSRP, JID and LQID instructions, ROM must often be thought of as being organized into 16 pages of 64 words each.

ROM addressing is accomplished by a 10-bit PC register. Its binary value selects one of the 1,024 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 10-bit binary count value. Three levels of subroutine nesting are implemented by the 10-bit subroutine save registers, SA, SB and SC, providing a last-in, first-out (LIFO) hardware subroutine stack.

ROM instruction words are fetched, decoded and executed by the Instruction Decode, Control and Skip Logic circuitry.

DATA MEMORY

Data memory consists of a 256-bit RAM, organized as 4 data registers of 16 4-bit digits. RAM addressing is implemented by a 6-bit B register whose upper 2 bits (Br) select 1 of 4 data registers and lower 4 bits (Bd) select 1 of 16 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) is usually loaded into or from, or exchanged with, the A register (accumulator), it may also be loaded into or from the Q latches or loaded from the L ports. RAM addressing may also be performed directly by the LDD and XAD instructions is based upon the 6-bit contents of the operand field of these instructions. The Bd register also serves as a source register for 4-bit data sent directly to the D outputs.

INTERNAL LOGIC

The 4-bit A register (accumulator) is the source and destination register for most I/O, arithmetic, logic and data memory access operations. It can also be used to load the Br and Bd portions of the B register, to load and input 4 bits of the 8-bit Q latch data, to input 4 bits of the 8-bit L I/O port data and to perform data exchanges with the SIO register.

A 4-bit adder performs the arithmetic and logic functions of the COP420H/421H, storing its results in A. It also outputs a carry bit to the 1-bit C register, most often employed to indicate arithmetic overflow. The C register, in conjunctions with the XAS instruction and the EN register, also serves to control the SK output. C can be outputted directly to SK or can enable SK to be a sync clock each instruction cycle time. (See XAS instruction and EN register description, below.)

Four general-purpose inputs, IN_3–IN_0, are provided.

The D register provides 4 general-purpose outputs and is used as the destination register for the 4-bit contents of Bd. The D outputs can be directly connected to the digits of a multiplexed LED display.

The G register contents are outputs to 4 general-purpose bidirectional I/O ports. G I/O ports can be directly connected to the digits of a multiplexed LED display.

The Q register is an internal, latched, 8-bit register, used to hold data loaded to or from M and A, as well as 8-bit data from ROM. Its contents are outputted to the L I/O ports when the L drivers are enabled under program control. (See LEI instruction.)

The 8 L drivers, when enabled, output the contents of latched Q data to the L I/O ports. Also, the contents of L may be read directly into A and M. L I/O ports can be directly connected to the segments of a multiplexed LED display (using the LED Direct Drive output configuration option) with Q data being outputted to the Sa–Sg and decimal point segments of the display.

The SIO register functions as a 4-bit serial-in/serial-out shift register or as a binary counter depending on the contents of the EN register. (See EN register description, below.) Its contents can be exchanged with A, allowing it to input or output a continuous serial data stream. SIO may also be used to provide additional parallel I/O by connecting SO to external serial-in/parallel-out shift registers.

The XAS instruction copies C into the SKL latch. In the counter mode, SK is the output of SKL; in the shift register mode, SK outputs SKL ANDed with the clock.

The EN register is an internal 4-bit register loaded under program control by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register (EN_3–EN_0).

1. The least significant bit of the enable register, EN_0, selects the SIO register as either a 4-bit shift register or a 4-bit binary counter. With EN_0 set, SIO is an asynchronous binary counter, *decrementing* its value by one upon each low-going pulse ("1" to "0") occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output is equal to the value of EN_3. With EN_0 reset, SIO is a serial shift register shifting left each instruction cycle time. The data present at SI goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. (See 4 below.) The SK output becomes a logic-controlled clock.
2. With EN_1 set the IN_1 input is enabled as an interrupt input. Immediately following an interrupt, EN_1 is reset to disable further interrupts.
3. With EN_2 set, the L drivers are enabled to output the data in Q to the L I/O ports. Resetting EN_2 disables

Functional Description (Continued)

the L drivers, placing the L I/O ports in a high-impedance input state.

4. EN_3, in conjunction with EN_0, affects the SO output. With EN_0 set (binary counter option selected) SO will output the value loaded into EN_3. With EN_0 reset (serial shift register option selected), setting EN_3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN_3 with the serial shift register option selected disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains reset to "0". The table below provides a summary of the modes associated with EN_3 and EN_0.

Enable Register Modes—Bits EN_3 and EN_0

EN_3	EN_0	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL = 1, SK = Clock If SKL = 0, SK = 0
1	0	Shift Register	Input to Shift Register	Serial Out	If SKL = 1, SK = Clock If SKL = 0, SK = 0
0	1	Binary Counter	Input to Binary Counter	0	If SKL = 1, SK = 1 If SKL = 0, SK = 0
1	1	Binary Counter	Input to Binary Counter	1	If SKL = 1, SK = 1 If SKL = 0, SK = 0

INTERRUPT

The following features are associated with the IN_1 interrupt procedure and protocol and must be considered by the programmer when utilizing interrupts.

a. The interrupt, once aknowledged as explained below, pushes the next sequential program counter address (PC + 1) onto the stack, pushing in turn the contents of the other subroutine-save registers to the next lower level (PC + 1 → SA → SB → SC). Any previous contents of SC are lost. The program counter is set to hex address 0FF (the last word of page 3) and EN_1 is reset.

b. An interrupt will be acknowledged only after the following conditions are met:
   1. EN_1 has been set.
   2. A low-going pulse ("1" to "0") at least two instruction cycles wide occurs on the IN_1 input.
   3. A currently executing instruction has been completed.
   4. All successive transfer of control instructions and successive LBIs have been completed (e.g., if the main program is executing a JP instruction which transfers program control to another JP instruction, the interrupt will not be acknowledged until the second JP instruction has been executed).

c. Upon acknowledgement of an interrupt, the skip logic status is saved and later restored upon popping of the stack. For example if an interrupt occurs during the execution of ASC (Add with Carry, Skip on Carry) instruction which results in carry, the skip logic status is saved and program control is transferred to the interrupt servicing routine at address 0FF. At the *end* of the interrupt routine, a RET instruction is executed to "pop" the stack and return program control to the instruction following the original ASC. *At this time,* the skip logic is enabled and skips this instruction because of the previous ASC carry. Subroutines and LQID instructions should not be nested within the interrupt servicing routine since their popping the stack will enable any previously saved main program skips, interfering with the orderly execution of the interrupt routine.

d. The first instruction of the interrupt routine at hex address 0FF must be a NOP.

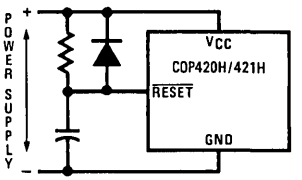
e. A LEI instruction can be put immediately before the RET to re-enable interrupts.

INITIALIZATION

The Reset Logic will initialize (clear) the device upon power-up if the power supply rise time is less than 1 ms and greater than 1 μs. If the power supply rise time is greater than 1 ms, the user must provide an external RC network and diode to the $\overline{RESET}$ pin as shown below. The $\overline{RESET}$ pin is configured as a Schmitt trigger input. If not used it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the $\overline{RESET}$ input, provided it stays low for at least three instruction cycle times.

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN, and G registers are cleared. The SK output is enabled as a SYNC output, providing a pulse each instruction cycle time. *Data Memory (RAM) is not cleared upon initialization.* The first instruction at address 0 must be a CLRA.

Power-Up Clear Circuit



TL/DD/8815–7

RC ≥ 5 × Power Supply Rise Time

Functional Description (Continued)

OSCILLATOR

There are three basic clock oscillator configurations available as shown by *Figure 4.*

a. **Crystal Controlled Oscillator.** CKI and CKO are connected to an external crystal. The instruction cycle time equals the crystal frequency divided by 32 (optional by 16 or 8).

b. **External Oscillator.** CKI is an external clock input signal. The external frequency is divided by 32 (optional by 16 or 8) to give the instruction cycle time. CKO is now available to be used as the RAM power supply (V_R) or as a general purpose input.

c. **RC Controlled Oscillator.** CKI is configured as a single pin RC controlled Schmitt trigger oscillator. The instruction cycle equals the oscillation frequency divided by 4. CKO is available as the RAM power supply (V_R) or as a general purpose input.

CKO PIN OPTIONS

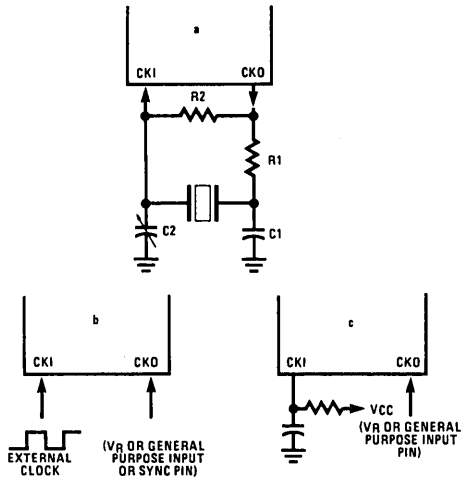
In a crystal controlled oscillator system, CKO is used as an output to the crystal network or it can be a general purpose input read into bit 2 of A (accumulator) upon execution of an INIL Instruction. As another option, CKO can be a RAM power supply pin (V_R), allowing its connection to a standby/backup power supply to maintain the integrity of RAM data with minimum power drain when the main supply is inoperative or shut down to conserve power. Using either option is appropriate in applications where the COP420H/421H system timing configuration does not require use of the CKO pin.

RAM KEEP-ALIVE OPTION (Not available on COP422H)

Selecting CKO as the RAM power supply (V_R) allows the user to shut off the chip power supply (V_{CC}) and maintain data in the RAM. To insure that RAM data integrity is maintained, the following conditions must be met:

1. $\overline{\text{RESET}}$ must go low before V_{CC} goes below spec during power-off; V_{CC} must be within spec before $\overline{\text{RESET}}$ goes high on power-up.
2. During normal operation V_R must be within the operating range of the chip, with $(V_{CC} - 1) \leq V_R \leq V_{CC}$.
3. V_R must be $\leq$ 3.3V with V_{CC} off.

Crystal Oscillator



TL/DD/8815–8

Crystal Oscillator

Crystal Value	Component Values		
	R1 (Ω)	R2 (Ω)	C (pF)
455 kHz	4.7k	1M	220
2.097 MHz	1k	1M	56
3.58 MHz	1k	1M	27
4 MHz	1k	1M	27

RC Controlled Oscillator

R (kΩ)	C (pF)	Instruction Cycle Time (μs)
51	100	19 ± 15%
82	56	19 ± 13%
12	100	5 ± 20%
6.8	220	5.3 ± 23%
8.2	300	8 ± 29%
22	100	8.6 ± 16%

Note: 200k ≥ R ≥ 5k
360 pF ≥ C ≤ 50 pF

FIGURE 4. COP420H/421H Oscillator

Functional Description (Continued)

I/O OPTIONS

COP420H/421H outputs have the following optional configurations, illustrated in *Figure 5*:

a. **Standard**—an enhancement mode device to ground in conjunction with a depletion-mode device to V_{CC}, compatible with LSTTL and CMOS input requirements. Available on SO, SK, and all D and G outputs.

b. **Open-Drain**—an enhancement-mode device to ground only, allowing external pull-up as required by the user's application. Available on SO, SK, and all D and G outputs.

c. **Push-Pull**—An enhancement-mode device to ground in conjunction with a depletion-mode device paralleled by an enhancement-mode device to V_{CC}. This configuration has been provided to allow for fast rise and fall times when driving capacitive loads. Available on SO and SK outputs only.

d. **Standard L**—same as **a.**, but may be disabled. Available on L outputs only.

e. **Open Drain L**—same as **b.**, but may be disabled. Available on L outputs only.

f. **LED Direct Drive**—an enhancement-mode device to ground and to V_{CC}, meeting the typical current sourcing requirements of the segments of an LED display. The sourcing device is clamped to limit current flow. These devices may be turned off under program control (see Functional Description, EN Register), placing the outputs in a high-impedance state to provide required LED segment blanking for a multiplexed display. Available on L outputs only.

g. **TRI-STATE Push-Pull**—an enhancement-mode device to ground and V_{CC}. These outputs are TRI-STATE outputs, allowing for connection of these outputs to a data bus shared by other bus drivers. Available on L outputs only.

COP420H/COP421H inputs have the following optional configurations:

h. An on-chip depletion load device to V_{CC}.

i. A Hi-Z input which must be driven to a "1" or "0" by external components.

The above input and output configurations share common enhancement-mode and depletion-mode devices. Specifically, all configurations use one or more of six devices (numbered 1–6, respectively). Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figure 6* for each of these devices to allow the designer to effectively use these I/O configurations in designing a COP420H/421H system.

The SO, SK outputs can be configured as shown in **a.**, **b.**, or **c.** The D and G outputs can be configured as shown in **a.** or **b.** Note that when inputting data to the G ports, the G outputs should be set to "1". The L outputs can be configured as in **d.**, **e.**, **f.** or **g.**

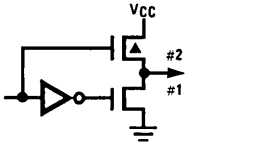
An important point to remember if using configuration **d.** or **f.** with the L drivers is that even when the L drivers are disabled, the depletion load device will source a small amount of current (see *Figure 6,* device 2); however, when the L lines are used as inputs, the disabled depletion device *cannot* be relied on to source sufficient current to pull an input to a logic 1.

COP421H

If the COP420H is bonded as a 24-pin device, it becomes the COP421H, illustrated in *Figure 2,* COP420H/421H Connection Diagrams. Note that the COP421H does not contain the four general purpose IN inputs (IN_3–IN_0). Use of this option precludes, of course, use of the IN options and the interrupt feature. All other options are available for the COP421H.

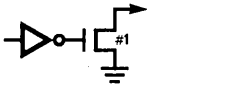
COP422H

If the COP421H is bonded as a 20-pin device, it becomes the COP422H, as illustrated in *Figure 2.* Note that the COP422H contains all the COP421H pins except D_0, D_1, G_0, and G_1. COP422H also does not allow RAM power supply input as a valid CKO pin option.



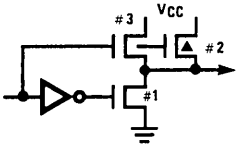
TL/DD/8815–9

a. Standard Output



TL/DD/8815–10

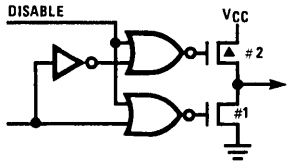
b. Open-Drain Output



TL/DD/8815–11

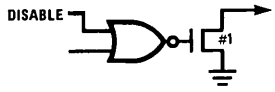
c. Push-Pull Output

Functional Description (Continued)



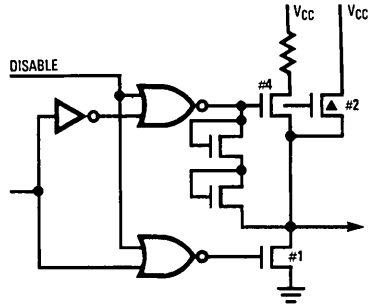
TL/DD/8815-12

d. Standard L Output



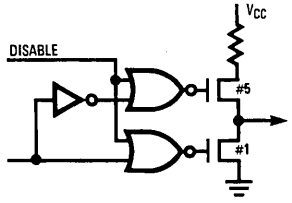
TL/DD/8815-13

e. Open-Drain L Output



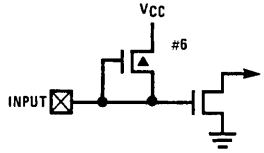
(▲ is Depletion Device) TL/DD/8815-14

f. LED (L Output)



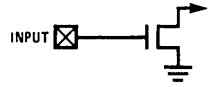
TL/DD/8815-15

g. TRI-STATE Push-Pull (L Output)



TL/DD/8815-16

h. Input with Load



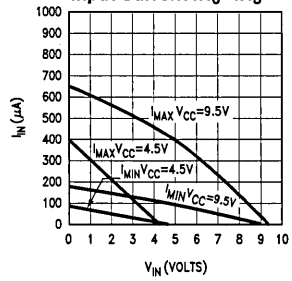
TL/DD/8815-17

i. HI-Z Input

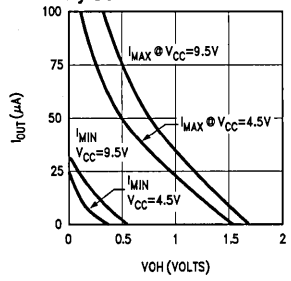
FIGURE 5. Output Configurations

Typical Performance Characteristics

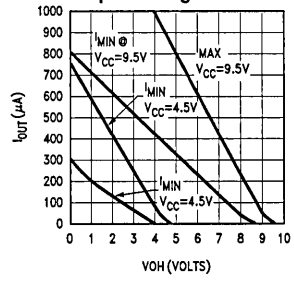
Input Current IN_0–IN_3



Input Current for L_0–L_7 when Output Programmed OFF by Software

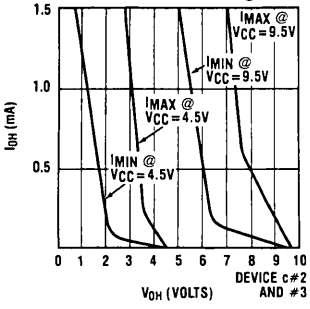


Source Current for Standard Output Configuration



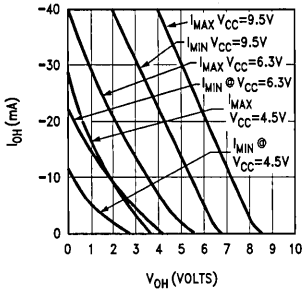
TL/DD/8815-18

Source Current for SO and SK in Push-Pull Configuration

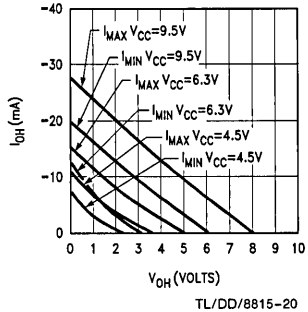


TL/DD/8815-19

Source Current for L_0–L_7 in TRI-STATE Configuration (High Current Option)



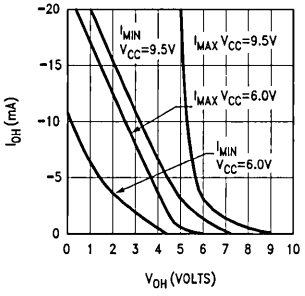
Source Current for L_0–L_7 in TRI-STATE Configuration (Low Current Option)



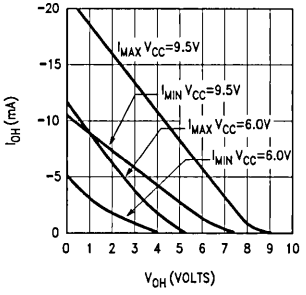
TL/DD/8815-20

Typical Performance Characteristics (Continued)

LED Output Source Current (High Current LED Option)

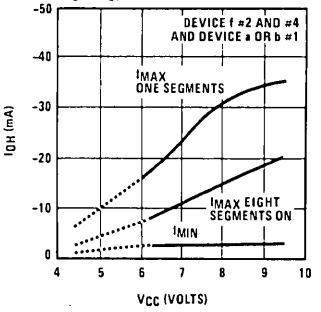


LED Output Source Current (Low Current LED Option)



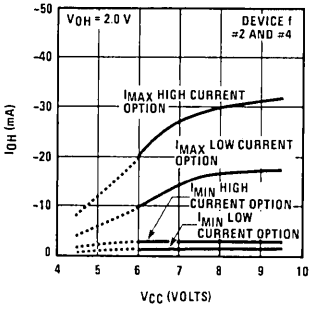
TL/DD/8815-21

LED Output Direct Segment and Digit Drive (High Current Options on L_0–L_7; Very High Current Options on D_0–D_3 or G_0–G_3)



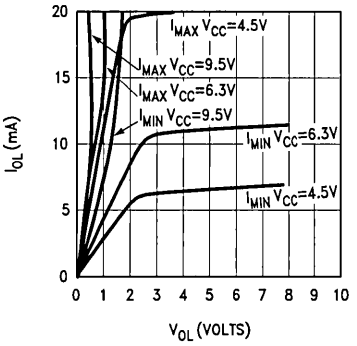
TL/DD/8815-22

LED Output Direct Segment Drive



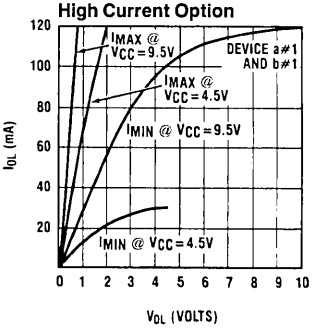
TL/DD/8815-23

Output Sink Current for SO and SK

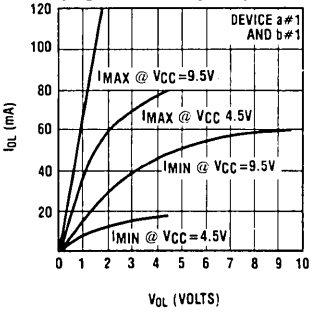


TL/DD/8815-24

Output Sink Current for G_0–G_3 and D_0–D_3 with Very High Current Option



Output Sink Current for G_0–G_3 and D_0–D_3 (High Current Option)



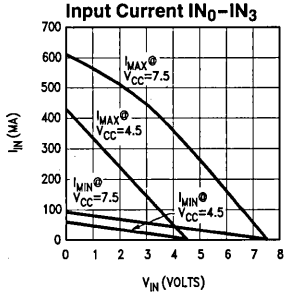
TL/DD/8815-25

TL/DD/8815-26

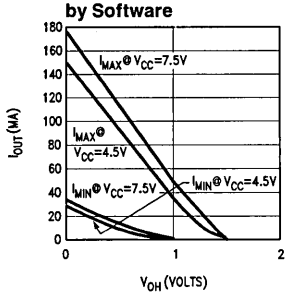
FIGURE 6. COP420H/COP421H/COP422H Input/Output Characteristics

Typical Performance Characteristics (Continued)

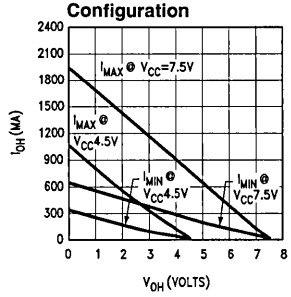
Input Current IN_0–IN_3



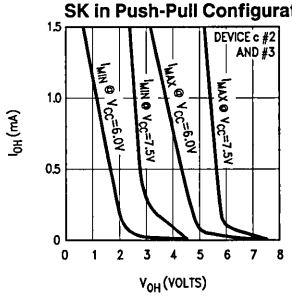
Input Current for L_0–L_7 when Output Programmed OFF by Software



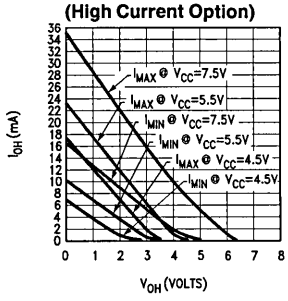
Source Current for Standard Configuration



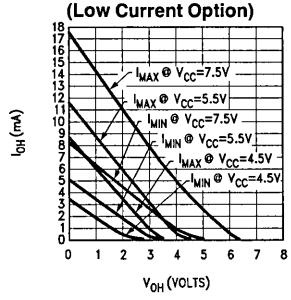
Source Current for SO and SK in Push-Pull Configuration



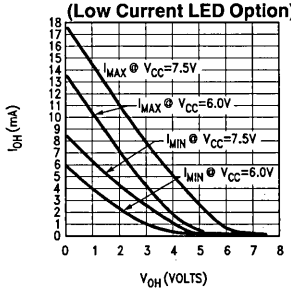
Source Current for L_0–L_7 in TRI-STATE Configuration (High Current Option)



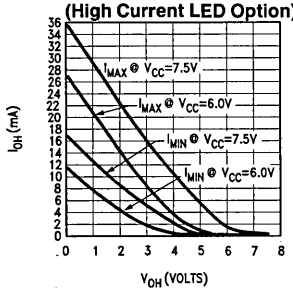
Source Current for L_0–L_7 in TRI-STATE Configuration (Low Current Option)



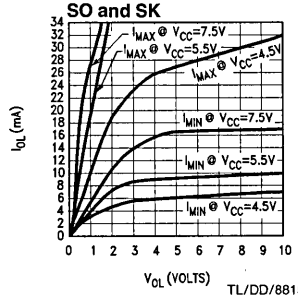
LED Output Source Current (Low Current LED Option)



LED Output Source Current (High Current LED Option)

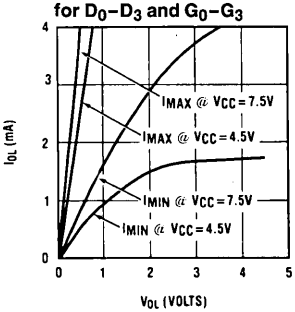


Output Sink Current for SO and SK

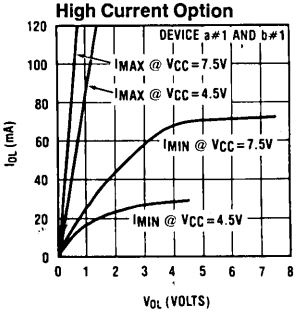


TL/DD/8815–29

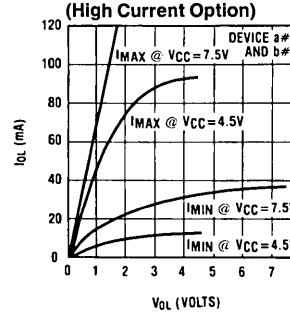
Output Sink Current for L_0–L_7 and Standard Drive Option for D_0–D_3 and G_0–G_3



Output Sink Current for G_0–G_3 and D_0–D_3 with Very High Current Option



Output Sink Current for G_0–G_3 and D_0–D_3 (High Current Option)



TL/DD/8815–27

FIGURE 7. COP320H/COP321H/COP322H Input/Output Characteristics

COP420H/COP421H Instruction Set

Table I is a symbol table providing internal architecture, instruction operand and operational symbols used in the instruction set table.

Table II provides the mnemonic, operand, machine code, data flow, skip conditions and description associated with each instruction in the COP420H/421H instruction set.

TABLE I. COP420H/421H Instruction Set Table Symbols

Symbol	Definition
INTERNAL ARCHITECTURE SYMBOLS	
A	4-bit Accumulator
B	6-bit RAM Address Register
Br	Upper 2 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry Register
D	4-bit Data Output Port
EN	4-bit Enable Register
G	4-bit Register to latch data for G I/O Port
IL	Two 1-bit Latches associated with the IN_3 or IN_0 inputs
IN	4-bit Input Port
L	8-bit TRI-STATE I/O Port
M	4-bit contents of RAM Memory pointed to by B Register
PC	10-bit ROM Address Register (program counter)
Q	8-bit Register to latch data for L I/O Port
SA	10-bit Subroutine Save Register A
SB	10-bit Subroutine Save Register B
SC	10-bit Subroutine Save Register C
SIO	4-bit Shift Register and Counter
SK	Logic-Controlled Clock Output

Symbol	Definition
INSTRUCTION OPERAND SYMBOLS	
d	4-bit Operand Field, 0–15 binary (RAM Digit Select)
r	2-bit Operand Field, 0–3 binary (RAM Register Select)
a	10-bit Operand Field, 0–1023 binary (ROM Address)
y	4-bit Operand Field, 0–15 binary (Immediate Data)
RAM(s)	Contents of RAM location addressed by s
ROM(t)	Contents of ROM location addressed by t
OPERATIONAL SYMBOLS	
+	Plus
−	Minus
→	Replaces
↔	Is exchanged with
=	Is equal to
$\overline{A}$	The ones complement of A
⊕	Exclusive-OR
:	Range of values

Instruction Set (Continued)

TABLE II. COP420H/421H Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	|0011|0000|	A + C + RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	|0011|0001|	A + RAM(B) → A	None	Add RAM to A
ADTD		4A	|0100|1010|	A + 10_{10} → A	None	Add Ten to A
AISC	y	5–	|0101| y |	A + y → A	Carry	Add Immediate, Skip on Carry (y ≠ 0)
CASC		5–	|0001|0000|	$\overline{A}$ + RAM(B) + C → A Carry → C	Carry	Compliment and Add with Carry, Skip on Carry
CLRA		00	|0000|0000|	0 → A	None	Clear A
COMP		40	|0100|0000|	$\overline{A}$ → A	None	Ones complement of A to A
NOP		44	|0100|0100|	None	None	No Operation
RC		32	|0011|0010|	"0" → C	None	Reset C
SC		22	|0010|0010|	"1" → C	None	Set C
XOR		02	|0000|0010|	A ⊕ RAM(B) → A	None	Exclusive-OR RAM with A
TRANSFER OF CONTROL INSTRUCTIONS						
JID		FF	|1111|1111|	ROM ($PC_{9:8}$, A,M) → $PC_{7:0}$	None	Jump Indirect (Note 3)
JMP	a	6– – –	|0110|00|$a_{9:8}$| |$a_{7:0}$|	a → PC	None	Jump
JP	a	– –	|1| $a_{6:0}$ | (pages 2,3 only) or	a → $PC_{6:0}$	None	Jump within Page (Note 4)
		– –	|11| $a_{5:0}$ | (all other pages)	a → $PC_{5:0}$		
JSRP	a	– –	|10| $a_{5:0}$ |	PC + 1 → SA → SB → SC 0010 → $PC_{9:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 5)
JSR	a	6– – –	|0110|10|$a_{9:8}$| |$a_{7:0}$|	PC + 1 → SA → SB → SC a → PC	None	Jump to Subroutine
RET		48	|0100|1000|	SC → SB → SA → PC	None	Return from Subroutine
RETSK		49	|0100|1001|	SC → SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip

Instruction Set (Continued)

TABLE II. COP420H/421H Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
MEMORY REFERENCE INSTRUCTIONS						
CAMQ		33	0011 | 0011	$A \rightarrow Q_{7:4}$	None	Copy A, RAM to Q
		3C	0011 | 1100	$RAM(B) \rightarrow Q_{3:0}$		
CQMA		33	0011 | 0011	$Q_{7:4} \rightarrow RAM(B)$	None	Copy Q to RAM, A
		2C	0010 | 1100	$Q_{3:0} \rightarrow A$		
LD	r	–5	00 | r | 0101	$RAM(B) \rightarrow A$ $Br \oplus r \rightarrow Br$	None	Load RAM into A, Exclusive-OR Br with r
LDD	r,d	23	0010 | 0011	$RAM(r,d) \rightarrow A$	None	Load A with RAM pointed to directly by r,d
		– –	00 | r | d			
LID		33	0011 | 0011			No Skip Condition
		19	0001 | 1001	ROM (PC 9:8, A, M) $\rightarrow$ M, A		Load RAM; A Indirect
LQID		BF	1011 | 1111	$ROM(PC_{9:8},A,M) \rightarrow Q$ $SB \rightarrow SC$	None	Load Q Indirect (Note 3)
RMB	0	4C	0100 | 1100	$0 \rightarrow RAM(B)_0$	None	Reset RAM Bit
	1	45	0100 | 0101	$0 \rightarrow RAM(B)_1$		
	2	42	0100 | 0010	$0 \rightarrow RAM(B)_2$		
	3	43	0100 | 0011	$0 \rightarrow RAM(B)_3$		
SMB	0	4D	0100 | 1101	$1 \rightarrow RAM(B)_0$	None	Set RAM Bit
	1	47	0100 | 1101	$1 \rightarrow RAM(B)_1$		
	2	46	0100 | 0110	$1 \rightarrow RAM(B)_2$		
	3	4B	0100 | 1011	$1 \rightarrow RAM(B)_3$		
STII	y	7–	0111 | y	$y \rightarrow RAM(B)$ $Bd + 1 \rightarrow Bd$	None	Store Memory Immediate and Increment Bd
X	r	–6	00 | r | 0110	$RAM(B) \leftrightarrow A$ $Br \oplus r \rightarrow Br$	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	r,d	23	0010 | 0011	$RAM(r,d) \leftrightarrow A$	None	Exchange A with RAM pointed to directly by (r,d)
		– –	10 | r | d			
XDS	r	–7	00 | r | 0111	$RAM(B) \leftrightarrow A$ $Bd - 1 \rightarrow Bd$ $Br \oplus r \rightarrow Br$	Bd decrements past 0	Exchange RAM with A and Decrement Bd, Exclusive-OR Br with r
XIS	r	–4	00 | r | 0100	$RAM(B) \leftrightarrow A$ $Bd + 1 \rightarrow Bd$ $Br \oplus r \rightarrow Br$	Bd increments past 15	Exchange RAM with A and Increment Bd, Exclusive-OR Br with r

OPTION LIST

The COP420H/421H mask-programmable options are assigned numbers which correspond with the COP*420H* pins.

The following is a list of COP420H options. When specifying a COP421H chip, Options 9, 10, 19, and 20 must all be set to zero. When specifying a COP422H chip, options 9, 10, 19, and 20 must all be set to zero; options 21 and 22 may not be set to one, three or five; and option 2 may not be set to one. The options are programmed at the same time as the ROM pattern to provide the user with the hardware flexibility to interface to various I/O components using little or no external circuitry.

Instruction Set (Continued)

TABLE II. COP420H/421H Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	0101 | 0000	A → Bd	None	Copy A to Bd
CBA		4E	0100 | 1110	Bd → A	None	Copy Bd to A
LBI	r,d	--	00 | r | (d−1) (d=0,9:15) or	r,d → B	Skip until not an LBI	Load B Immediate with r,d (Note 6)
		33	0011 | 0011			
		--	10 | r | d (any d)			
LEI	y	33	0011 | 0011	y → EN	None	Load EN Immediate (Note 7)
		6-	0110 | y			
XABR		12	0001 | 0010	A ⟷ Br (0,0 → A_3,A_2)	None	Exchange A with Br
TEST INSTRUCTIONS						
SKC		20	0010 | 0000		C = "1"	Skip if C is True
SKE		21	0010 | 0001		A = RAM(B)	Skip if A Equals RAM
SKGZ		33	0011 | 0011		$G_{3:0}$ = 0	Skip if G is Zero (all 4 bits)
		21	0010 | 0001			
SKGBZ		33	0011 | 0011	1st byte		Skip if G Bit is Zero
	0	01	0000 | 0001	2nd byte	G_0 = 0	
	1	11	0001 | 0001		G_1 = 0	
	2	03	0000 | 0011		G_2 = 0	
	3	13	0001 | 0011		G_3 = 0	
SKMBZ	0	01	0000 | 0001		$RAM(B)_0$ = 0	Skip if RAM Bit is Zero
	1	11	0001 | 0001		$RAM(B)_1$ = 0	
	2	03	0000 | 0011		$RAM(B)_2$ = 0	
	3	13	0001 | 0011		$RAM(B)_3$ = 0	
SKT		41	0100 | 0001		A time-base counter carry has occurred since last test	Skip on Timer (Note 3)

Instruction Set (Continued)

TABLE II. COP420H/421H Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
INPUT/OUTPUT INSTRUCTIONS						
ING		33	0011 | 0011	G → A	None	Input G Ports to A
		2A	0010 | 1010			
ININ		33	0011 | 0011	IN → A	None	Input IN Inputs to A (Note 2)
		28	0010 | 1000			
INIL		33	0011 | 0011	IL_3, CKO, "0", IL_0 → A	None	Input IL Latches to A (Note 3)
		29	0010 | 1001			
INL		33	0011 | 0011	$L_{7:4}$ → RAM(B)	None	Input L Ports to RAM, A
		2E	0010 | 1110	$L_{3:0}$ → A		
OBD		33	0011 | 0011	Bd → D	None	Output Bd to D Outputs
		3E	0011 | 1110			
OGI	y	33	0011 | 0011	y → G	None	Output to G Ports Immediate
		5–	0101 | y			
OMG		33	0011 | 0011	RAM(B) → G	None	Output RAM to G Ports
		3A	0011 | 1010			
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO (Note 3)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicitly defined (e.g., Br and Bd are explicitly defined). Bits are numbered 0 to N where 0 signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit A register.

Note 2: The ININ instruction is only available on the 28-pin COP420L as the other devices do not contain the IN inputs.

Note 3: For additional information on the operation of the XAS, JID, LQID, INIL, and SKT instructions, see below.

Note 4: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 5: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 6: LBI is a single-byte instruction if d = 0, 9, 10, 11, 12, 13, 14, or 15. The machine code for the lower 4 bits equals the binary value of the "d" data *minus 1*, e.g., to load the lower four bits of B (Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Note 7: Machine code for operand field y for LEI instruction should equal the binary value to be latched into EN, where a "1" or "0" in each bit of EN corresponds with the selection or deselection of a particular function associated with each bit. (See Functional Description, EN Register.)

Description of Selected Instructions

The following information is provided to assist the user in understanding the operation of several unique instructions and to provide notes useful to programmers in writing COP420H/421H programs.

XAS INSTRUCTION

XAS (Exchange A with SIO) exchanges the 4-bit contents of the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. An XAS instruction will also affect the SK output. (See Functional Description, EN Register, above.) If SIO is selected as a shift register, an XAS instruction must be performed once every 4 instruction cycles to effect a continuous data stream.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower 8 bits of the ROM address register PC with the *contents* of ROM addressed by the 10-bit word, $PC_{9:8}$, A, M. PC_9 and PC_8 are not affected by this instruction.

Note that JID requires 2 instruction cycles to execute.

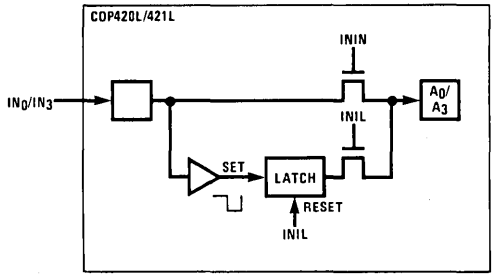
Description of Selected Instructions (Continued)

INIL INSTRUCTION

INIL (Input IL Latches to A) inputs 2 latches, IL_3 and IL_0 (see *Figure 8*) and CKO into A. The IL_3 and IL_0 latches are set if a low-going pulse ("1" to "0") has occurred on the IN_3 and IN_0 inputs since the last INIL instruction, provided the input pulse stays low for at least two instruction times. Execution of an INIL inputs IL_3 and IL_0 into A3 and A0 respectively, and resets these latches to allow them to respond to subsequent low-going pulses on the IN_3 and IN_0 lines. If CKO is mask programmed as a general purpose input, an INIL will input the state of CKO into A2. If CKO has not been so programmed, a "1" will be placed in A2. A "0" is always placed in A1 upon the execution of an INIL. The general purpose inputs IN_3–IN_0 are input to A upon execution of an ININ instruction. (See Table II, ININ instruction.) INIL is useful in recognizing pulses of short duration or pulses which occur too often to be read conveniently by an ININ instruction. IL latches are *not cleared* on reset.

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 10-bit word PC_9, PC_8, A, M. LQID can be used for table lookup or code conversion such as BCD to seven-segment. The LQID instruction "pushes" the stack (PC + 1 → SA → SB → SC) and replaces the least significant 8 bits of PC as follows: A → $PC_{7:4}$, RAM(B) → $PC_{3:0}$, leaving PC_9 and PC_8 unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SC → SB → SA → PC), restoring the saved value of PC to continue sequential program execution. Since LQID pushes SB → SC, the previous contents of SC are lost. Also, when LQID pops the stack, the previously pushed contents of SB are left in SC. The net result is that the contents of SB are placed in SC (SB → SC). Note the LQID takes two instruction cycle times to execute.



TL/DD/8815–28

FIGURE 8. INIL Hardware Implementation

SKT INSTRUCTION

The SKT (Skip On Timer) instruction tests the state of an internal 10-bit time-base counter. This counter divides the instruction cycle clock frequency by 1024 and provides a latched indication of counter overflow. The SKT instruction tests this latch, executing the next program instruction if the latch is not set. If the latch has been set since the previous test, the next program instruction is skipped and the latch is reset. The features associated with this instruction, therefore, allow the COP420H/421H to generate its own time-base for real-time processing rather than relying on an external input signal.

For example, using a 2.097 MHz crystal as the time-base to the clock generator, the instruction cycle clock frequency will be 65 kHz (crystal frequency ÷ 32) and the binary counter output pulse frequency will be 64 Hz. For time-of-day or similar real-time processing, the SKT instruction can call a routine which increments a "seconds" counter every 64 ticks.

INSTRUCTION SET NOTES

a. The first word of a COP420H/421H program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, one instruction cycle time is devoted to skipping each byte of the skipped instruction. Thus all program paths except JID and LQID take the same number of cycle times whether instructions are skipped or executed. JID and LQID instructions take 2 cycles if executed and 1 cycle if skipped.

c. The ROM is organized into 16 pages of 64 words each. The Program Counter is a 10-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID or LQID instruction is located in the last word of a page, the instruction operates as if it were in the next page. For example: a JP located in the last word of a page will jump to a location in the next page. Also, a LQID or JID located in the last word of page 3, 7, 11, or 15 will access data in the next group of four pages.

Option List

The Option Table should be copied and sent in with your EPROM or disc.

Option 1 = 0: Ground Pin—no options available

Option 2: CKO Output
- = 0: clock generator output to crystal/resonator (0 not allowable value if Option 3 = 3)
- = 1: pin is RAM power supply (V_R) input (not available on the COP422L)
- = 2: general purpose input with load device to V_{CC}
- = 3: general purpose input, Hi-Z

Option 3: CKI Input
- = 0: oscillator input divided by 32 (4 MHz max.)
- = 1: oscillator input divided by 16 (4 MHz max.)
- = 2: oscillator input divided by 8 (2 MHz max.)
- = 3: single-pin RC controlled oscillator (÷4)
- = 4: Schmitt trigger clock input (÷4)

Option 4: RESET Input
- = 0: load device to V_{CC}
- = 1: Hi-Z Input

Option 5: L_7 Driver
- = 0: Standard output
- = 1: Open-drain output
- = 2: High current LED direct segment drive output
- = 3: High current TRI-STATE push-pull output
- = 4: Low-current LED direct segment drive output
- = 5: Low-current TRI-STATE push-pull output

Option 6: L_6 Driver
same as Option 5

Option 7: L_5 Driver
same as Option 5

Option 8: L_4 Driver
same as Option 5

Option 9: IN_1 Input
- = 0: load device to V_{CC}
- = 1: Hi-Z input

Option 10: IN_2 Input
same as Option 9

Option 11: V_{CC} pin
- = 0: Standard V_{CC}
- = 1: Optional higher voltage V_{CC}

Option 12: L_3 Driver
same as Option 5

Option 13: L_2 Driver
same as Option 5

Option 14: L_1 Driver
same as Option 5

Option 15: L_0 Driver
same as Option 5

Option 16: SI Input
same as Option 9

Option 17: SO Driver
- = 0: standard output
- = 1: open-drain output
- = 2: push-pull output

Option 18: SK Driver
same as Option 17

Option 19: IN_0 Input
same as Option 9

Option 20: IN_3 Input
same as Option 9

Option 21: G_0 I/O Port
- = 0: very-high current standard output
- = 1: very-high current open-drain output
- = 2: high current open-drain output
- = 3: high current open-drain output
- = 4: standard LSTTL output (fanout = 1)
- = 5: open-drain LSTTL output (fanout = 1)

Option 22: G_1 I/O Port
same as Option 21

Option 23: G_2 I/O Port
same as Option 21

Option 24: G_3 I/O Port
same as Option 21

Option 25: D_3 Output
same as Option 21

Option 26: D_2 Output
same as Option 21

Option 27: D_1 Output
same as Option 21

Option 28: D_0 Output
same as Option 21

Option 29: L Input Levels
- = 0: standard TTL input levels ("0" = 0.8V, "1" = 2.0V)
- = 1: higher voltage input levels ("0" = 1.2V, "1" = 3.6V)

Option 30: IN Input Levels
same as Option 29

Option 31: G Input Levels
same as Option 29

Option 32: SI Input Levels
same as Option 29

Option 33: $\overline{\text{RESET}}$ Input
- = 0: Schmitt trigger input
- = 1: standard TTL input levels
- = 2: higher voltage input levels

Option 34: CKO Input Levels
(CKO = input; Option 2 = 2,3)
same as Option 29

Option 35: Internal Initialization Logic
- = 0: Enabled
- = 1: Disable

Option 36: COP Bonding
- = 0: COP420L (28-pin device)
- = 1: COP421L (24-pin device)
- = 2: 28- and 24-pin versions
- = 3: COP422L (20-pin device)
- = 4: 28- and 20-pin versions
- = 5: 24- and 20-pin versions
- = 6: 28-, 24-, and 20-pin versions

Option 36: Internal Initialization Logic
- = 0: normal operation
- = 1: no internal initialization logic

COP420H/COP421H/COP422H Option Table

Please fill out the Option List and send it with the EPROM.

Option Data

OPTION 1 VALUE = _____ IS: GROUND PIN
OPTION 2 VALUE = _____ IS: CKO OUTPUT
OPTION 3 VALUE = _____ IS: CKI INPUT
OPTION 4 VALUE = _____ IS: RESET INPUT
OPTION 5 VALUE = _____ IS: L (7) DRIVER
OPTION 6 VALUE = _____ IS: L (6) DRIVER
OPTION 7 VALUE = _____ IS: L (5) DRIVER
OPTION 8 VALUE = _____ IS: L (4) DRIVER
OPTION 9 VALUE = _____ IS: IN1 INPUT
OPTION 10 VALUE = _____ IS: IN2 INPUT
OPTION 11 VALUE = _____ IS: VCC PIN
OPTION 12 VALUE = _____ IS: L (3) DRIVER
OPTION 13 VALUE = _____ IS: L (2) DRIVER
OPTION 14 VALUE = _____ IS: L (1) DRIVER
OPTION 15 VALUE = _____ IS: L (0) DRIVER
OPTION 16 VALUE = _____ IS: SI INPUT
OPTION 17 VALUE = _____ IS: SO DRIVER
OPTION 18 VALUE = _____ IS: SK DRIVER
OPTION 19 VALUE = _____ IS: IN0 INPUT
OPTION 20 VALUE = _____ IS: IN3 INPUT
OPTION 21 VALUE = _____ IS: G0 I/O PORT
OPTION 22 VALUE = _____ IS: G1 I/O PORT
OPTION 23 VALUE = _____ IS: G2 I/O PORT
OPTION 24 VALUE = _____ IS: G3 I/O PORT
OPTION 25 VALUE = _____ IS: D3 OUTPUT
OPTION 26 VALUE = _____ IS: D2 OUTPUT
OPTION 27 VALUE = _____ IS: D1 OUTPUT
OPTION 28 VALUE = _____ IS: D0 OUTPUT
OPTION 29 VALUE = _____ IS: L INPUT LEVELS
OPTION 30 VALUE = _____ IS: IN INPUT LEVELS
OPTION 31 VALUE = _____ IS: G INPUT LEVELS
OPTION 32 VALUE = _____ IS: SI INPUT LEVELS
OPTION 33 VALUE = _____ IS: RESET INPUT
OPTION 34 VALUE = _____ IS: CKO INPUT LEVELS
OPTION 35 VALUE = _____ IS: INTERNAL INITIALIZATION LOGIC
OPTION 36 VALUE = _____ IS: COP BONDING

Development Support

The MOLE (Microcontroller On Line Emulator) is a low cost development system and real time emulator for COPS' products. They also include TMP, 8050 and the new 16-bit HPC microcontroller family. The MOLE provides effective support for the development of both software and hardware in the user's application.

The purpose of the MOLE is to provide a tool to write and assemble code, emulate code for the target microcontroller and assist in debugging of the system.

The MOLE can be connected to various hosts, IBM PC, STARPLEX™, Kaypro, Apple and Intel systems, via RS-232 port. This link facilitates the up loading/down loading of code, supports host assembly and mass storage.

The MOLE consists of three parts; brain, personality and optional host software.

The brain board is the computing engine of the system. It is a self-contained computer with its own firmware which provides for all system operation, emulation control, communication, from programming and diagnostic operation. It has three serial ports which can be connected to a terminal, host system, printer, modem or to other MOLE's in a multi-MOLE environment.

The personality board contains the necessary hardware and firmware needed to emulate the target microcontroller. The emulation cable which replaces the target controller attaches to this board. The software contains a cross assembler and a communications program for up loading and down loading code from the MOLE.

MOLE Ordering Information

P/N	Description
MOLE-BRAIN	MOLE Computer Board
MOLE-COPS-PB1	COPS Personality Board
MOLE-XXX-YYY	Optional Software

Where XXX = COPS
YYY = Host System, IBM, Apple, KAY (Kaypro), CP/M

COPS Programming Manual

For detailed information on writing COPS programs, the COPS Programming Manual 424410284-001 provides an in-depth discussion of the COPS architecture, instruction set and general techniques of COPS programming. This manual is written with the programmer in mind.

National Semiconductor Corporation

ADVANCED INFORMATION

COP420P/COP444CP/COP444LP Piggyback EPROM Microcontrollers

General Description

The COP420P, COP444CP, and COP444LP are piggyback versions of the COPS™ microcontroller families. These devices are identical to their respective device except the program ROM has been removed. The device package incorporates the circuitry and socket on top of package to accommodate the piggyback EPROM—MM2716, NMC27C16 or other appropriate EPROMs. With the addition of an EPROM the device performs exactly as its masked equivalent.

The device is a complete microcontroller system with CPU, RAM, I/O and EPROM socket in a 28-lead package. The completed package allows field test of the system in the final electrical and mechanical configuration. This important benefit facilitates development and debug of the COP400 program prior to masking of a production part.

These devices are also economical in low and medium volume applications or when the program may require changing.

Device Selection	Device Emulated	Piggyback Device
Low Power NMOS	COP420L, COP444L	COP444LP
High Speed NMOS	COP420	COP420P
Low Power CMOS	COP424C, COP444C	COP444CP

Features

COP444LP

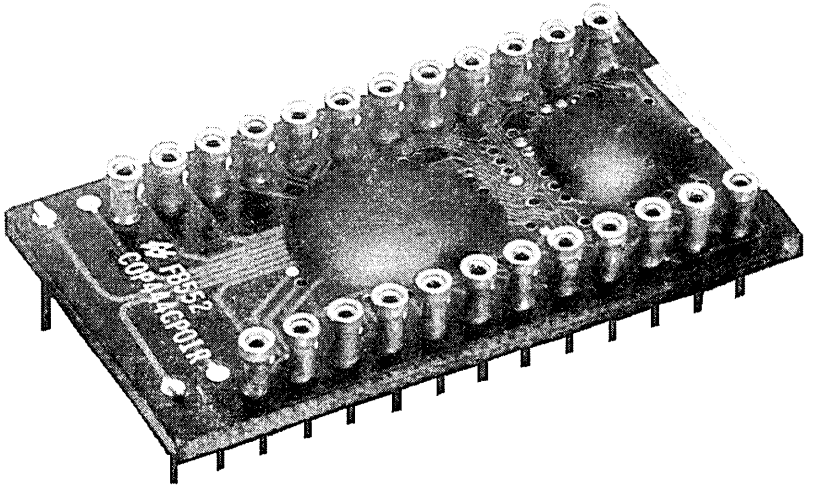
- Low power NMOS
- 16 μs instruction time

COP420P

- 4 μs instruction time

COP444CP

- 4 μs instruction time
- Fully static (can turn off clock)
- Power-saving IDLE state and Halt mode
- Lowest power dissipation—CMOS



TL/DD/8705-10

National Semiconductor Corporation

PRELIMINARY

COP424C, COP425C, COP426C, COP324C, COP325C, COP326C and COP444C, COP445C, COP344C, COP345C Single-Chip 1k and 2k CMOS Microcontrollers

General Description

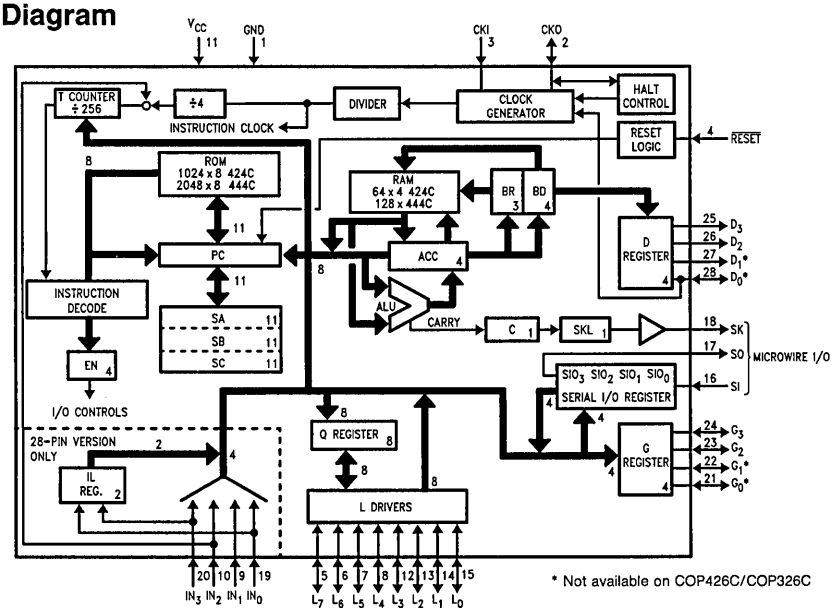
The COP424C, COP425C, COP426C, COP444C and COP445C fully static, Single-Chip CMOS Microcontrollers are members of the COPS™ family, fabricated using double-poly, silicon gate microCMOS technology. These Controller Oriented Processors are complete microcomputers containing all system timing, internal logic, ROM, RAM, and I/O necessary to implement dedicated control functions in a variety of applications. Features include single supply operation, a variety of output configuration options, with an instruction set, internal architecture and I/O scheme designed to facilitate keyboard input, display output and BCD data manipulation. The COP424C and COP444C are 28 pin chips. The COP425C and COP445C are 24-pin versions (4 inputs removed) and COP426C is 20-pin version with 15 I/O lines. Standard test procedures and reliable high-density techniques provide the medium to large volume customers with a customized microcontroller at a low end-product cost. These microcontrollers are appropriate choices in many demanding control environments especially those with human interface.

The COP424C is an improved product which replaces the COP420C.

Features

- Lowest power dissipation (50 μW typical)
- Fully static (can turn off the clock)
- Power saving IDLE state and HALT mode
- 4 μs instruction time, plus software selectable clocks
- 2k x 8 ROM, 128 x 4 RAM (COP444C/COP445C)
- 1k x 8 ROM, 64 x 4 RAM (COP424C/COP425C/COP426C)
- 23 I/O lines (COP444C and COP424C)
- True vectored interrupt, plus restart
- Three-level subroutine stack
- Single supply operation (2.4V to 5.5V)
- Programmable read/write 8-bit timer/event counter
- Internal binary counter register with MICROWIRE™ serial I/O capability
- General purpose and TRI-STATE® outputs
- LSTTL/CMOS output compatible
- Microbus™ compatible
- Software/hardware compatible with COP400 family
- Extended temperature range devices COP324C/COP325C/COP326C and COP344C/COP345C (−40°C to +85°C)
- Military devices (−55°C to +125°C) to be available

Block Diagram



* Not available on COP426C/COP326C

TL/DD/5259–1

FIGURE 1

COP424C/COP425C/COP426C and COP444C/COP445C

Absolute Maximum Ratings

Supply Voltage (V_{CC})	6V
Voltage at any Pin	−0.3V to V_{CC} + 0.3V
Total Allowable Source Current	25 mA
Total Allowable Sink Current	25 mA
Operating Temperature Range	0°C to +70°C
Storage Temperature Range	−65°C to +150°C
Lead Temperature (soldering, 10 seconds)	300°C

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $0°C \le T_A \le 70°C$ unless otherwise specified

Parameter	Conditions	Min	Max	Units
Operating Voltage		2.4	5.5	V
Power Supply Ripple (Note 5)	Peak to Peak		0.1 V_{CC}	V
Supply Current (Note 1)	V_{CC} = 2.4V, tc = 64 μs		120	μA
	V_{CC} = 5.0V, tc = 16 μs		700	μA
	V_{CC} = 5.0V, tc = 4 μs (tc is instruction cycle time)		3000	μA
HALT Mode Current (Note 2)	V_{CC} = 5.0V, F_{IN} = 0 kHz		30	μA
	V_{CC} = 2.4V, F_{IN} = 0 kHz		10	μA
Input Voltage Levels				
RESET, CKI, D_0 (clock input)				
Logic High		0.9 V_{CC}		V
Logic Low			0.1 V_{CC}	V
All Other Inputs				
Logic High		0.7 V_{CC}		V
Logic Low			0.2 V_{CC}	V
Input Pull-Up Current	V_{CC} = 4.5V, V_{IN} = 0	30	330	μA
Hi-Z Input Leakage		−1	+1	μA
Input Capacitance (Note 4)			7	pF
Output Voltage Levels	Standard Outputs			
LSTTL Operation	V_{CC} = 5.0V ± 10%			
Logic High	I_{OH} = −100 μA	2.7		V
Logic Low	I_{OL} = 400 μA		0.4	V
CMOS Operation				
Logic High	I_{OH} = −10 μA	V_{CC} − 0.2		V
Logic Low	I_{OL} = 10 μA		0.2	V
Output Current Levels (except CKO)				
Sink (Note 6)	V_{CC} = 4.5V, V_{OUT} = V_{CC}	1.2		mA
	V_{CC} = 2.4V, V_{OUT} = V_{CC}	0.2		mA
Source (Standard Option)	V_{CC} = 4.5V, V_{OUT} = 0V	0.5		mA
	V_{CC} = 2.4V, V_{OUT} = 0V	0.1		mA
Source (Low Current Option)	V_{CC} = 4.5V, V_{OUT} = 0V	30	330	μA
	V_{CC} = 2.4V, V_{OUT} = 0V	6	80	μA
CKO Current Levels (As Clock Out)				
Sink ÷4	V_{CC} = 4.5V, CKI = V_{CC}, V_{OUT} = V_{CC}	0.3		mA
÷8		0.6		mA
÷16		1.2		mA
Source ÷4	V_{CC} = 4.5V, CKI = 0V, V_{OUT} = 0V	0.3		mA
÷8		0.6		mA
÷16		1.2		mA
Allowable Sink/Source Current per Pin (Note 6)			5	mA
Allowable Loading on CKO (as HALT)			100	pF
Current Needed to Over-Ride HALT (Note 3)				
To Continue	V_{CC} = 4.5V, V_{IN} = 0.2V_{CC}		0.7	mA
To Halt	V_{CC} = 4.5V, V_{IN} = 0.7V_{CC}		1.6	mA
TRI-STATE or Open Drain Leakage Current		−2.5	+2.5	μA

COP324C/COP325C/COP326C and COP344C/COP345C

Absolute Maximum Ratings

Supply Voltage	6V
Voltage at any Pin	−0.3V to V_{CC} + 0.3V
Total Allowable Source Current	25 mA
Total Allowable Sink Current	25 mA
Operating Temperature Range	−40°C to +85°C
Storage Temperature Range	−65°C to +150°C
Lead Temperature (soldering, 10 seconds)	300°C

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $-40°C \le T_A \le +85°C$ unless otherwise specified

Parameter	Conditions	Min	Max	Units
Operating Voltage		3.0	5.5	V
Power Supply Ripple (Note 5)	Peak to Peak		0.1 V_{CC}	V
Supply Current (Note 1)	V_{CC} = 3.0V, tc = 64 μs		180	μA
	V_{CC} = 5.0V, tc = 16 μs		800	μA
	V_{CC} = 5.0V, tc = 4 μs (tc is instruction cycle time)		3600	μA
HALT Mode Current (Note 2)	V_{CC} = 5.0V, F_{IN} = 0 kHz		30	μA
	V_{CC} = 3.0V, F_{IN} = 0 kHz		15	μA
Input Voltage Levels				
RESET, CKI, D_O (clock input)				
Logic High		0.9 V_{CC}		V
Logic Low			0.1 V_{CC}	V
All Other Inputs				
Logic High		0.7 V_{CC}		V
Logic Low			0.2 V_{CC}	V
Input Pull-Up Current	V_{CC} = 4.5V, V_{IN} = 0	30	440	μA
Hi-Z Input Leakage		−2	+2	μA
Input Capacitance (Note 4)			7	pF
Output Voltage Levels	Standard Outputs			
LSTTL Operation	V_{CC} = 5.0V ±10%			
Logic High	I_{OH} = −100 μA	2.7		V
Logic Low	I_{OL} = 400 μA		0.4	V
CMOS Operation				
Logic High	I_{OH} = −10 μA	V_{CC}−0.2		V
Logic Low	I_{OL} = 10 μA		0.2	V
Output Current Levels (except CKO)				
Sink (Note 6)	V_{CC} = 4.5V, V_{OUT} = V_{CC}	1.2		mA
	V_{CC} = 3.0V, V_{OUT} = V_{CC}	0.2		mA
Source (Standard Option)	V_{CC} = 4.5V, V_{OUT} = 0V	0.5		mA
	V_{CC} = 3.0V, V_{OUT} = 0V	0.1		mA
Source (Low Current Option)	V_{CC} = 4.5V, V_{OUT} = 0V	30	440	μA
	V_{CC} = 3.0V, V_{OUT} = 0V	8	200	μA
CKO Current Levels (As Clock Out)				
Sink ÷4	V_{CC} = 4.5V, CKI = V_{CC}, V_{OUT} = V_{CC}	0.3		mA
÷8		0.6		mA
÷16		1.2		mA
Source ÷4	V_{CC} = 4.5V, CKI = 0V, V_{OUT} = 0V	0.3		mA
÷8		0.6		mA
÷16		1.2		mA
Allowable Sink/Source Current per Pin (Note 6)			5	mA
Allowable Loading on CKO (as HALT)			100	pF
Current Needed to Over-Ride HALT (Note 3)				
To Continue	V_{CC} = 4.5V, V_{IN} = 0.2V_{CC}		0.9	mA
To Halt	V_{CC} = 4.5V, V_{IN} = 0.7V_{CC}		2.1	mA
TRI-STATE or Open Drain Leakage Current		−5	+5	μA

COP424C/COP425C/COP426C and COP444C/COP445C

AC Electrical Characteristics $0°C \le T_A \le 70°C$ unless otherwise specified.

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time (tc)	$V_{CC} \ge 4.5V$	4	DC	μs
	$4.5V > V_{CC} \ge 2.4V$	16	DC	μs
Operating CKI Frequency ÷4 mode	$V_{CC} \ge 4.5V$	DC	1.0	MHz
÷8 mode	$V_{CC} \ge 4.5V$	DC	2.0	MHz
÷16 mode	$V_{CC} \ge 4.5V$	DC	4.0	MHz
÷4 mode	$4.5V > V_{CC} \ge 2.4V$	DC	250	kHz
÷8 mode	$4.5V > V_{CC} \ge 2.4V$	DC	500	kHz
÷16 mode	$4.5V > V_{CC} \ge 2.4V$	DC	1.0	MHz
Duty Cycle (Note 4)	$f_1 = 4$ MHz	40	60	%
Rise Time (Note 4)	$f_1 = 4$ MHz External Clock		60	ns
Fall Time (Note 4)	$f_1 = 4$ MHz External Clock		40	ns
Instruction Cycle Time RC Oscillator (Note 4)	$R = 30k$, $V_{CC} = 5V$, $C = 82$ pF (÷4 Mode)	8	16	μs
Inputs: (See *Figure 3*)				
t_{SETUP} G Inputs	$V_{CC} \ge 4.5V$	tc/4+.7		μs
t_{SETUP} SI Input	$V_{CC} \ge 4.5V$	0.3		μs
t_{SETUP} All Others	$V_{CC} \ge 4.5V$	1.7		μs
t_{HOLD}	$V_{CC} \ge 4.5V$	0.25		μs
	$4.5V > V_{CC} \ge 2.4V$	1.0		μs
Output Propagation Delay	$V_{OUT} = 1.5V$, $C_L = 100$ pF, $R_L = 5k$			
t_{PD1}, t_{PD0}	$V_{CC} \ge 4.5V$		1.0	μs
t_{PD1}, t_{PD0}	$4.5V > V_{CC} \ge 2.4V$		4.0	μs
Microbus Timing	$CL = 50$ pF, $V_{CC} = 5V \pm 5\%$			
Read Operation (*Figure 4*)				
Chip Select Stable before $\overline{RD}$ – t_{CSR}		65		ns
Chip Select Hold Time for $\overline{RD}$ – t_{RCS}		20		ns
$\overline{RD}$ Pulse Width – t_{RR}		400		ns
Data Delay from $\overline{RD}$ – t_{RD}			375	ns
$\overline{RD}$ to Data Floating – t_{DF} (Note 4)			250	ns
Write Operation (*Figure 5*)				
Chip Select Stable before $\overline{WR}$ – t_{CSW}		65		ns
Chip Select Hold Time for $\overline{WR}$ – t_{WCS}		20		ns
$\overline{WR}$ Pulse Width – t_{WW}		400		ns
Data Set-Up Time for $\overline{WR}$ – t_{DW}		320		ns
Data Hold Time for $\overline{WR}$ – t_{WD}		100		ns
INTR Transition Time from $\overline{WR}$ – t_{WI}			700	ns

Note 1: Supply current is measured after running for 2000 cycle times with a square-wave clock on CKI, CKO open, and all other pins pulled up to V_{CC} with 5k resistors. See current drain equation on page 17.

Note 2: The HALT mode will stop CKI from oscillating in the RC and crystal configurations. Test conditions: all inputs tied to V_{CC}, L lines in TRI-STATE mode and tied to ground, all outputs low and tied to ground.

Note 3: When forcing HALT, current is only needed for a short time (approx. 200 ns) to flip the HALT flip-flop.

Note 4: This parameter is only sampled and not 100% tested. Variation due to the device included.

Note 5: Voltage change must be less than 0.5 volts in a 1 ms period.

Note 6: SO output sink current must be limited to keep V_{OL} less than $0.2V_{CC}$ when part is running in order to prevent entering test mode.

COP324C/COP325C/COP326C and COP344C/COP345C

AC Electrical Characteristics $-40°C \le T_A \le +85°C$ unless otherwise specified.

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time (tc)	$V_{CC} \ge 4.5V$	4	DC	μs
	$4.5V > V_{CC} \ge 3.0V$	16	DC	μs
Operating CKI Frequency ÷4 mode	$V_{CC} \ge 4.5V$	DC	1.0	MHz
÷8 mode	$V_{CC} \ge 4.5V$	DC	2.0	MHz
÷16 mode	$V_{CC} \ge 4.5V$	DC	4.0	MHz
÷4 mode	$4.5V > V_{CC} \ge 3.0V$	DC	250	kHz
÷8 mode	$4.5V > V_{CC} \ge 3.0V$	DC	500	kHz
÷16 mode	$4.5V > V_{CC} \ge 3.0V$	DC	1.0	MHz
Duty Cycle (Note 4)	$f_1 = 4$ MHz	40	60	%
Rise Time (Note 4)	$f_1 = 4$ MHz external clock		60	ns
Fall Time (Note 4)	$f_1 = 4$ MHz external clock		40	ns
Instruction Cycle Time RC Oscillator (Note 4)	R = 30k, V_{CC} = 5V C = 82 pF (÷4 Mode)	8	16	μs
Inputs: (See *Figure 3*)				
t_{SETUP} G Inputs	$V_{CC} \ge 4.5V$	tc/4+.7		μs
t_{SETUP} SI Inputs	$V_{CC} \ge 4.5V$	0.3		μs
t_{SETUP} All Others	$V_{CC} \ge 4.5V$	1.7		μs
t_{HOLD}	$V_{CC} \ge 4.5V$	0.25		μs
	$4.5V > V_{CC} \ge 3.0V$	1.0		μs
Output Propagation Delay	$V_{OUT} = 1.5V$, $C_L = 100$ pF, $R_L = 5k$			
t_{PD1}, t_{PD0}	$V_{CC} \ge 4.5V$		1.0	μs
t_{PD1}, t_{PD0}	$4.5V > V_{CC} \ge 3.0V$		4.0	μs
Microbus Timing	$C_L = 50$ pF, $V_{CC} = 5V \pm 5\%$			
Read Operation (*Figure 4*)				
Chip Select Stable before $\overline{RD}$ $-t_{CSR}$		65		ns
Chip Select Hold Time for $\overline{RD}$ $-t_{RCS}$		20		ns
$\overline{RD}$ Pulse Width $-t_{RR}$		400		ns
Data Delay from $\overline{RD}$ $-t_{RD}$			375	ns
$\overline{RD}$ to Data Floating $-t_{DF}$ (Note 4)			250	ns
Write Operation (*Figure 5*)				
Chip Select Stable before $\overline{WR}$ $-t_{CSW}$		65		ns
Chip Select Hold Time for $\overline{WR}$ $-t_{WCS}$		20		ns
$\overline{WR}$ Pulse Width $-t_{WW}$		400		ns
Data Set-Up Time for $\overline{WR}$ $-t_{DW}$		320		ns
Data Hold Time for $\overline{WR}$ $-t_{WD}$		100		ns
INTR Transition Time from $\overline{WR}$ $-t_{WI}$			700	ns

Note 1: Supply current is measured after running for 2000 cycle times with a square-wave clock on CKI, CKO open, and all other pins pulled up to V_{CC} with 5k resistors. See current drain equation on page 17.

Note 2: The HALT mode will stop CKI from oscillating in the RC and crystal configurations. Test conditions: all inputs tied to V_{CC}, L lines in TRI-STATE mode and tied to ground, all outputs low and tied to ground.

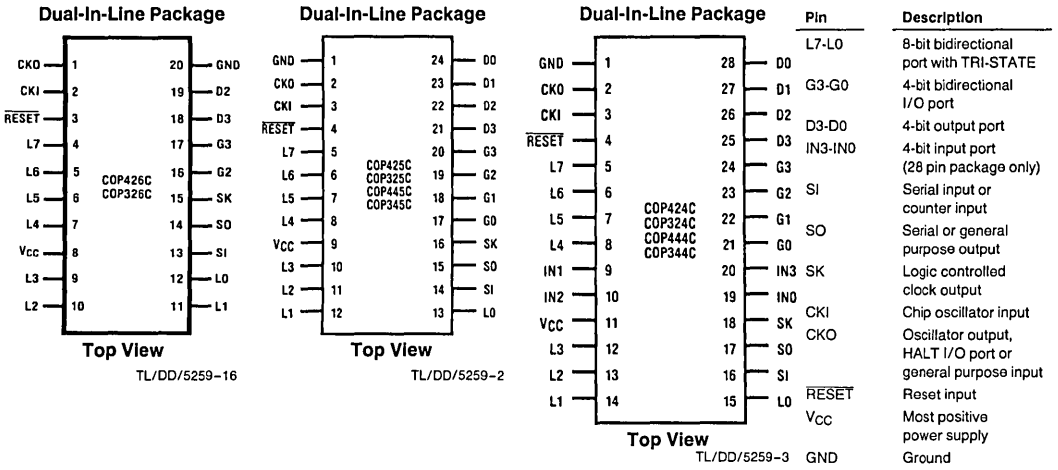
Note 3: When forcing HALT, current is only needed for a short time (approx. 200 ns) to flip the HALT flip-flop.

Note 4: This parameter is only sampled and not 100% tested. Variation due to the device included.

Note 5: Voltage change must be less than 0.5 volts in a 1 ms period.

Note 6: SO output sink current must be limited to keep V_{OL} less than $0.2V_{CC}$ when part is running in order to prevent entering test mode.

Connection Diagrams



Pin	Description
L7-L0	8-bit bidirectional port with TRI-STATE
G3-G0	4-bit bidirectional I/O port
D3-D0	4-bit output port
IN3-IN0	4-bit input port (28 pin package only)
SI	Serial input or counter input
SO	Serial or general purpose output
SK	Logic controlled clock output
CKI	Chip oscillator input
CKO	Oscillator output, HALT I/O port or general purpose input
$\overline{\text{RESET}}$	Reset input
V_{CC}	Most positive power supply
GND	Ground

FIGURE 2

Order Number COP326C/D, C/N, or COP426C/D, C/N
See NS Package D20A or N20A

Order Number COP325C/D, C/N, COP425C/D, C/N, COP345C/D, C/N, or COP445C/D, C/N
See NS Package D24C or N24A

Order Number COP324C/D, C/N, COP424C/D, C/N, COP344C/D, C/N, or COP444C/D, C/N
See NS Package D28C or N28B

Functional Description

The internal architecture is shown in *Figure 1*. Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1", when a bit is reset, it is a logic "0".

For ease of reading only the COP424C/425C/COP426C/444C/445C are referenced; however, all such references apply equally to COP324C/325C/COP326C/344C/345C.

PROGRAM MEMORY

Program Memory consists of ROM, 1024 bytes for the COP424C/425C/426C and 2048 bytes for the COP444C/445C. These bytes of ROM may be program instructions, constants or ROM addressing data.

ROM addressing is accomplished by a 11-bit PC register which selects one of the 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 11-bit binary count value.

Three levels of subroutine nesting are implemented by a three level deep stack. Each subroutine call or interrupt pushes the next PC address into the stack. Each return pops the stack back into the PC register.

DATA MEMORY

Data memory consists of a 512-bit RAM for the COP444C/445C, organized as 8 data registers of 16 × 4-bit digits. RAM addressing is implemented by a 7-bit B register whose upper 3 bits (Br) select 1 of 8 data registers and lower 4 bits (Bd) select 1 of 16 4-bit digits in the selected data register.

Data memory consists of a 256-bit RAM for the COP424C/425C/426C, organized as 4 data registers of 16 × 4-bits digits. The B register is 6 bits long. Upper 2 bits (Br) select 1 of 4 data registers and lower 4 bits (Bd) select 1 of 16 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) are usually loaded into or from, or exchanged with, the A register (accumulator), it may also be loaded into or from the Q latches or T counter or loaded from the L ports. RAM addressing may also be performed directly by the LDD and XAD instructions based upon the immediate operand field of these instructions.

The Bd register also serves as a source register for 4-bit data sent directly to the D outputs.

INTERNAL LOGIC

The processor contains its own 4-bit A register (accumulator) which is the source and destination register for most I/O, arithmetic, logic, and data memory access operations. It can also be used to load the Br and Bd portions of the B register, to load and input 4 bits of the 8-bit Q latch or T counter, to input 4 bits of L I/O ports data, to input 4-bit G, or IN ports, and to perform data exchanges with the SIO register.

A 4-bit adder performs the arithmetic and logic functions, storing the results in A. It also outputs a carry bit to the 1-bit C register, most often employed to indicate arithmetic overflow. The C register in conjunction with the XAS instruction and the EN register, also serves to control the SK output.

The 8-bit T counter is a binary up counter which can be loaded to and from M and A using CAMT and CTMA instructions. This counter may be operated in two modes depending on a mask-programmable option: as a timer or as an external event counter. When the T counter overflows, an overflow flag will be set (see SKT and IT instructions below). The T counter is cleared on reset. A functional block diagram of the timer/counter is illustrated in *Figure 10a*.

Four general-purpose inputs, IN3-IN0, are provided. IN1, IN2 and IN3 may be selected, by a mask-programmable option as Read Strobe, Chip Select, and Write Strobe inputs, respectively, for use in Microbus application.

Functional Description (Continued)

The D register provides 4 general-purpose outputs and is used as the destination register for the 4-bit contents of Bd. In the dual clock mode, D0 latch controls the clock selection (see dual oscillator below).

The G register contents are outputs to a 4-bit general-purpose bidirectional I/O port. G0 may be mask-programmed as an output for Microbus applications.

The Q register is an internal, latched, 8-bit register, used to hold data loaded to or from M and A, as well as 8-bit data from ROM. Its contents are outputted to the L I/O ports when the L drivers are enabled under program control. With the Microbus option selected, Q can also be loaded with the 8-bit contents of the L I/O ports upon the occurrence of a write strobe from the host CPU.

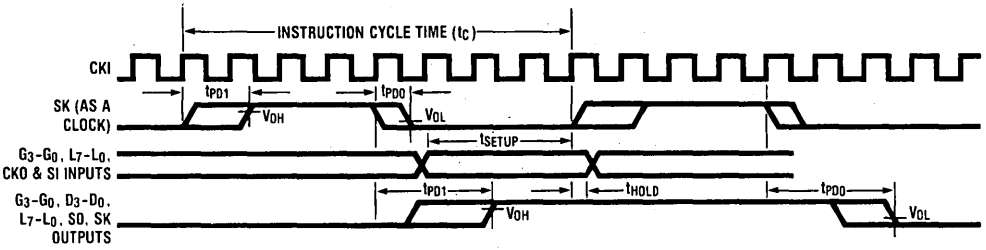
The 8 L drivers, when enabled, output the contents of latched Q data to the L I/O port. Also, the contents of L may be read directly into A and M. As explained above, the Microbus option allows L I/O port data to be latched into the Q register.

The SIO register functions as a 4-bit serial-in/serial-out shift register for MICROWIRE I/O and COPS peripherals, or as a binary counter (depending on the contents of the EN register). Its contents can be exchanged with A.

The XAS instruction copies C into the SKL latch. In the counter mode, SK is the output of SKL; in the shift register mode, SK outputs SKL ANDed with the clock.

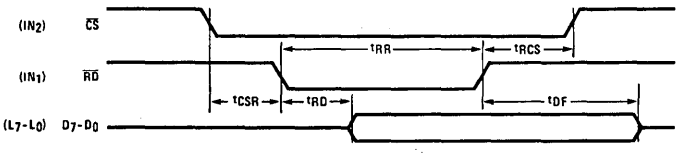
EN is an internal 4-bit register loaded by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register:

0. The least significant bit of the enable register, EN0, selects the SIO register as either a 4-bit shift register or a 4-bit binary counter. With EN0 set, SIO is an asynchronous binary counter, decrementing its value by one upon each low-going pulse ("1" to "0") occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output equals the value of EN3. With EN0 reset, SIO is a serial shift register left shifting 1 bit each instruction cycle time. The data present at SI goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. The SK outputs SKL ANDed with the instruction cycle clock.
1. With EN1 set, interrupt is enabled. Immediately following an interrupt, EN1 is reset to disable further interrupts.
2. With EN2 set, the L drivers are enabled to output the data in Q to the L I/O port. Resetting EN2 disables the L drivers, placing the L I/O port in a high-impedance input state.



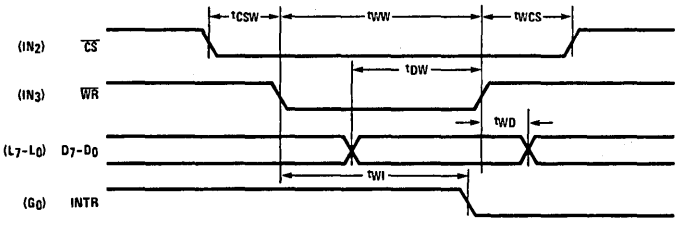
TL/DD/5259-4

FIGURE 3. Input/Output Timing Diagrams (divide by 8 mode)



TL/DD/5259-5

FIGURE 4. Microbus Read Operation Timing



TL/DD/5259-6

FIGURE 5. Microbus Write Operation Timing

Functional Description (Continued)

3. EN3, in conjunction with EN0, affects the SO output. With EN0 set (binary counter option selected) SO will output the value loaded into EN3. With EN0 reset (serial shift register option selected), setting EN3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN3 with the serial shift register option selected disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains set to "0".

INTERRUPT

The following features are associated with interrupt procedure and protocol and must be considered by the programmer when utilizing interrupts.

a. The interrupt, once recognized as explained below, pushes the next sequential program counter address (PC+1) onto the stack. Any previous contents at the bottom of the stack are lost. The program counter is set to hex address 0FF (the last word of page 3) and EN1 is reset.

b. An interrupt will be recognized only on the following conditions:
   1. EN1 has been set.
   2. A low-going pulse ("1" to "0") at least two instruction cycles wide has occurred on the IN_1 input.
   3. A currently executing instruction has been completed.
   4. All successive transfer of control instructions and successive LBIs have been completed (e.g. if the main program is executing a JP instruction which transfers program control to another JP instruction, the interrupt will not be acknowledged until the second JP instruction has been executed).

c. Upon acknowledgement of an interrupt, the skip logic status is saved and later restored upon popping of the stack. For example, if an interrupt occurs during the execution of ASC (Add with Carry, Skip on Carry) instruction which results in carry, the skip logic status is saved and program control is transferred to the interrupt servicing routine at hex address 0FF. At the end of the interrupt routine, a RET instruction is executed to pop the stack and return program control to the instruction following the original ASC. At this time, the skip logic is enabled and skips this instruction because of the previous ASC carry. Subroutines should not be nested within the interrupt service routine, since their popping of the stack will enable any previously saved main program skips, interfering with the orderly execution of the interrupt routine.

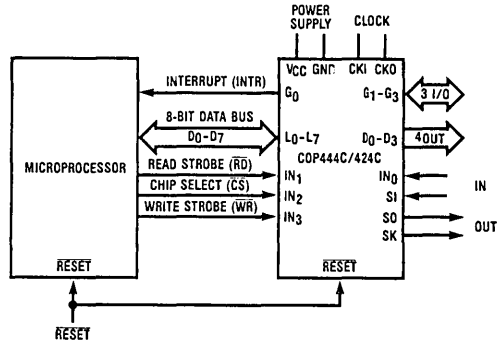
d. The instruction at hex address 0FF must be a NOP.

e. An LEI instruction may be put immediately before the RET instruction to re-enable interrupts.

MICROBUS INTERFACE

The COP444C/424C has an option which allows it to be used as a peripheral microprocessor device, inputting and outputting data from and to a host microprocessor (μP). IN1, IN2 and IN3 general purpose inputs become Microbus compatible read-strobe, chip-select, and write-strobe lines, respectively. IN1 becomes $\overline{RD}$ — a logic "0" on this input will cause Q latch data to be enabled to the L ports for input to the uP. IN2 becomes $\overline{CS}$ — a logic "0" on this line selects the COP444C/424C as the uP peripheral device by enabling the operation of the $\overline{RD}$ and $\overline{WR}$ lines and allows for the selection of one of several peripheral components. IN3 becomes $\overline{WR}$ — a logic "0" on this line will write bus data from the L ports to the Q latches for input to the COP444C/424C. G0 becomes INTR a "ready" output, reset by a write pulse from the uP on the $\overline{WR}$ line, providing the "handshaking" capability necessary for asynchronous data transfer between the host CPU and the COP444C/424C.

This option has been designed for compatibility with National's Microbus — a standard interconnect system for 8-bit parallel data transfer between MOS/LSI CPUs and interfacing devices. (See Microbus National Publication.) The functioning and timing relationships between the signal lines affected by this option are as specified for the Microbus interface, and are given in the AC electrical characteristics and shown in the timing diagrams (*Figures 4* and *5*). Connection of the COP444C/424C to the Microbus is shown in *Figure 6*.



TL/DD/5259–7

FIGURE 6. Microbus Option Interconnect

TABLE I. Enable Register Modes — Bits EN0 and EN3

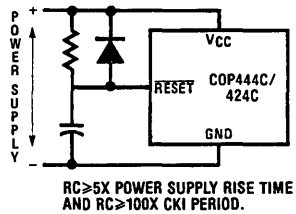
EN0	EN3	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL=1,SK=clock If SKL=0,SK=0
0	1	Shift Register	Input to Shift Register	Serial out	If SKL=1,SK=clock If SKL=0,SK=0
1	0	Binary Counter	Input to Counter	0	SK=SKL
1	1	Binary Counter	Input to Counter	1	SK=SKL

Functional Description (Continued)

INITIALIZATION

The internal reset logic will initialize the device upon power-up if the power supply rise time is less than 1 ms and if the operating frequency at CKI is greater than 32 kHz, otherwise the external RC network shown in *Figure 7* must be connected to the $\overline{\text{RESET}}$ pin (the conditions in *Figure 7* must be met). The $\overline{\text{RESET}}$ pin is configured as a Schmitt trigger input. If not used, it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the $\overline{\text{RESET}}$ input, providing it stays low for at least three instruction cycle times.

Note: If CKI clock is less than 32 kHz, the internal reset logic (option #29 = 1) MUST be disabled and the external RC circuit must be used.



TL/DD/5259–8

FIGURE 7. Power-Up Circuit

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN, IL, T and G registers are cleared. The SKL latch is set, thus enabling SK as a clock output. Data Memory (RAM) is not cleared upon initialization. The first instruction at address 0 must be a CLRA (clear A register).

TIMER

There are two modes selected by mask option:

a. Time-base counter. In this mode, the instruction cycle frequency generated from CKI passes through a 2-bit divide-by-4 prescaler. The output of this prescaler increments the 8-bit T counter thus providing a 10-bit timer. The prescaler is cleared during execution of a CAMT instruction and on reset.

For example, using a 4 MHz crystal with a divide-by-16 option, the instruction cycle frequency of 250 kHz increments the 10-bit timer every 4 μs. By presetting the counter and detecting overflow, accurate timeouts between 16 μs (4 counts) and 4.096 ms (1024 counts) are possible. Longer timeouts can be achieved by accumulating, under software control, multiple overflows.

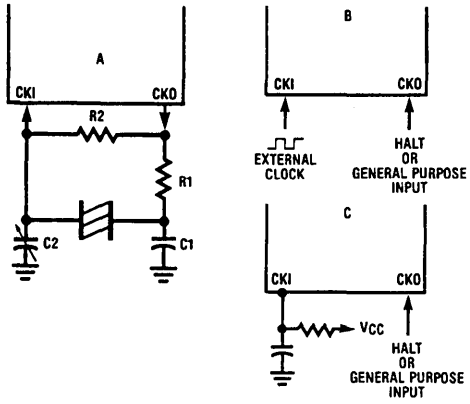
b. External event counter. In this mode, a low-going pulse ("1" to "0") at least 2 instruction cycles wide on the IN2 input will increment the 8-bit T counter.

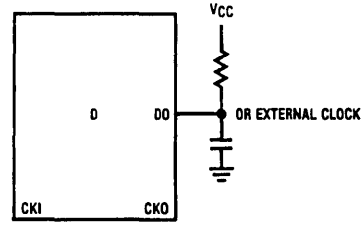
Note: The IT instruction is not allowed in this mode.

HALT MODE

The COP444C/445C/424C/425C/426C is a FULLY STATIC circuit; therefore, the user may stop the system oscillator at any time to halt the chip. The chip may also be halted by the HALT instruction or by forcing CKO high when it is mask-programmed as an HALT I/O port. Once in the HALT mode, the internal circuitry does not receive any clock signal and is therefore frozen in the exact state it was in when halted. All information is retained until continuing. The chip may be awakened by one of two different methods:

- Continue function: by forcing CKO low, if it mask-programmed as an HALT I/O port, the system clock is re-enabled and the circuit continues to operate from the point where it was stopped.
- Restart: by forcing the $\overline{\text{RESET}}$ pin low (see Initialization).





TL/DD/5259–9

Crystal or Resonator

Crystal Value	Component Values			
	R1	R2	C1(pF)	C2(pF)
32 kHz	220k	20M	30	6–36
455 kHz	5k	10M	80	40
2.096 MHz	2k	1M	30	6–36
4.0 MHz	1k	1M	30	6–36

RC Controlled Oscillator

R	C	Cycle Time	V_{CC}
15k	82 pF	4–9 μs	≥4.5V
30k	82 pF	8–16 μs	≥4.5V
60k	100 pF	16–32 μs	2.4–4.5V

Note: 15k≤R≤150k
50 pF≤C≤150 pF

FIGURE 8. Oscillator Component Values

Functional Description (Continued)

The HALT mode is the minimum power dissipation state.

Note: If the user has selected dual-clock with D0 as external oscillator (option 30=2) AND the COP444C/424C is running with the D0 clock, the HALT mode — either hardware or software — will NOT be entered. Thus, the user should switch to the CKI clock to HALT. Alternatively, the user may stop the D0 clock to minimize power.

CKO PIN OPTIONS

a. Two-pin oscillator — (Crystal). See *Figure 9A*.

In a crystal controlled oscillator system, CKO is used as an output to the crystal network. The HALT mode may be entered by program control (HALT instruction) which forces CKO high, thus inhibiting the crystal network. The circuit can be awakened only by forcing the $\overline{\text{RESET}}$ pin to a logic "0" (restart).

b. One-pin oscillator — (RC or external). See *Figure 9B*.

If a one-pin oscillator system is chosen, two options are available for CKO:

- CKO can be selected as the HALT I/O port. In that case, it is an I/O flip-flop which is an indicator of the HALT status. An external signal can over-ride this pin to start and stop the chip. By forcing a high level to CKO, the chip will stop as soon as CKI is high and CKO output will stay high to keep the chip stopped if the external driver returns to high impedance state. By forcing a low level to CKO, the chip will continue and CKO will stay low.
- As another option, CKO can be a general purpose input, read into bit 2 of A (accumulator) upon execution of an INIL instruction.

OSCILLATOR OPTIONS

There are four basic clock oscillator configurations available as shown by *Figure 8*.

a. Crystal Controlled Oscillator. CKI and CKO are connected to an external crystal. The instruction cycle time equals the crystal frequency optionally divided by 4, 8 or 16.

b. External Oscillator. The external frequency is optionally divided by 4, 8 or 16 to give the instruction cycle time. CKO is the HALT I/O port or a general purpose input.

c. RC Controlled Oscillator. CKI is configured as a single pin RC controlled Schmitt trigger oscillator. The instruction cycle equals the oscillation frequency divided by 4. CKO is the HALT I/O port or a general purpose input.

d. Dual oscillator. By selecting the dual clock option, pin D0 is now a single pin oscillator input. Two configurations are available: RC controlled Schmitt trigger oscillator or external oscillator.

The user may software select between the D0 oscillator (in that case, the instruction cycle time equals the D0 oscillation frequency divided by 4) by setting the D0 latch high or the CKI (CKO) oscillator by resetting D0 latch low. Note that even in dual clock mode, the counter, if mask-programmed as a time-base counter, is always connected to the CKI oscillator.

For example, the user may connect up to a 1 MHz RC circuit to D0 for faster processing and a 32 kHz watch crystal to CKI and CKO for minimum current drain and time keeping.

Note: CTMA instruction is not allowed when chip is running from D0 clock.

Figures 10A and *10B* show the clock and timer diagrams with and without Dual clock.

COP445C AND COP425C 24-PIN PACKAGE OPTION

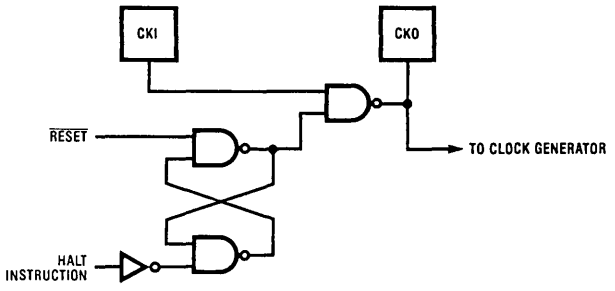
If the COP444C/424C is bonded in a 24-pin package, it becomes the COP445C/425C, illustrated in *Figure 2*, Connection diagrams. Note that the COP445C/425C does not contain the four general purpose IN inputs (IN3–IN0). Use of this option precludes, of course, use of the IN options, interrupt feature, external event counter feature, and the Microbus option which uses IN1–IN3. All other options are available for the COP445C/425C.

Note: If user selects the 24-pin package, options 9, 10, 19 and 20 must be selected as a "0" (load to V_{CC} on the IN inputs). See option list.

COP426C 20-PIN PACKAGE OPTION

If the COP425C is bonded as 20-pin device it becomes the COP426C. Note that the COP426C contains all the COP425C pins except D_0, D_1, G_0, and G_1.

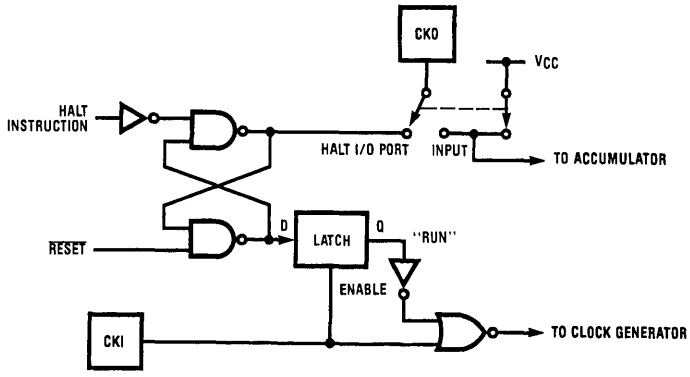
Block Diagram (Continued)



TL/DD/5259–10

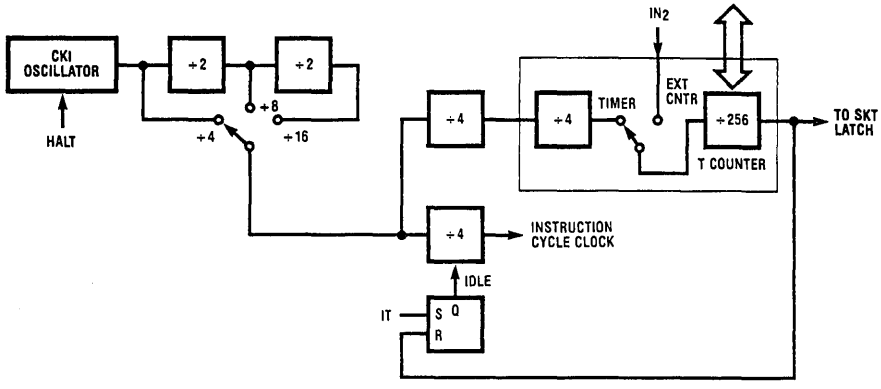
FIGURE 9A. Halt Mode — Two-Pin Oscillator

Block Diagram (Continued)



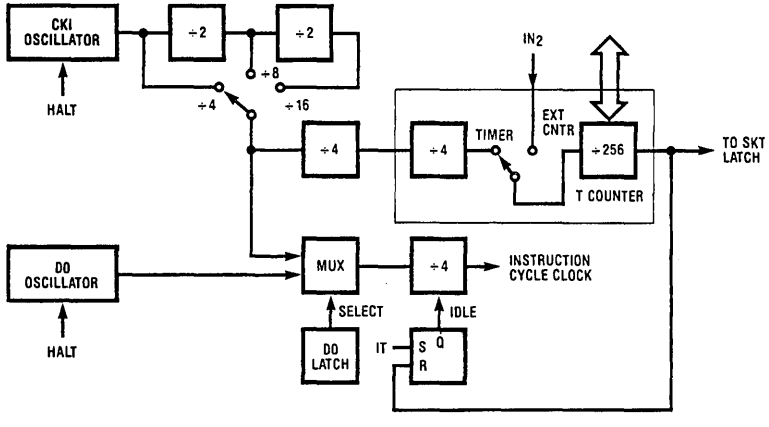
TL/DD/5259–11

FIGURE 9B. Halt Mode — One-Pin Oscillator



TL/DD/5259–12

FIGURE 10A. Clock and Timer without Dual-Clock



TL/DD/5259–13

FIGURE 10B. Clock and Timer with Dual-Clock

Instruction Set

Table II is a symbol table providing internal architecture, instruction operan and operation symbols used in the instruction set table.

TABLE II. Instruction Set Table Symbols

Symbol	Definition
Internal Architecture Symbols	
A	4-bit accumulator
B	7-bit RAM address register (6-bit for COP424C)
Br	Upper 3 bits of B (register address) (2-bit for COP424C)
Bd	Lower 4 bits of B (digit address)
C	1-bit carry register
D	4-bit data output port
EN	4-bit enable register
G	4-bit general purpose I/O port
IL	two 1-bit (IN0 and IN3) latches
IN	4-bit input port
L	8-bit TRI-STATE I/O port
M	4-bit contents of RAM addressed by B
PC	11-bit ROM address program counter
Q	8-bit latch for L port
SA,SB,SC	11-bit 3-level subroutine stack
SIO	4-bit shift register and counter
SK	Logic-controlled clock output
SKL	1-bit latch for SK output
T	8-bit timer

Table III provides the mnemonic, operand, machine code data flow, skip conditions and description of each instruction.

Instruction Operand Symbols	
d	4-bit operand field, 0–15 binary (RAM digit select)
r	3(2)-bit operand field, 0–7(3) binary (RAM register select)
a	11-bit operand field, 0–2047 (1023)
y	4-bit operand field, 0–15 (immediate data)
RAM(x)	RAM addressed by variable x
ROM(x)	ROM addressed by variable x

Operational Symbols	
+	Plus
−	Minus
→	Replaces
⟷	Is exchanged with
=	Is equal to
$\overline{A}$	One's complement of A
⊕	Exclusive-or
:	Range of values

TABLE III. COP444C/445C Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	|0011|0000|	A+C+RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	|0011|0001|	A+RAM(B) → A	None	Add RAM to A
ADT		4A	|0100|1010|	A+10_{10} → A	None	Add Ten to A
AISC	y	5–	|0101| y |	A+y → A	Carry	Add Immediate. Skip on Carry (y ≠ 0)
CASC		10	|0001|0000|	$\overline{A}$+RAM(B)+C → A Carry → C	Carry	Complement and Add with Carry, Skip on Carry
CLRA		00	|0000|0000|	0 → A	None	Clear A
COMP		40	|0100|0000|	$\overline{A}$ → A	None	Ones complement of A to A
NOP		44	|0100|0100|	None	None	No Operation
RC		32	|0011|0010|	"0" → C	None	Reset C
SC		22	|0010|0010|	"1" → C	None	Set C
XOR		02	|0000|0010|	A⊕RAM(B) → A	None	Exclusive-OR RAM with A

Instruction Set (Continued)

Table III. COP444C/445C Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
TRANSFER CONTROL INSTRUCTIONS						
JID		FF	1111 | 1111	ROM ($PC_{10:8}$ A,M) → $PC_{7:0}$	None	Jump Indirect (Notes 1, 3)
JMP	a	6–	0110 | 0 | $a_{10:8}$	a → PC	None	Jump
		––	$a_{7:0}$			
JP	a	––	1 | $a_{6:0}$ (pages 2,3 only)	a → $PC_{6:0}$	None	Jump within Page (Note 4)
			or			
		––	11 | $a_{5:0}$ (all other pages)	a → $PC_{5:0}$		
JSRP	a	––	10 | $a_{5:0}$	PC+1 → SA → SB → SC 00010 → $PC_{10:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 5)
JSR	a	6–	0110 | 1 | $a_{10:8}$	PC+1 → SA → SB → SC a → PC	None	Jump to Subroutine
		––	$a_{7:0}$			
RET		48	0100 | 1000	SC → SB → SA → PC	None	Return from Subroutine
RETSK		49	0100 | 1001	SC → SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
HALT		33	0011 | 0011		None	HALT Processor
		38	0011 | 1000			
IT		33	0011 | 0011			IDLE till Timer
		39	0011 | 1001		None	Overflows then Continues
MEMORY REFERENCE INSTRUCTIONS						
CAMT		33	0011 | 0011	A → $T_{7:4}$		
		3F	0011 | 1111	RAM(B) → $T_{3:0}$	None	Copy A, RAM to T
CTMA		33	0011 | 0011	$T_{7:4}$ → RAM(B)		
		2F	0010 | 1111	$T_{3:0}$ → A	None	Copy T to RAM, A (Note 9)
CAMQ		33	0011 | 0011	A → $Q_{7:4}$	None	Copy A, RAM to Q
		3C	0011 | 1100	RAM(B) → $Q_{3:0}$		
CQMA		33	0011 | 0011	$Q_{7:4}$ → RAM(B)	None	Copy Q to RAM, A
		2C	0010 | 1100	$Q_{3:0}$ → A		
LD	r	–5	00 | r | 0101 (r=0:3)	RAM(B) → A Br⊕r → Br	None	Load RAM into A, Exclusive-OR Br with r
LDD	r,d	23	0010 | 0011	RAM(r,d) → A	None	Load A with RAM pointed to directly by r,d
		––	0 | r | d			
LQID		BF	1011 | 1111	ROM($PC_{10:8}$,A,M) → Q SB → SC	None	Load Q Indirect (Note 3)
RMB	0	4C	0100 | 1100	0 → $RAM(B)_0$	None	Reset RAM Bit
	1	45	0100 | 0101	0 → $RAM(B)_1$		
	2	42	0100 | 0010	0 → $RAM(B)_2$		
	3	43	0100 | 0011	0 → $RAM(B)_3$		
SMB	0	4D	0100 | 1101	1 → $RAM(B)_0$	None	Set RAM Bit
	1	47	0100 | 0111	1 → $RAM(B)_1$		
	2	46	0100 | 0110	1 → $RAM(B)_2$		
	3	4B	0100 | 1011	1 → $RAM(B)_3$		

Instruction Set (Continued)

Table III. COP444C/445C Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
MEMORY REFERENCE INSTRUCTIONS (Continued)						
STII	y	7–	|0111| y |	y → RAM(B) Bd + 1 → Bd	None	Store Memory Immediate 1 and Increment Bd
X	r	–6	|00| r |0110| (r=0:3)	RAM(B) ⟷ A Br ⊕ r → Br	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	r,d	23 – –	|0010|0011| |1| r | d |	RAM(r,d) ⟷ A	None	Exchange A with RAM Pointed to Directly by r,d
XDS	r	–7	|00| r|0111| (r=0:3)	RAM(B) ⟷ A Bd–1 → Bd Br ⊕ r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd. Exclusive-OR Br with r
XIS	r	–4	|00 | r |0100| (r=0:3)	RAM(B) ⟷ A Bd+1 → Bd Br ⊕ r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd, Exclusive-OR Br with r
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	|0101|0000|	A → Bd	None	Copy A to Bd
CBA		4E	|0100|1110|	Bd → A	None	Copy Bd to A
LBI	r,d	– –	|00|r|(d–1)| (r=0:3; d=0,9:15) or 33 – – |1| r | d | (any r, any d)	r,d → B	Skip until not a LBI	Load B Immediate with r,d (Note 6)
LEI	y	33 6–	|0011|0011| |0110| y |	y → EN	None	Load EN Immediate (Note 7)
XABR		12	|0001|0010|	A ⟷ Br	None	Exchange A with Br (Note 8)
TEST INSTRUCTIONS						
SKC		20	|0010|0000|		C="1"	Skip if C is True
SKE		21	|0010|0001|		A=RAM(B)	Skip if A Equals RAM
SKGZ		33 21	|0011|0011| |0010|0001|		$G_{3:0}=0$	Skip if G is Zero (all 4 bits)
SKGBZ		33	|0011|0011|	1st byte		Skip if G Bit is Zero
	0	01	|0000|0001|	2nd byte	$G_0=0$	
	1	11	|0001|0001|		$G_1=0$	
	2	03	|0000|0011|		$G_2=0$	
	3	13	|0001|0011|		$G_3=0$	
SKMBZ	0	01	|0000|0001|		$RAM(B)_0=0$	Skip if RAM Bit is Zero
	1	11	|0001|0001|		$RAM(B)_1=0$	
	2	03	|0000|0011|		$RAM(B)_2=0$	
	3	13	|0001|0011|		$RAM(B)_3=0$	
SKT		41	|0100|0001|		A time-base counter carry has occurred since last test	Skip on Timer (Note 3)

Instruction Set (Continued)

Table III. COP444C/445C Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
INPUT/OUTPUT INSTRUCTIONS						
ING		33	0011 | 0011	G → A	None	Input G Ports to A
		2A	0010 | 1010			
ININ		33	0011 | 0011	IN → A	None	Input IN Inputs to A (Note 2)
		28	0010 | 1000			
INIL		33	0011 | 0011	IL_3, CKO,"0", IL_0 → A	None	Input IL Latches to A (Note 3)
		29	0010 | 1001			
INL		33	0011 | 0011	$L_{7:4}$ → RAM(B)	None	Input L Ports to RAM,A
		2E	0010 | 1110	$L_{3:0}$ → A		
OBD		33	0011 | 0011	Bd → D	None	Output Bd to D Outputs
		3E	0011 | 1110			
OGI	y	33	0011 | 0011	y → G	None	Output to G Ports Immediate
		5–	0101 | y			
OMG		33	0011 | 0011	RAM(B) → G	None	Output RAM to G Ports
		3A	0011 | 1010			
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO (Note 3)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicitly defined (e.g., Br and Bd are explicitly defined). Bits are numbered 0 to N where 0 signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit A register.

Note 2: The ININ instruction is not available on the 24-pin packages since these devices do not contain the IN inputs.

Note 3: For additional information on the operation of the XAS, JID, LQID, INIL, and SKT instructions, see below.

Note 4: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 5: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 6: LBI is a single-byte instruction if d = 0, 9, 10, 11, 12, 13, 14, or 15. The machine code for the lower 4 bits equals the binary value of the "d" data *minus 1*, e.g., to load the lower four bits of B(Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Note 7: Machine code for operand field y for LEI instruction should equal the binary value to be latched into EN, where a "1" or "0" in each bit of EN corresponds with the selection or deselection of a particular function associated with each bit. (See Functional Description, EN Register.)

Note 8: For 2K ROM devices, A ⟷ Br (0 → A3). For 1K ROM devices, A ⟷ Br (0,0 → A3, A2).

Note 9: Do not use CTMA instruction when dual-clock option is selected and part is running from D_0 clocks.

Description of Selected Instructions

XAS INSTRUCTION

XAS (Exchange A with SIO) copies C to the SKL latch and exchanges the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. If SIO is selected as a shift register, an XAS instruction can be performed once every 4 instruction cycles to effect a continuous data stream.

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 11-bit word PC10:PC8,A,M. LQID can be used for table lookup or code conversion such as BCD to seven-segment. The LQID instruction "pushes" the stack (PC+1 → SA → SB → SC) and replaces the least significant 8 bits of the PC as follows: A → PC(7:4), RAM(B) → PC(3:0), leaving PC(10), PC(9) and PC(8) unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SC → SB → SA → PC), restoring the saved value of PC to continue sequential program execution. Since LQID pushes SB → SC, the previous contents of SC are lost.

Note: LQID uses 2 instruction cycles if executed, one if skipped.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower 8 bits of the ROM address register PC with the contents of ROM addressed by the 11-bit word, PC10:8,A,M. PC10,PC9 and PC8 are not affected by JID.

Note: JID uses 2 instruction cycles if executed, one if skipped.

SKT INSTRUCTION

The SKT (Skip On Timer) instruction tests the state of the T counter overflow latch (see internal logic, above), executing the next program instruction if the latch is not set. If the latch has been set since the previous test, the next program instruction is skipped and the latch is reset. The features associated with this instruction allow the processor to generate its own time-base for real-time processing, rather than relying on an external input signal.

Note: If the most significant bit of the T counter is a 1 when a CAMT instruction loads the counter, the overflow flag will be set. The following sample of codes should be used when loading the counter:

```
CAMT   ; load T counter
SKT    ; skip if overflow flag is set and reset it
NOP
```

IT INSTRUCTION

The IT (idle till timer) instruction halts the processor and puts it in an idle state until the time-base counter overflows. This idle state reduces current drain since all logic (except the oscillator and time base counter) is stopped. IT instruction is not allowed if the T counter is mask-programmed as an external event counter (option #31 = 1).

INIL INSTRUCTION

INIL (Input IL Latches to A) inputs 2 latches, IL3 and IL0, CKO and 0 into A. The IL3 and IL0 latches are set if a low-going pulse ("1" to "0") has occurred on the IN3 and IN0 inputs since the last INIL instruction, provided the input pulse stays low for at least two instruction cycles. Execution of an INIL inputs IL3 and IL0 into A3 and A0 respectively, and resets these latches to allow them to respond to subsequent low-going pulses on the IN3 and IN0 lines. If CKO is mask programmed as a general purpose input, an INIL will input the state of CKO into A2. If CKO has not been so programmed, a "1" will be placed in A2. A0 is input into A1. IL latches are cleared on reset. IL latches are not available on the COP445C/425C, and COP426C.

INSTRUCTION SET NOTES

a. The first word of a program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, they are still fetched from the program memory. Thus program paths take the same number of cycles whether instructions are skipped or executed except for JID, and LQID.

c. The ROM is organized into pages of 64 words each. The Program Counter is a 11-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID, or LQID is the last word of a page, it operates as if it were in the next page. For example: a JP located in the last word of a page will jump to a location in the next page. Also, a JID or LQID located in the last word of every fourth page (i.e. hex address 0FF, 1FF, 2FF, 3FF, 4FF, etc.) will access data in the next group of four pages.

Note: The COP424C/425C/426C needs only 10 bits to address its ROM. Therefore, the eleventh bit (P10) is ignored.

Power Dissipation

The lowest power drain is when the clock is stopped. As the frequency increases so does current. Current is also lower at lower operating voltages. Therefore, the user should run at the lowest speed and voltage that his application will allow. The user should take care that all pins swing to full supply levels to insure that outputs are not loaded down and that inputs are not at some intermediate level which may draw current. Any input with a slow rise or fall time will draw additional current. A crystal or resonator generated clock input will draw additional current. An R/C oscillator will draw even more current since the input is a slow rising signal.

If using an external squarewave oscillator, the following equation can be used to calculate operating current drain.

$I_{CO} = I_Q + V \times 40 \times Fi + V \times 1400 \times Fi/Dv$

where I_{CO} = chip operating current drain in microamps
I_Q quiescent leakage current (from curve)
Fi CKI frequency in MegaHertz
V chip V_{CC} in volts
Dv divide by option selected

For example at 5 volts V_{CC} and 400 kHz (divide by 4)

$I_{CO} = 20 + 5 \times 40 \times 0.4 + 5 \times 1400 \times 0.4/4$

$I_{CO} = 20 + 80 + 700 = 800\ \mu A$

At 2.4 volts V_{CC} and 30 kHz (divide by 4)

$I_{CO} = 6 + 2.4 \times 40 \times 0.03 + 2.4 \times 1400 \times 0.03/4$

$I_{CO} = 6 + 2.88 + 25.2 = 34.08\ \mu A$

Power Dissipation (Continued)

If an IT instruction is executed, the chip goes into the IDLE mode until the timer overflows. In IDLE mode, the current drain can be calculated from the following equation:

$$Ici = I_Q + V \times 40 \times Fi$$

For example, at 5 volts V_{CC} and 400 kHz

$$Ici = 20 + 5 \times 40 \times 0.4 = 100\ \mu A$$

The total average current will then be the weighted average of the operating current and the idle current:

$$Ita = I_{CO} \times \frac{To}{To+Ti} + Ici \times \frac{Ti}{To+Ti}$$

where: Ita = total average current
I_{CO} = operating current
Ici = idle current
To = operating time
Ti = idle time

I/O OPTIONS

Outputs have the following optional configurations, illustrated in *Figure 11*:

a. Standard — A CMOS push-pull buffer with an N-channel device to ground in conjunction with a P-channel device to V_{CC}, compatible with CMOS and LSTTL.

b. Low Current — This is the same configuration as a. above except that the sourcing current is much less.

c. Open Drain — An N-channel device to ground only, allowing external pull-up as required by the user's application.

d. Standard TRI-STATE L Output — A CMOS output buffer similar to a. which may be disabled by program control.

e. Low-Current TRI-STATE L Output — This is the same as d. above except that the sourcing current is much less.

f. Open-Drain TRI-STATE L Output — This has the N-channel device to ground only.

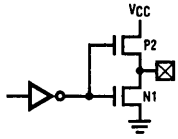
All inputs have the following options:

g. Input with on chip load device to V_{CC}.

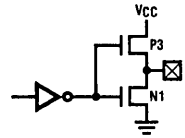
h. Hi-Z input which must be driven by the users logic.

When using either the G or L I/O ports as inputs, a pull-up device is necessary. This can be an external device or the following alternative is available: Select the low-current output option. Now, by setting the output registers to a logic "1" level, the P-channel devices will act as the pull-up load. Note that when using the L ports in this fashion the Q registers must be set to a logic "1" level and the L drivers MUST BE ENABLED by an LEI instruction (see description above).

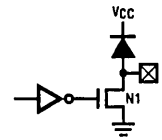
All output drivers use one or more of three common devices numbered 1 to 3. Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figure 12* for each of these devices to allow the designer to effectively use these I/O configurations.



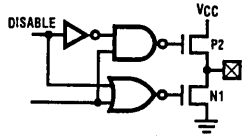
a. Standard Push-Pull Output



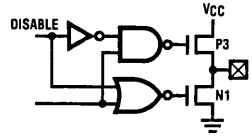
b. Low Current Push-Pull Output



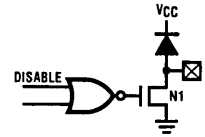
c. Open-Drain Output



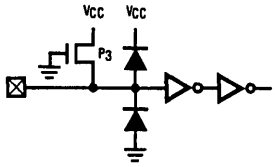
d. Standard TRI-STATE "L" Output



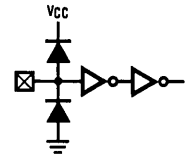
e. Low Current TRI-STATE "L" Output



f. Open Drain TRI-STATE "L" Output



g. Input with Load

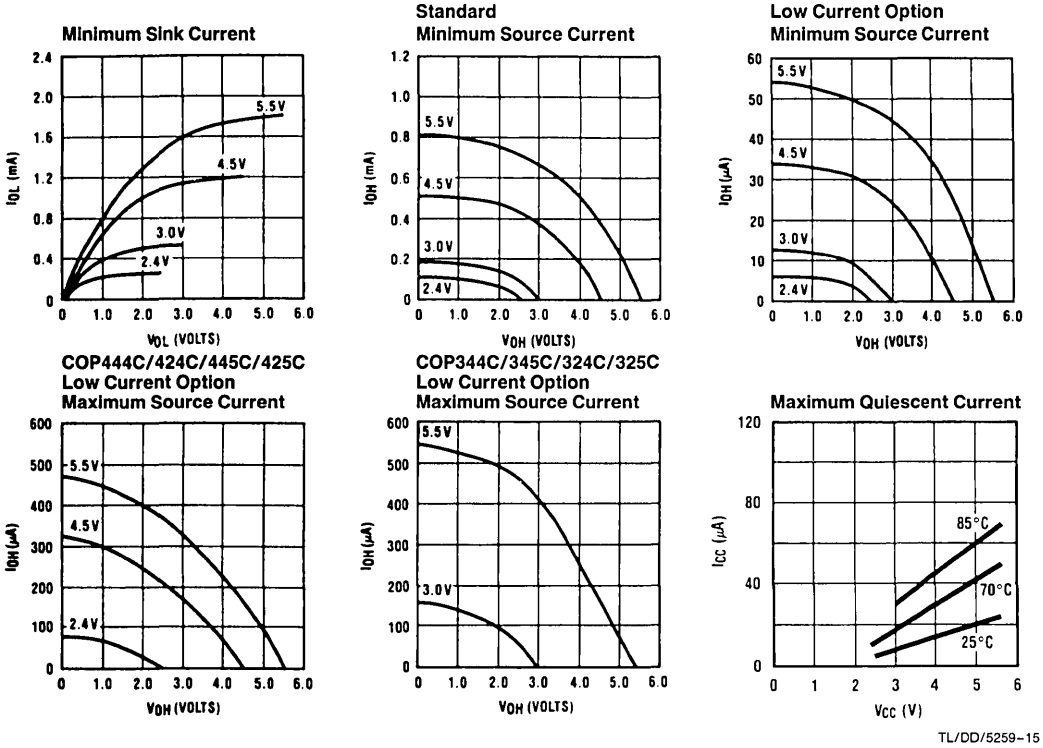


h. Hi-Z Input

TL/DD/5259–14

FIGURE 11. Input/Output Configurations

Power Dissipation (Continued)



TL/DD/5259–15

FIGURE 12. Input/Output Characteristics

Option List

The COP444C/445C/424C/425C/COP426C mask-programmable options are assigned numbers which correspond with the COP444C/424C pins.

The following is a list of options. The options are programmed at the same time as the ROM pattern to provide the user with the hardware flexibility to interface to various I/O components using little or no external circuitry.

PLEASE FILL OUT THE OPTION TABLE on the next page. Xerox the option data and send it in with your disk or EPROM.

Option 1 = 0: Ground Pin — no options available

Option 2: CKO Pin
- = 0: clock generator output to crystal/resonator
- = 1: HALT I/O port
- = 2: general purpose input with load device to V_{CC}
- = 3: general purpose input, high-Z

Option 3: CKI input
- = 0: Crystal controlled oscillator input divide by 4
- = 1: Crystal controlled oscillator input divide by 8
- = 2: Crystal controlled oscillator input divide by 16
- = 4: Single-pin RC controlled oscillator (divide by 4)
- = 5: External oscillator input divide by 4
- = 6: External oscillator input divide by 8
- = 7: External oscillator input divide by 16

Option 4: $\overline{\text{RESET}}$ input
- = 0: load device to V_{CC}
- = 1: Hi-Z input

Option 5: L7 Driver
- = 0: Standard TRI-STATE push-pull output
- = 1: Low-current TRI-STATE push-pull output
- = 2: Open-drain TRI-STATE output

Option 6: L6 Driver — (same as option 5)

Option 7: L5 Driver — (same as option 5)

Option 8: L4 Driver — (same as option 5)

Option 9: IN1 input
- = 0: load device to V_{CC}
- = 1: Hi-Z input

Option 10: IN2 input — (same as option 9)

Option 11 = 0: V_{CC} Pin — no option available

Option 12: L3 Driver — (same as option 5)

Option 13: L2 Driver — (same as option 5)

Option 14: L1 Driver — (same as option 5)

Option 15: L0 Driver — (same as option 5)

Option 16: SI input — (same as option 9)

Option 17: SO Driver
- = 0: Standard push-pull output
- = 1: Low-current push-pull output
- = 2: Open-drain output

Option List (Continued)

Option 18: SK Driver — (same as option 17)
Option 19: IN0 Input — (same as option 9)
Option 20: IN3 Input — (same as option 9)
Option 21: G0 I/O Port — (same as option 17)
Option 22: G1 I/O Port — (same as option 17)
Option 23: G2 I/O Port — (same as option 17)
Option 24: G3 I/O Port — (same as option 17)
Option 25: D3 Output — (same as option 17)
Option 26: D2 Output — (same as option 17)
Option 27: D1 Output — (same as option 17)
Option 28: D0 Output — (same as option 17)
Option 29: Internal Initialization Logic
=0: Normal operation
=1: No internal initialization logic
Option 30: Dual Clock
=0: Normal operation
=1: Dual Clock. D0 RC oscillator
=2: Dual Clock. D0 ext. clock input
} (opt. #28 must=2)
Option 31: Timer
=0: Time-base counter
=1: External event counter
Option 32: Microbus
=0: Normal
=1: Microbus (opt. #31 must=0)
Option 33: COP bonding
(1k and 2K Microcontroller)
=0: 28-pin package
=1: 24-pin package
=2: Same die purchased in both 24 and 28 pin version.
(1K Microcontroller only)
=3: 20-pin package
=4: 28- and 20-pin package
=5: 24- and 20-pin package
=6: 28-, 24- and 20-pin package

Note:—if opt. #33=1 or 2 then opt.#9, 10, 19, 20 and 32 must = 0—if opt. #33=3, 4, 5 or 6 then opt. #9, 10, 19, 20, 21, 22, 30 and 32 must = 0.

Option Table

The following option information is to be sent to National along with the EPROM.

OPTION DATA		OPTION DATA	
OPTION 1 VALUE = _____	IS: GROUND PIN	OPTION 17 VALUE = _____	IS: SO DRIVER
OPTION 2 VALUE = _____	IS: CKO PIN	OPTION 18 VALUE = _____	IS: SK DRIVER
OPTION 3 VALUE = _____	IS: CKI INPUT	OPTION 19 VALUE = _____	IS: IN0 INPUT
OPTION 4 VALUE = _____	IS: RESET INPUT	OPTION 20 VALUE = _____	IS: IN3 INPUT
OPTION 5 VALUE = _____	IS: L(7) DRIVER	OPTION 21 VALUE = _____	IS: G0 I/O PORT
OPTION 6 VALUE = _____	IS: L(6) DRIVER	OPTION 22 VALUE = _____	IS: G1 I/O PORT
OPTION 7 VALUE = _____	IS: L(5) DRIVER	OPTION 23 VALUE = _____	IS: G2 I/O PORT
OPTION 8 VALUE = _____	IS: L(4) DRIVER	OPTION 24 VALUE = _____	IS: G3 I/O PORT
OPTION 9 VALUE = _____	IS: IN1 INPUT	OPTION 25 VALUE = _____	IS: D3 OUTPUT
OPTION 10 VALUE = _____	IS: IN2 INPUT	OPTION 26 VALUE = _____	IS: D2 OUTPUT
OPTION 11 VALUE = _____	IS: VCC PIN	OPTION 27 VALUE = _____	IS: D1 OUTPUT
OPTION 12 VALUE = _____	IS: L(3) DRIVER	OPTION 28 VALUE = _____	IS: D0 OUTPUT
OPTION 13 VALUE = _____	IS: L(2) DRIVER	OPTION 29 VALUE = _____	IS: INT INIT LOGIC
OPTION 14 VALUE = _____	IS: L(1) DRIVER	OPTION 30 VALUE = _____	IS: DUAL CLOCK
OPTION 15 VALUE = _____	IS: L(0) DRIVER	OPTION 31 VALUE = _____	IS: TIMER
OPTION 16 VALUE = _____	IS: SI INPUT	OPTION 32 VALUE = _____	IS: MICROBUS
		OPTION 33 VALUE = _____	IS: COP BONDING

National Semiconductor Corporation

PRELIMINARY

COP427C, COP428C, COP429C, COP447C, COP448C, COP449C, COP327C, COP328C, COP329C, COP227C, COP228C, COP229C, and COP347C, COP348C, COP349C Single Chip 1k and 2k CMOS Microcontrollers

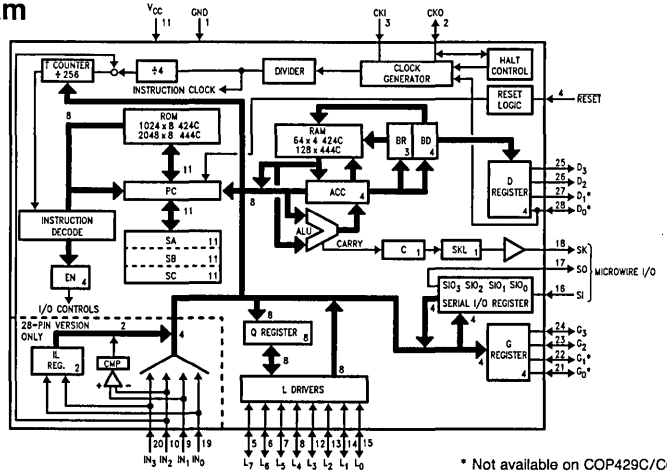
General Description

The COP427C, COP428C, COP429C, COP447C, COP448C and COP449C fully static, Single-Chip CMOS Microcontrollers are members of the COPS™ family, fabricated using double-poly, silicon gate microCMOS technology. These Controller Oriented Processors are complete microcomputers containing all system timing, internal logic, ROM, RAM, and I/O necessary to implement dedicated control functions in a variety of applications. Features include single supply operation, a variety of output configuration options, with an instruction set, internal architecture and I/O scheme designed to facilitate keyboard input, display output and BCD data manipulation. The COP427C and COP447C are 28 pin chips. The COP428C and COP448C are 24-pin versions (4 inputs removed) and COP429C and COP449C are 20-pin versions with 15 I/O lines. Standard test procedures and reliable high-density techniques provide the medium to large volume customers with a customized microcontroller at a low end-product cost. These microcontrollers are appropriate choices in many demanding control environments especially those with human interface.

Features

- 2 μs instruction time, plus software selectable clocks
- Lowest power dissipation (25 μW typical)
- Fully static (can turn off the clock)
- Power saving IDLE state and HALT mode
- 2k x 8 ROM, 128 x 4 RAM (COP447C/COP448C)
- 1k x 8 ROM, 64 x 4 RAM (COP427C/COP428C/COP429C)
- 23 I/O lines (COP447C and COP427C)
- True vectored interrupt, plus restart
- Three-level subroutine stack
- Single supply operation (2.4V to 6.0V)
- Programmable read/write 8-bit timer/event counter
- Internal binary counter register with MICROWIRE™ serial I/O capability
- General purpose and TRI-STATE® outputs
- LSTTL/CMOS output compatible
- MICROBUS™ compatible
- Software/hardware compatible with COP400 family
- Open-drain outputs and Hi-Z inputs operational at 12 volts
- Internal initialization on power-up optionally available
- External HALT mode wake-up optionally available
- On-chip comparator with software enable provided (IN1, IN2)
- High-sink current optionally available
- Malfunction detector option available
- MICROBUS compatible
- Extended temperature range devices COP327C/COP328C/COP329C and COP347C/COP348C/COP349C (−40°C to +85°C)
- Militay temperature range devices COP227C/COP228C/COP229C and COP247C/COP248C/COP249C (−55°C to +125°C)

Block Diagram



TL/DD/9129-1

* Not available on COP429C/COP449C

FIGURE 1

COP427C/COP428C/COP429C and COP447C/COP448C/COP449C
COP327C/COP328C/COP329C and COP347C/COP348C/COP349C

Absolute Maximum Ratings $0°C \le T_A \le +70°C$

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage (V_{CC})	−0.3V to 7.0V
Latch-Up Susceptibility (Note 8)	400 mW
ESD Susceptibility (Note 9)	3500V
Maximum Voltage at Pin (Hi-Z Inputs and Open Drain Outputs)	14V
Total Allowable Sink Current	60 mA
Total Allowable Source Current	48 mA
Allowable Source Current per Pin	5 mA
Power Dissipation (T = 110°C, V_{CC} = 6V)	500 mW
Storage Temperature Range	−65°C to +150°C
Lead Temperature (soldering, 10 sec.)	300°C

DC Electrical Characteristics

$0°C \le T_A \le 70°C$ for COP400 Series, $-40°C \le T_A \le 85°C$ for COP300 Series

Parameter	Conditions	Min	Max	Units
Operating Voltage		2.4	6.0	V
Power Supply Ripple (Note 5)	Peak to Peak		0.1 V_{CC}	V
Supply Current (Notes 1, 7)	V_{CC} = 6.0V, tc = 4 μs, SK = CKI/32		6.2	mA
	V_{CC} = 6.0V, tc = 2 μs, SK = CKI/16		6.2	mA
	V_{CC} = 6.0V, tc = 2 μs, SK = CKI/4		1.1	mA
	V_{CC} = 2.4V, tc = 16 μs, SK = CKI/32		250	μA
	V_{CC} = 2.4V, tc = 8 μs, SK = CKI/16		250	μA
	V_{CC} = 2.4V, tc = 8 μs, SK = CKI/4		55	μA
HALT Mode Current (Note 2)	V_{CC} = 6.0V		5	μA
	V_{CC} = 3.0V		3	μA
Input Voltage Levels				
RESETp, CKI, D_0 (clock input)				
Logic High		0.8 V_{CC}		V
Logic Low			0.2 V_{CC}	V
All Other Inputs				
Logic High		0.7 V_{CC}		V
Logic Low			0.3 V_{CC}	V
G-Port				
Logic High	Optional SCHMITT Trigger	0.8 V_{CC}		V
Logic Low			0.2 V_{CC}	V
IN1, IN2 as Comparator Inputs				
Common-Mode Voltage		0	V_{CC}-1.5	V
Input Offset Voltage		100		mV
Input Pull-Up Current (optional)	V_{CC} = 4V, V_{IN} = 0V	30	120	μA
Hi-Z Input Leakage	V_{IN} = V_{CC}	−1	+1	μA
	V_{IN} = 12V	−15	+15	
Input Capacitance (Note 4)			7	pF

COP427C/COP428C/COP429C and COP447C/COP448C/COP449C
COP327C/COP328C/COP329C and COP347C/COP348C/COP349C

DC Electrical Characteristics

$0°C \le T_A \le 70°C$ for COP400 Series, $-40°C \le T_A \le 85°C$ for COP300 Series (Continued)

Parameter	Conditions	Min	Max	Units
Output Voltage Levels	(Standard Outputs)			
LSTTL Operation	V_{CC} = 4.0V to 6.0V			
Logic High	$I_{OH} = -500\ \mu A$	2.7		V
Logic Low	$I_{OL} = 600\ \mu A$		0.4	V
CMOS Operation	$2.4V \le V_{CC} \le 6.0V$			
Logic High	$I_{OH} = -50\ \mu A$	$V_{CC} - 0.2$		V
Logic Low	$I_{OL} = 175\ \mu A$		0.2	V
Output Current Levels (except CKO)				
Sink (Note 6)				
High Current Option	V_{CC} = 4V, V_{OUT} = 2.0V	15.0		mA
Source				
Low Current Option	V_{CC} = 4V, V_{OUT} = 2.7V	50.0	300	μA
	V_{CC} = 2.4V, V_{OUT} = 2.2V	5.0	30	μA
CKO Current Levels (As Clock Out)				
Sink ÷4		0.44		mA
All Others	V_{CC} = 4V, CKI = V_{CC}, V_{OUT} = V_{CC}	0.88		mA
Source ÷4		0.44		mA
All Others	V_{CC} = 4V, CKI = 0V, V_{OUT} = 0V	0.88		mA
Allowable Loading on CKO (as HALT)			100	pF
Current Needed to Over-Ride HALT (Note 3)				
To Continue	V_{CC} = 4V, V_{IN} = 0.3 V_{CC}		0.7	mA
To Halt	V_{CC} = 4V, V_{IN} = 0.7 V_{CC}		1.6	mA
TRI-STATE or Open Drain Leakage	$0V \le V_{OUT} \le V_{CC}$	−2.5	+2.5	μA
Open-Drain Leakage	V_{OUT} = 12V	−15	+15	μA
Total Sink Current Allowed				
All Outputs Combined			60	mA
D, G Ports			60	mA
D or G per pin			20	mA
All Other Pins			5	mA
Total Source Current Allowed				
All I/O Combined			48	mA
Each Pin			5	mA

COP427C/COP428C/COP429C and COP447C/COP448C/COP449C
COP327C/COP328C/COP329C and COP347C/COP348C/COP349C

Electrical Characteristics

$0°C \le T_A \le +70°C$ for COP400 Series, unless otherwise specified, $-40°C \le T_A \le 85°C$ for COP300 Series (Continued)

Parameter	Conditions	Min	Max	Units
Instruction Cycle	$V_{CC} \ge 4V$	2.0	DC	μs
Time (tc)	$2.4V \le V_{CC} < 4V$	8.0	DC	μs
Operating CKI Frequency	$\div 4$, $V_{CC} \ge 4V$	DC	2.0	MHz
	$\div 8$, $V_{CC} \ge 4V$	DC	4.0	MHz
	$\div 16$, $V_{CC} \ge 4V$	DC	8.0	MHz
	$\div 32$, $V_{CC} \ge 4V$	DC	8.0	MHz
	Crystal $\div 16$, $V_{CC} \ge 4V$	DC	4.0	MHz
	Crystal $\div 32$, $V_{CC} > 4V$	DC	4.0	MHz
	$\div 4$, $2.4V \le V_{CC} < 4V$	DC	0.5	MHz
	$\div 8$, $2.4V \le V_{CC} < 4V$	DC	1.0	MHz
	$\div 16$, $2.4V < V_{CC} < 4V$	DC	2.0	MHz
	$\div 32$, $2.4V \le V_{CC} < 4V$	DC	2.0	MHz
	Crystal $\div 16$, $2.4V \le V_{CC} < 4V$	DC	1.0	MHz
	Crystal $\div 32$, $2.4V < V_{CC} < 4V$	DC	1.0	MHz
Duty Cycle (Note 4)	F1 = 8 MHz	40	60	%
Rise Time (Note 4)	F1 = 8 MHz		15	ns
Fall Time (Note 4)	F1 = 8 MHz		10	ns
Instruction Cycle Time RC Oscillator (Note 4)	$R = 15\ K\Omega$, $V_{CC} = 5V$ $C = 82$ pF ($\div 4$ mode)	4	8	μs
Inputs (See *Figure 3*) T_{SETUP}	G Inputs, $V_{CC} \ge 4V$	tc/4 + 0.35		μs
	SI Input, $V_{CC} \ge 4V$	0.15		μs
	All others, $V_{CC} \ge 4V$	0.85		μs
T_{HOLD}	$V_{CC} \ge 4V$	0.2		μs
	$4V > V_{CC} \ge 2.4V$	0.5		μs
Output Propagation Delay	$V_{OUT} = 1.5V$, $CL = 100$ pF, $R_L = 5K$			
T_{PD1}, T_{PD0}	$V_{CC} = 4V$		600	ns
T_{PD1}, T_{PD0}	$4V > V_{CC} > = 2.4V$		2.0	μs
MICROBUS Timing	$C_L = 50$ pF, $V_{CC} = 5V \pm 5\%$			
Read Operation *(Figure 4)*				
Chip Select Stable before RD $-t_{CSR}$		65		ns
Chip Select Hold Time for RD $-t_{CSR}$		20		ns
RD Pulse Width $-t_{RR}$		400		ns
Data Delay from RD $-t_{RD}$			375	ns
RD to Data Floating $-t_{DF}$ (Note 4)			250	ns
Write Operation *(Figure 5)*				
Chip Select Stable before WR $-t_{CSW}$		65		ns
Chip Select Hold Time for WR $-t_{WCS}$		20		ns
WR Pulse Width $-t_{WW}$		400		ns
Data Setup Time for WR $-t_{DW}$		320		ns
Data Hold Time for WR $-t_{WD}$		100		ns
INTR Transition Time from WR $-t_{W1}$			700	ns

Note 1: Supply current is measured after running 2000 cycle times with a square-wave clock on CKI, CKO open, and all other pins pulled to V_{CC} with 20K resistors. See current drain equation, see power dissipation.

Note 2: The HALT mode will stop CKI from oscillating in the RC and crystal configurations. Test conditions: all inputs tied to V_{CC}, L lines in TRI-STATE mode and tied to ground, all outputs low and tied to ground.

Note 3: When forcing HALT, current is only needed for a short time (approx. 200 ns) to flip the HALT flip-flop.

Note 4: This parameter is only sampled and not 100% tested.

Note 5: Voltage change must be less than 0.5 volts in a 1 ms period.

Note 6: SO output sink curent must be limited to keep V_{OL} less than 0.3 V_{CC} when part is running in order to prevent entering test mode.

Note 7: If CKI is crystal divide by 16 option, maximum CKI frequency is 4 MHz, $t_c = 4$ μs for $V_{CC} \ge 4V$ and 1 MHz, $t_c = 16$ μs for $2.4V \le V_{CC} < 4V$. If CKI is crystal $\div 32$ option, maximum CKI frequency is 4 MHz, $t_c = 8$ μs for $V_{CC} \ge 4V$ and 1 MHz, $t_c = 32$ μs for $2.4V \le V_{CC} < 4V$.

Note 8: I/O pin voltage held above V_{CC} or below GND (e.g., +2V/−2V with +200 mA/−200 mA, or +20V/−20V with +20 mA/−20 mA).

Note 9: Human Body Model, 100 pF through 1.5 kΩ.

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage (V_{CC})	−0.3V to 7V
Latch-Up Susceptibility (Note 8)	400 mW
ESD Susceptibility (Note 9)	3500V
Maximum Voltage at Pin (Hi-Z Inputs and Open Drain Outputs)	14V
Total Allowable Sink Current	60 mA
Total Allowable Source Current	48 mA
Allowable Source Current Per Pin	5 mA
Power Dissipation (T = 125°C V_{CC} = 6V)	500 mW
Operating Temperature Range	−55°C to +125°C
Storage Temperature Range	−65°C to +150°C
Lead Temperature (soldering, 10 sec.)	300°C

DC Electrical Characteristics $-55°C \le T_A \le +125°C$ unless otherwise specified

Parameter	Conditions	Min	Max	Units
Operating Voltage		2.4	6.0	V
Power Supply Ripple (Note 5)	Peak to Peak		0.1 V_{CC}	V
Supply Current (Notes 1, 7)	V_{CC} = 6.0V, tc = 5.0 μs, SK = CKI/32		9.5	mA
	V_{CC} = 6.0V, tc = 2.5 μs, SK = CKI/16		9.5	mA
	V_{CC} = 6.0V, tc = 2.5 μs, SK = CKI/4		2.4	mA
	V_{CC} = 2.4V, tc = 20.0 μs, SK = CKI/32		1.2	mA
	V_{CC} = 2.4V, tc = 10.0 μs, SK = CKI/16		1.2	mA
	V_{CC} = 2.4V, tc = 10.0 μs, SK = CKI/4		330	μA
HALT Mode Current (Note 2)	V_{CC} = 6.0V		25	μA
	V_{CC} = 3.0V		12	μA
Input Voltage Levels				
RESETp, CKI,				
D_O (clock input)				
Logic High		0.8 V_{CC}		V
Logic Low			0.2 V_{CC}	V
All Other Inputs				
Logic High		0.7 V_{CC}		V
Logic Low			0.3 V_{CC}	V
G-Port				
Logic High	Optional SCHMITT Trigger	0.8 V_{CC}		
Logic Low			0.2 V_{CC}	V
IN1, IN2 as				
Comparator Inputs				
Common-mode Voltage		0	V_{CC}−1.5	V
Input Offset Voltage		100		mV
Input Pull-up Current (Optional)	V_{CC} = 4V, V_{IN} = 0V	15	200	μA
Hi-Z Input Leakage	V_{IN} = V_{CC}	−10	+10	μA
	V_{IN} = 12V	−50	+50	μA
Input Capacitance (Note 4)			7	pF
Output Voltage Levels				
LSTTL Operation	V_{CC} = 4.0V to 6.0			
Logic High	I_{OH} = −450 μA	2.7		V
Logic Low	I_{OL} = 550 μA		0.4	V
CMOS Operation	2.4V ≤ V_{CC} ≤ 6.0V			
Logic High	I_{OH} = −40 μA	V_{CC}−0.2		V
Logic Low	I_{OL} = 150 μA		0.2	V

COP227C/COP228C/COP229C and COP247C/COP248C/COP249C

DC Electrical Characteristics $-55°C \le T_A \le +125°C$ unless otherwise specified (Continued)

Parameter		Conditions	Min	Max	Units
Output Current Levels (except CKO)					
Sink (Note 6)					
High Current Option		$V_{CC} = 4V, V_{OUT} = 2.0V$	12.0		mA
Source					
Low Current Option		$V_{CC} = 4V, V_{OUT} = 2.7V$	45	325	μA
		$V_{CC} = 2.4V, V_{OUT} = 2.2V$	4	35	μA
CKO Current Levels (As Clock Out)					
Sink	÷4		0.3		mA
	All Others	$V_{CC} = 4V$, CKI = V_{CC}, $V_{OUT} = V_{CC}$	0.7		mA
Source	÷4		0.3		mA
	All Others	$V_{CC} = 4V$, CKI = 0V, $V_{OUT} = 0V$	0.7		mA
Allowable Loading on CKO (as HALT)				50	pF
Current Needed to Over-Ride HALT (Note 3)					
To Continue		$V_{CC} = 4V, V_{IN} = 0.3\ V_{CC}$		2.0	mA
To Halt		$V_{CC} = 4V, V_{IN} = 0.7\ V_{CC}$		3.0	mA
TRI-STATE or Open Drain Leakage		$0V \le V_{OUT} \le V_{CC}$	−10	+10	μA
Open-drain Leakage		$V_{OUT} = 12V$	−50	+50	μA
Total Sink Current Allowed					
All Outputs Combined				60	mA
D, G Ports				60	mA
D or G per Pin				20	mA
All Other Pins				5	mA
Total Source Current Allowed					
All I/O Combined				48	mA
Each Pin				5	mA

COP227C/COP228C/COP229C and COP247C/COP248C/COP249C

Electrical Characteristics $-55°C \le T_A \le 125°C$ unless otherwise specified

Parameter	Conditions	Min	Max	Units
Instruction Cycle	$V_{CC} \ge 4V$	2.5	DC	μs
Time (tc)	$2.4V \le V_{CC} < 4V$	10.0	DC	μs
Operating CKI Frequency	÷4, $V_{CC} \ge 4V$	DC	1.6	MHz
	÷8, $V_{CC} \ge 4V$	DC	3.2	MHz
	÷16, $V_{CC} \ge 4V$	DC	6.4	MHz
	Crystal ÷16, $V_{CC} \ge 4V$	DC	3.2	MHz
	÷4, $2.4V \le V_{CC} < 4V$	DC	0.4	MHz
	÷8, $2.4V \le V_{CC} < 4V$	DC	0.8	MHz
	÷16, $2.4V \le V_{CC} < 4V$	DC	1.6	MHZ
	Crystal ÷16, $2.4 \le V_{CC} < 4V$	DC	0.8	MHz
Duty Cycle (Note 4)	F1 = 6.4 MHz	40	60	%
Rise Time (Note 4)	F1 = 6.4 MHZ		15	ns
Fall Time (Note 4)	F1 = 6.4 MHZ		10	ns
Instruction Cycle Time RC Oscillator (Note 4)	R = 15 KΩ, V_{CC} = 5V C = 82 pF (÷4 mode)	4	8	μs
Inputs (See *Figure 3*) t_{SETUP}	G Inputs, $V_{CC} \ge 4V$	tc/4+0.4		μs
	SI Input, $V_{CC} \ge 4V$	0.18		μs
	All Others, $V_{CC} \le 4V$	0.95		μS
T_{HOLD}	$V_{CC} > 4V$	0.25		μS
	$4V > V_{CC} \ge 2.4V$	0.75		μs
Output Propagation Delay	V_{OUT} = 1.5V, C_L = 100 pF, R_L = 5K			
T_{PD1}, T_{PD0}	$V_{CC} \ge 4V$		750	ns
T_{PD1}, T_{PD0}	$4V > V_{CC} \ge 2.4V$	2.5		μs
MICROBUS Timing	C_L = 50 pF, V_{CC} = 5V ±5%			
Read Operation *(Figure 4)*				
Chip Select before RD $-t_{CSR}$		80		ns
Chip Select Hold Time for RD $-t_{CSR}$		25		ns
RD Pulse Width $-t_{RR}$		500		ns
Data Delay from RD $-t_{RD}$			470	ns
RD to Data Floating $-t_{DF}$ (Note 4)			310	ns
Write Operation *(Figure 5)*				
Chip Select Stable before WR $-t_{CSW}$		80		ns
Chip Select Hold Time for WR $-t_{WCS}$		25		ns
WR Pulse Width $-t_{WW}$		500		ns
Data Set-up Time for WR $-t_{DW}$		400		ns
Data Hold Time for WR $-t_{WD}$		125		ns
INTR Transition Time from WR $-t_{W1}$			875	ns

Note 1: Supply current is measured after running 2000 cycle times with a square-wave clock on CKI, CKO open, and all other pins pulled to V_{CC} with 20K resistors. See current drain equation, see power dissipation.

Note 2: The HALT mode will stop CKI from oscillating in the RC and crystal configurations. Test conditions: all inputs tied to V_{CC}, L lines in TRI-STATE mode and tied to ground, all outputs low and tied to ground.

Note 3: When forcing HALT, current is only needed for a short time (approx. 200 ns) to flip the HALT flip-flop.

Note 4: This parameter is only sampled and not 100% tested.

Note 5: Voltage change must be less than 0.5 volts in a 1 ms period.

Note 6: SO output sink curent must be limited to keep V_{OL} less than 0.3 V_{CC} when part is running in order to prevent entering test mode.

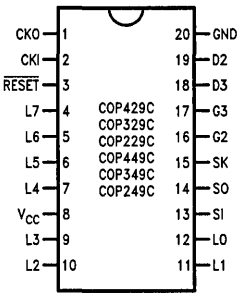
Note 7: If CKI is crystal divide by 16 option, maximum CKI frequency is 3.2 MHz, t_c = 5 μs for $V_{CC} \ge 4V$ and 0.8 MHz, t_c = 20 μs for $2.4V \le V_{CC} < 4V$. If CKI is crystal ÷32 option, maximum CKI frequency is 3.2 MHz, t_c = 10 μs for $V_{CC} \ge 4V$ and 0.8 MHz, t_c = 40 μs for $2.4V \le V_{CC} < 4V$.

Note 8: I/O pin voltage held above V_{CC} or below GND (e.g., +2V/−2V with +200 mA/−200 mA, or +20V/−20V with +20 mA/−20 mA).

Note 9: Human Body Model, 100 pF through 1.5 KΩ.

Connection Diagrams

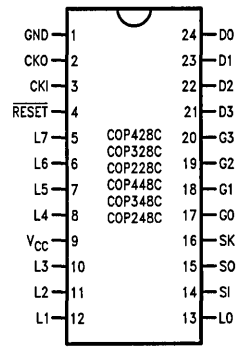
Dual-In-Line Package



TL/DD/9129-2

Top View

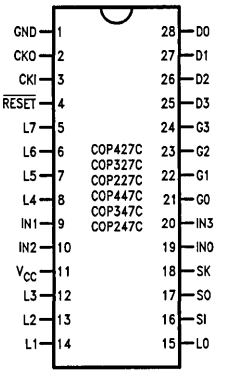
Dual-In-Line Package



TL/DD/9129-3

Top View

Dual-In-Line Package



TL/DD/9129-4

Top View

Pin	Description
L7-L0	8-bit bidirectional port with TRI-STATE
G3-G0	4-bit bidirectional I/O port
D3-D0	4-bit output port
IN3-IN0	4-bit input port (28 pin package only)
SI	Serial input or counter input
SO	Serial or general purpose output
SK	Logic controlled clock output
CKI	Chip oscillator input
CKO	Oscillator output, HALT I/O port or general purpose input
$\overline{\text{RESET}}$	Reset input
V_{CC}	Most positive power supply
GND	Ground

FIGURE 2

Functional Description

The internal architecture is shown in *Figure 1*. Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1", when a bit is reset, it is a logic "0".

For ease of reading only the COP427C/428C/429C and COP447C/448C are referenced; however, all such references apply equally to COP327C/328C/329C/347C/348C and COP227C/228C/229C and COP247C/248C.

PROGRAM MEMORY

Program Memory consists of ROM, 1024 bytes for the COP427C/428C/429C and 2048 bytes for the COP447C/448C. These bytes of ROM may be program instructions, constants or ROM addressing data.

ROM addressing is accomplished by a 11-bit PC register which selects one of the 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 11-bit binary count value.

Three levels of subroutine nesting are implemented by a three level deep stack. Each subroutine call or interrupt pushes the next PC address into the stack. Each return pops the stack back into the PC register.

DATA MEMORY

Data memory consists of a 512-bit RAM for the COP447C/448C, organized as 8 data registers of 16 × 4-bit digits. RAM addressing is implemented by a 7-bit B register whose upper 3 bits (Br) select 1 of 8 data registers and lower 4 bits (Bd) select 1 of 16 4-bit digits in the selected data register.

Data memory consists of a 256-bit RAM for the COP427C/428C/429C, organized as 4 data registers of 16 × 4-bits digits. The B register is 6 bits long. Upper 2 bits (Br) select 1 of 4 data registers and lower 4 bits (Bd) select 1 of 16 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) are usually loaded into or from, or exchanged with, the A register (accumulator), it may also be loaded into or from the Q latches or T counter or loaded from the L ports. RAM addressing may also be performed directly by the LDD and XAD instructions based upon the immediate operand field of these instructions.

The Bd register also serves as a source register for 4-bit data sent directly to the D outputs.

INTERNAL LOGIC

The processor contains its own 4-bit A register (accumulator) which is the source and destination register for most I/O, arithmetic, logic, and data memory access operations. It can also be used to load the Br and Bd portions of the B register, to load and input 4 bits of the 8-bit Q latch or T counter, to input 4 bits of L I/O ports data, to input 4-bit G, or IN ports, and to perform data exchanges with the SIO register.

A 4-bit adder performs the arithmetic and logic functions, storing the results in A. It also outputs a carry bit to the 1-bit C register, most often employed to indicate arithmetic overflow. The C register in conjunction with the XAS instruction and the EN register, also serves to control the SK output.

The 8-bit T counter is a binary up counter which can be loaded to and from M and A using CAMT and CTMA instructions. This counter may be operated in two modes depending on a mask-programmable option: as a timer or as an external event counter. When the T counter overflows, an overflow flag will be set (see SKT and IT instructions below). The T counter is cleared on reset. A functional block diagram of the timer/counter is illustrated in *Figure 10a*.

Note: If desired, the user may mask program the T counter clock source to be permanently external event counter or internally clocked timer. Also, a mask program option is provided to select the T counter to be cleared on chip RESET, or not to be cleared on chip RESET.

The D register provides 4 general-purpose outputs and is used as the destination register for the 4-bit contents of Bd. In the dual clock mode, D0 latch controls the clock selection (see dual oscillator below). On the COP429HC or COP449C (20-pin), D2 may be mask programmed as either dual clock input, IN1 input (for interrupt capability) or as IN2 input (for external event counter). On the COP428C or COP448C

Functional Description (Continued)

(24-pin), D2 may be mask programmed as either IN1 input or IN2 input NOTE: If D2 is optioned as IN1 or IN2, the comparator is not functional.

The G register contents are outputs to a 4-bit general-purpose bidirectional I/O port. G0 may be mask-programmed as an output for Microbus applications. G port inputs can be mask programmed as standard trip or Schmitt trigger inputs.

Both the D-port and G-port outputs may be mask programmed with either LSTTL or high sink I_{OL} capability (See D.C. Characteristics).

The Q register is an internal, latched, 8-bit register, used to hold data loaded to or from M and A, as well as 8-bit data from ROM. Its contents are outputted to the L I/O ports when the L drivers are enabled under program control. With the Microbus option selected, Q can also be loaded with the 8-bit contents of the L I/O ports upon the occurrence of a write strobe from the host CPU.

The 8 L drivers, when enabled, output the contents of latched Q data to the L I/O port. Also, the contents of L may be read directly into A and M. As explained above, the Microbus option allows L I/O port data to be latched into the Q register.

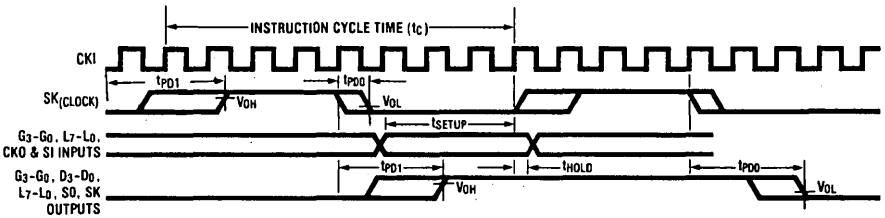
The SIO register functions as a 4-bit serial-in/serial-out shift register for MICROWIRE I/O and COPS peripherals, or as a binary counter (depending on the contents of the EN register). Its contents can be exchanged with A.

The XAS instruction copies C into the SKL latch. In the counter mode, SK is the output of SKL; in the shift register mode, SK outputs SKL ANDed with the clock.

An analog comparator has been provided. IN1 feeds directly to the plus (+) side of the comparator and IN2 is the input of the minus (−) side. The comparator output is latched and the output state can be loaded into the A register by using the INIL instruction. Also, comparator transitions can cause interrupts under program control. Comparator transitions must be at least two instruction cycles wide to be recognized.

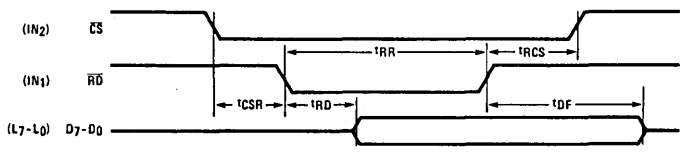
EN is an internal 8-bit register loaded by two instructions. LEI loads the lower four bits (EN3–EN0), and LUEI loads the upper four bits (EN7–EN4). The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register.

0. The least significant bit of the enable register, EN0, selects the SIO register as either a 4-bit shift register or a 4-bit binary counter. With EN0 set, SIO is an asynchronous binary counter, decrementing its value by one upon each low-going pulse ("1" to "0") occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output equals the value of EN3. With EN0 reset, SIO is a serial shift register left shifting 1 bit each instruction cycle time. The data present at SI goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. The SK outputs SKL ANDed with the instruction cycle clock.
1. With EN1 set, external interrupts or comparator transition are enabled. Immediately following an interrupt or comparator transition interrupt, EN1 is reset to disable further external or comparator transition interrupts.



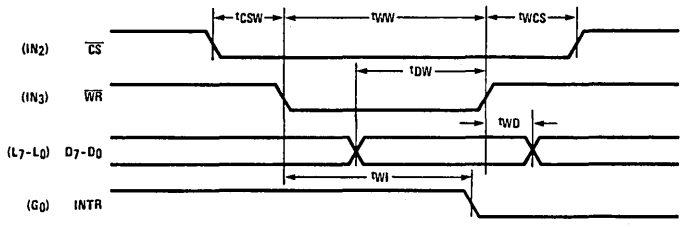
TL/DD/9129-5

FIGURE 3. Input/Output Timing Diagrams (divide by 8 mode)



TL/DD/9129-6

FIGURE 4. Microbus Read Operation Timing



TL/DD/9129-7

FIGURE 5. Microbus Write Operation Timing

Functional Description (Continued)

2. With EN2 set, the L drivers are enabled to output the data in the Q latch to the L I/O port. Resetting EN2 disables the L drivers, placing the L I/O port in a high-impedance input state.
3. EN3, in conjunction with EN0, affects the SO output. With EN0 set (binary counter option selected) SO will output the value loaded into EN3. With EN0 reset (serial shift register option selected), setting EN3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN3 with the serial shift register option selected disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains set to "0".
4. As stated above, the T counter clock source can be selected under program control. This is accomplished with EN4. With EN4 set, the T counter operates as an external event counter, clocked by IN2. When EN4 is reset, the T counter operates as a timer, being clocked at the instruction rate with a two bit prescaler.

NOTE: The IT instruction is not allowed when EN4 is set.

5. EN5 controls the access of the comparator latch. When EN5 is set, the comparator is enabled, and the state of the comparator output can be read by the INIL instruction. Also, with EN5 set, an ININ instruction will read 'ones' on the IN1 and IN2 inputs. With EN5 set, the external interrupt source is on comparator transitions. When EN5 is reset, the comparator is disabled (to eliminate any DC current), an INIL will load a 'zero' into bit one of A, and the external interrupt source is on IN1. CAUTION: If the programmer wishes to detect a comparator difference by using interrupts then EN5 should be set before EN1 to avoid false interrupts. Also, EN5 should be reset before going into halt mode to avoid approximately 5 microamps of DC current due to the comparator being enabled.
6. With EN6 set, a T counter overflow will cause an internal interrupt. When EN6 is reset, T counter overflows will not cause an interrupt. If EN6 is set, a timer overflow interrupt will cause EN6 to be reset. The T counter source may also be mask programmed.
7. EN7 is the T counter start/stop enable bit. When EN7 is set, the T counter is disabled from counting. When EN7 is reset, the T counter counts up in binary.

INTERRUPTS

The following features are associated with interrupt procedure and protocol and must be considered by the programmer when utilizing interrupts.

A. The interrupt, once recognized as explained below, pushes the next sequential program counter address (PC+1) onto the stack. Any previous contents at the bottom of the stack are lost. The program counter is set to hex address 0FF (the last word of page 3) and both EN1 and EN6 are reset.

B. An interrupt will be recognized only on the following conditions:
   1. EN1 has been set in anticipation of external interrupts or EN6 has been set for an internal timer overflow interrupt.
   2. With EN1 set and EN5 reset, a low going pulse ('1' to '0') at least two instruction cycles wide on the IN1 input will cause an interrupt.
   3. With EN1 set and EN5 set, a transition on the comparator output that is at least two instruction cycles wide will cause an interrupt.
   4. With EN6 set, a T counter overflow will cause an interrupt.
   5. All currently executing instructions must be completed.
   6. All successive transfer of control instructions and successive LBIs must be completed (e.g., if the main program is executing a JP instruction, the interrupt will not be acknowledged until the second JP instruction has been executed).

C. Upon acknowledgement of an interrupt, the skip logic status is saved and later restored upon popping of the stack. For example, if an interrupt occurs during the execution of ASC (Add with Carry, Skip on Carry) instruction which results in carry, the skip logic status is saved and program control is transferred to the interrupt servicing routine at hex address 0FF. At the end of the interrupt routine, a RET instruction is executed to pop the stack and return program control to the instruction following the original ASC. At this time, the skip logic is enabled and skips this instruction because of the previous ASC carry. Subroutines should not be nested within the interrupt service routine, since their popping of the stack will enable any previously saved main program skips, interfering with the orderly execution of the interrupt routine.

D. The instruction at hex address 0FF must be a NOP.

E. An LEI or LUEI instruction may be put immediately before the RET instruction to re-enable interrupts. NOTE: EN1 and EN6 may be set together at the same point in time to await both internal and external interrupts. If both are set and an interrupt occurs, then the SKT latch must be checked to see if it was the T counter that overflowed. If both interrupts occur on the same cycle, there is no priority, other than the T-counter will be serviced first if SKT is checked in the interrupt service routine.

MICROBUS INTERFACE

The COP447C/427C has an option which allows it to be used as a peripheral microprocessor device, inputting and outputting data from and to a host microprocessor (μP). IN1, IN2 and IN3 general purpose inputs become Microbus compatible read-strobe, chip-select, and write-strobe lines, respectively. IN1 becomes $\overline{RD}$ — a logic "0" on this input will cause Q latch data to be enabled to the L ports for input to the μP. IN2 becomes $\overline{CS}$ — a logic "0" on this line selects the COP447C/427C as the μP peripheral device by enabling the operation of the $\overline{RD}$ and $\overline{WR}$ lines and allows for the selection of one of several peripheral components.

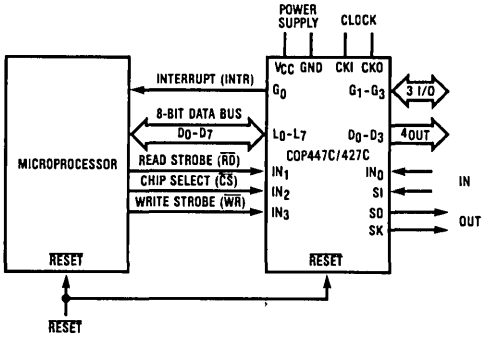
IN3 becomes $\overline{WR}$ — a logic "0" on this line will write bus data from the L ports to the Q latches for input to the COP447C/427C. G0 becomes INTR a "ready" output, reset by a write pulse from the μP on the $\overline{WR}$ line, providing the "handshaking" capability necessary for asynchronous data transfer between the host CPU and the COP447C/427C.

This option has been designed for compatibility with National's Microbus — a standard interconnect system for 8-bit parallel data transfer between MOS/LSI CPUs and interfacing devices. (See Microbus National Publication.) The functioning and timing relationships between the signal lines affected by this option are as specified for the Microbus interface, and are given in the AC electrical characteristics and shown in the timing diagrams (*Figures 4* and *5*). Connection of the COP447C/427C to the Microbus is shown in *Figure 6*.

Functional Description (Continued)

TABLE I. Enable Register Modes
Bits EN5, EN4 and EN1

EN1	EN5	EN4	IN1	IN2	INIL Bit 1	Interrupts
0	0	0	STD Input	STD Input	0	None
0	0	1	STD Input	EXT CNTR	0	None
0	1	0	+ of COMPAR	− of COMPAR	CMP	None
1	0	0	EXT INTERRUPT	STD Input	0	EXT IN1
1	1	0	+ of COMPAR	− of COMPAR	CMP	+ or − CMP Edge
1	0	1	EXT INTERRUPT	EXT CNTR	0	EXT IN1
0	1	1	Not Allowed			
1	1	1	Not Allowed			



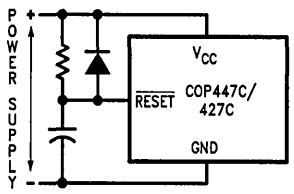
TL/DD/9129-8

FIGURE 6. Microbus Option Interconnect

INITIALIZATION

The internal reset logic will initialize the device upon power-up if the power supply rise time is less than 1 ms and if the operating frequency at CKI is greater than 32 kHz, otherwise the external RC network shown in *Figure 7* must be connected to the $\overline{\text{RESET}}$ pin (the conditions in *Figure 7* must be met). The $\overline{\text{RESET}}$ pin is configured as a Schmitt trigger input. If not used, it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the $\overline{\text{RESET}}$ input, providing it stays low for at least three instruction cycle times.

Note: If CKI clock is less than 32 kHz, the internal reset logic (option #29 = 1) MUST be disabled and the external RC circuit must be used.



TL/DD/9129-9

RC ≥ 5X Power Supply Rise Time
and RC ≥ 100X CKI Period.

FIGURE 7. Power-Up Circuit

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN, IL, T and G registers are cleared. The SKL latch is set, thus enabling SK as a clock output. Data Memory (RAM) is not cleared upon initialization. The first instruction at address 0 must be a CLRA (clear A register).

TIMER

There are two modes selected by mask option:

a. Time-base counter. In this mode, the instruction cycle frequency generated from CKI passes through a 2-bit divide-by-4 prescaler. The output of this prescaler increments the 8-bit T counter thus providing a 10-bit timer. The prescaler is cleared during execution of a CAMT instruction and on reset.

For example, using a 4 MHz crystal with a divide-by-16 option, the instruction cycle frequency of 250 kHz increments the 10-bit timer every 4 μs. By presetting the counter and detecting overflow, accurate timeouts between 16 μs (4 counts) and 4.096 ms (1024 counts) are possible. Longer timeouts can be achieved by accumulating, under software control, multiple overflows.

b. External event counter. In this mode, a low-going pulse ("1" to "0") at least 2 instruction cycles wide on the IN2 input will increment the 8-bit T counter.

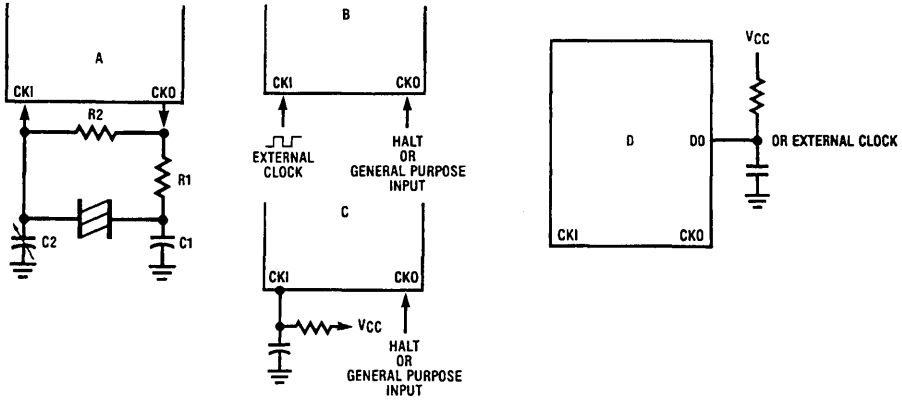
Note: The IT instruction is not allowed in this mode.

HALT MODE

The COP447C/448C/449C/427C/428C/429C is a FULLY STATIC circuit; therefore, the user may stop the system oscillator at any time to halt the chip. The chip may also be halted by the HALT instruction or by forcing CKO high when it is mask-programmed as an HALT I/O port. Once in the HALT mode, the internal circuitry does not receive any clock signal and is therefore frozen in the exact state it was in when halted. All information is retained until continuing. The chip may be awakened by one of two different methods:

- Continue function: by forcing CKO low, if it mask-programmed as an HALT I/O port, the system clock is re-enabled and the circuit continues to operate from the point where it was stopped.
- Wake up from HALT: Any combination of the L port lines can be mask programmed as wake up pins (this is for crystal oscillator or CKO general purpose input options, where CKO cannot be the HALT I/O port). Before going into HALT mode, user must guarantee that the L lines are held high, i.e., write ones to the Q latch and enable the L line drivers, or leave the L port in TRI-STATE mode and provide external loads to hold logic 'ones'. Then, after going into halt mode, the internal clocks can 'wake up' by forcing any one of the mask programmed L lines low. (See characteristic curves to determine how much current is needed to override L line I_{OH}.) NOTE: L-port output source current can be different... see options.
- Restart: by forcing the $\overline{\text{RESET}}$ pin low (see Initialization).

Functional Description (Continued)



TL/DD/9129-10

Crystal or Resonator

Crystal Value	Component Values			
	R1	R2	C1(pF)	C2(pF)
32 kHz	220k	20M	30	6-36
455 kHz	5k	10M	80	40
2.096 MHz	2k	1M	30	6-36
4.0 MHz	1k	1M	30	6-36

RC Controlled Oscillator

R	C	Cycle Time	V_{CC}
15k	82 pF	4-9 μs	≥4.5V
30k	82 pF	8-16 μs	≥4.5V
60k	100 pF	16-32 μs	2.4-4.5V

Note: 15k≤R≤150k
50 pF≤C≤150 pF

FIGURE 8. Oscillator Component Values

The HALT mode is the minimum power dissipation state.

Note: If the user has selected dual-clock with D0 as external oscillator (option 30=2) AND the COP447C/427C is running with the D0 clock, the HALT mode — either hardware or software — will NOT be entered. Thus, the user should switch to the CKI clock to HALT. Alternatively, the user may stop the D0 clock to minimize power.

CKO PIN OPTIONS

a. Two-pin oscillator — (Crystal). See *Figure 9A*.

In a crystal controlled oscillator system, CKO is used as an output to the crystal network. The HALT mode may be entered by program control (HALT instruction) which forces CKO high, thus inhibiting the crystal network. The circuit can be awakened only by forcing the $\overline{\text{RESET}}$ pin to a logic "0" (restart).

b. One-pin oscillator — (RC or external). See *Figure 9B*.

If a one-pin oscillator system is chosen, two options are available for CKO:

- CKO can be selected as the HALT I/O port. In that case, it is an I/O flip-flop which is an indicator of the HALT status. An external signal can over-ride this pin to start and stop the chip. By forcing a high level to CKO, the chip will stop as soon as CKI is high and CKO output will stay high to keep the chip stopped if the external driver returns to high impedance state. By forcing a low level to CKO, the chip will continue and CKO will stay low.
- As another option, CKO can be a general purpose input, read into bit 2 of A (accumulator) upon execution of an INIL instruction.

OSCILLATOR OPTIONS

There are four basic clock oscillator configurations available as shown by *Figure 8*.

a. Crystal Controlled Oscillator. CKI and CKO are connected to an external crystal. The instruction cycle time equals the crystal frequency optionally divided by 4, 8 or 16.

b. External Oscillator. The external frequency is optionally divided by 4, 8 or 16 to give the instruction cycle time. CKO is the HALT I/O port or a general purpose input.

c. RC Controlled Oscillator. CKI is configured as a single pin RC controlled Schmitt trigger oscillator. The instruction cycle equals the oscillation frequency divided by 4. CKO is the HALT I/O port or a general purpose input.

d. Dual oscillator. By selecting the dual clock option, pin D0 is now a single pin oscillator input. Two configurations are available: RC controlled Schmitt trigger oscillator or external oscillator.

The user may software select between the D0 oscillator (in that case, the instruction cycle time equals the D0 oscillation frequency divided by 4) by setting the D0 latch high or the CKI (CKO) oscillator by resetting D0 latch low. Note that even in dual clock mode, the counter, if mask-programmed as a time-base counter, is always connected to the CKI oscillator. If D2 is selected as the D0 dual clock function for a 20 pin part, the D0 latch controls the clock selection.

For example, the user may connect up to a 2 MHz RC circuit to D0 for faster processing and a 32 kHz watch crystal to CKI and CKO for minimum current drain and time keeping.

Note: CTMA instruction is not allowed when chip is running from D0 clock.

Functional Description (Continued)

Figures 10A and *10B* show the clock and timer diagrams with and without Dual clock.

MALFUNCTION DETECTOR

With a mask option, the user can invoke circuitry which will detect loss of program control. When selected the G_3 pin is configured as a RESET pin. A high level, VIHMD, on G_3 will reset the chip. The malfunction detector uses an external RC network connected to the G_3 pin which pulls G_3 toward V_{CC}. Under program control, using the OMG or OGI instruction, the G_3 driver will discharge the external RC network. Failure to write a zero to G_3, loss of program control, will cause the chip to reset itself when the external RC network charges G_3 to a Logic 1, VIHMD. In this mode the G_3 latch is configured such that when a zero is written to G_3 the pin will force a zero until the external RC network is discharged to the VOLMD level, then continue to force a zero for one instruction cycle, then the latch will set and the pad will begin to charge toward V_{CC}. This latch configuration allows the use of a large external capacitor and eliminates the need to write a zero followed by a one to the G_3 pin. The G_3 driver may optionally be configured with an on-board pull-up enabled so that the only external component required is a capacitor. Sink and source current for the malfunction detector is selected with the G_3 driver option, option 24. See DC specification and characteristic curves to determine charge and discharge times. The G_3 pad will write a zero when HALT state is entered, the G_3 latch is set when a continue occurs. CAUTION must be used when entering IDLE state. If the IT instruction is executed when the G_3 pad is writing a zero, G_3 latch at a zero, the malfunction detector will be enabled but the G_3 pad will continue to write a zero until chip clocking resumes. However if the IT instruction is executed when the G_3 latch is at one the malfunction detector is active and will reset the chip out of IDLE state when the G_3 pad is charged up to the VIHMD level.

DC Specs	Min	Max
VIHMD	0.45 V_{CC}	0.80 V_{CC}
VOLMD		0.50 V_{CC}
VIHMD-VOLMD	0.20 V_{CC}	
External R STD Sink HIGH Sink	 10 KΩ 5 KΩ	

EXAMPLES

V_{CC} = 5.0V, temp = 25°C, VIHMD = 2.25V, VOLMD = 1.0V, charge to 2.0V

Time	Driver	External R	External C	Discharge Time	Charge
sec.	HIGH Sink	20 KΩ	1.0 μF	40 μs	5.7 m
sec.	HIGH Sink	560 KΩ	0.22 μF	10 μs	36 m
sec.	HIGH Sink	47 KΩ	2.2 μF	95 μs	30 m
sec.	STD Sink	200 KΩ	0.1 μF	20 μs	5.7 m

COP448C AND COP428C 24-PIN PACKAGE OPTION

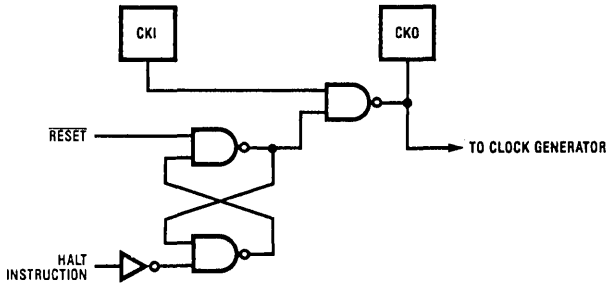
If the COP447C/427C is bonded in a 24-pin package, it becomes the COP448C/428C, illustrated in *Figure 2*, Connection diagrams. Note that the COP448C/428C does not contain the four general purpose IN inputs (IN3–IN0). Use of this option precludes, of course, use of the IN options, interrupt feature, external event counter feature, and the Microbus option which uses IN1–IN3. However, a mask programmable option exists to select D2 pin to be either IN1 or IN2 for all functions on those pins except Comparator and MICROBUS. All other options are available for the COP448C/428C.

COP429C AND COP449C 20-PIN PACKAGE OPTION

If the COP428C/COP447C is bonded as a 20-pin device it becomes the COP429C/COP449C. Note that the COP429C/COP449C contains all the COP427C/COP447C pins except D_0, D_1, G_0, and G_1. However, D2 can be mask optioned as explained above to be either IN1 or IN2, OR it can be optioned to be the dual clock function of D0.

Note: If user selects the 24-pin or 20-pin package, options 9, 10, 19 and 20 must be selected as a "0" (load to V_{CC} on the IN inputs). See option list.

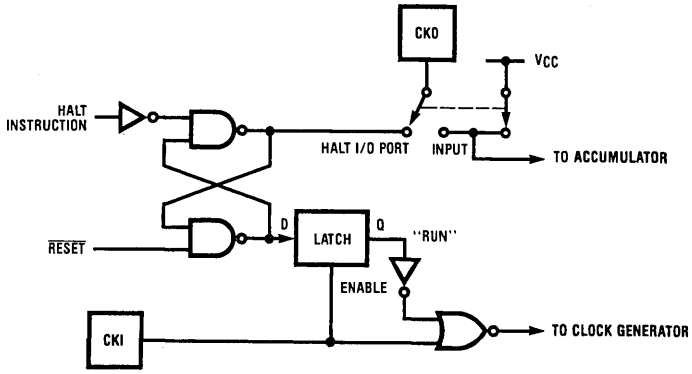
Block Diagrams



TL/DD/9129-11

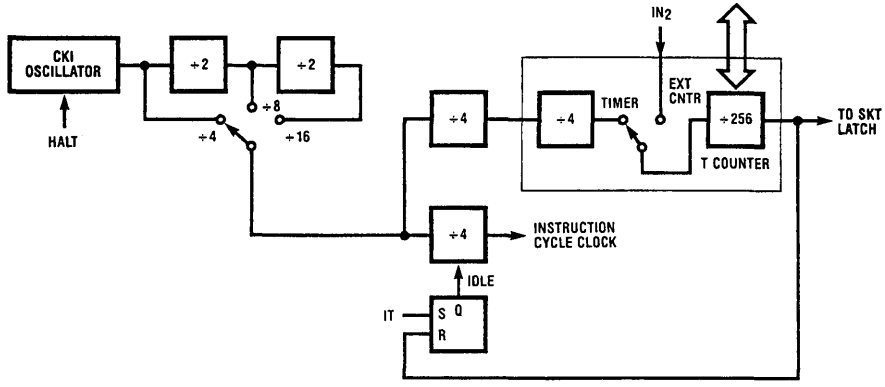
FIGURE 9A. Halt Mode — Two-Pin Oscillator

Block Diagrams (Continued)



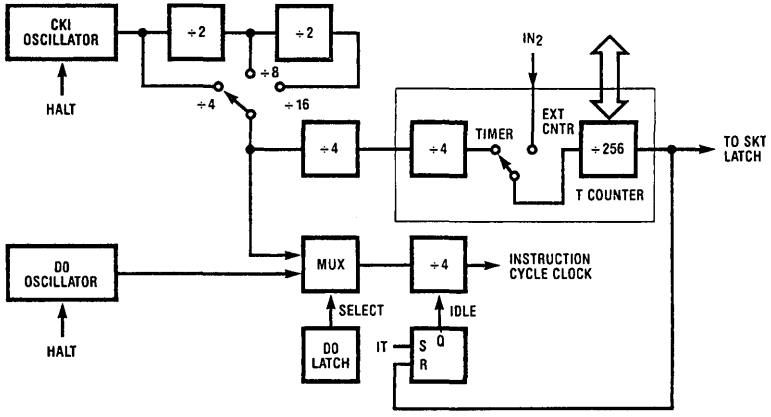
TL/DD/9129–12

FIGURE 9B. Halt Mode — One-Pin Oscillator



TL/DD/9129–13

FIGURE 10A. Clock and Timer without Dual-Clock



TL/DD/9129–14

FIGURE 10B. Clock and Timer with Dual-Clock

Instruction Set

Table II is a symbol table providing internal architecture, instruction operand and operation symbols used in the instruction set table.

TABLE II. Instruction Set Table Symbols

Symbol	Definition
Internal Architecture Symbols	
A	4-bit accumulator
B	7-bit RAM address register (6-bit for COP427C)
Br	Upper 3 bits of B (register address) (2-bit for COP427C)
Bd	Lower 4 bits of B (digit address)
C	1-bit carry register
D	4-bit data output port
EN	8-bit enable register
G	4-bit general purpose I/O port
IL	two 1-bit (IN0 and IN3) latches
IN	4-bit input port
L	8-bit TRI-STATE I/O port
M	4-bit contents of RAM addressed by B
PC	11-bit ROM address program counter
Q	8-bit latch for L port
SA,SB,SC	11-bit 3-level subroutine stack
SIO	4-bit shift register and counter
SK	Logic-controlled clock output
SKL	1-bit latch for SK output
T	8-bit timer
Cmp	1-bit latch for comparator output

Table III provides the mnemonic, operand, machine code data flow, skip conditions and description of each instruction.

Instruction Operand Symbols	
d	4-bit operand field, 0–15 binary (RAM digit select)
r	3(2)-bit operand field, 0–7(3) binary (RAM register select)
a	11-bit operand field, 0–2047 (1023)
y	4-bit operand field, 0–15 (immediate data)
RAM(x)	RAM addressed by variable x
ROM(x)	ROM addressed by variable x

Operational Symbols	
+	Plus
−	Minus
→	Replaces
⟷	Is exchanged with
=	Is equal to
$\overline{A}$	One's complement of A
⊕	Exclusive-or
:	Range of values

TABLE III. COP427C/447C Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	0011 | 0000	A+C+RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	0011 | 0001	A+RAM(B) → A	None	Add RAM to A
ADT		4A	0100 | 1010	$A+10_{10}$ → A	None	Add Ten to A
AISC	y	5−	0101 | y	A+y → A	Carry	Add Immediate. Skip on Carry (y ≠ 0)
CASC		10	0001 | 0000	$\overline{A}$+RAM(B)+C → A Carry → C	Carry	Complement and Add with Carry, Skip on Carry
CLRA		00	0000 | 0000	0 → A	None	Clear A
COMP		40	0100 | 0000	$\overline{A}$ → A	None	Ones complement of A to A
NOP		44	0100 | 0100	None	None	No Operation
RC		32	0011 | 0010	"0" → C	None	Reset C
SC		22	0010 | 0010	"1" → C	None	Set C
XOR		02	0000 | 0010	A⊕RAM(B) → A	None	Exclusive-OR RAM with A

Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
TRANSFER CONTROL INSTRUCTIONS						
JID		FF	1111 | 1111	ROM ($PC_{10:8}$ A,M) → $PC_{7:0}$	None	Jump Indirect (Notes 1, 3)
JMP	a	6–	0110 | 0 | $a_{10:8}$	a → PC	None	Jump
		––	$a_{7:0}$			
JP	a	––	1 | $a_{6:0}$ (pages 2,3 only)	a → $PC_{6:0}$	None	Jump within Page (Note 4)
			or			
		––	11 | $a_{5:0}$ (all other pages)	a → $PC_{5:0}$		
JSRP	a	––	10 | $a_{5:0}$	PC+1 → SA → SB → SC 00010 → $PC_{10:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 5)
JSR	a	6–	0110 | 1 | $a_{10:8}$	PC+1 → SA → SB → SC	None	Jump to Subroutine
		––	$a_{7:0}$	a → PC		
RET		48	0100 | 1000	SC → SB → SA → PC	None	Return from Subroutine
RETSK		49	0100 | 1001	SC → SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
HALT		33	0011 | 0011		None	HALT Processor
		38	0011 | 1000			
IT		33	0011 | 0011			IDLE till Timer
		39	0011 | 1001		None	Overflows then Continues
MEMORY REFERENCE INSTRUCTIONS						
CAMT		33	0011 | 0011	A → $T_{7:4}$		
		3F	0011 | 1111	RAM(B) → $T_{3:0}$	None	Copy A, RAM to T
CTMA		33	0011 | 0011	$T_{7:4}$ → RAM(B)		
		2F	0010 | 1111	$T_{3:0}$ → A	None	Copy T to RAM, A (Note 9)
CAMQ		33	0011 | 0011	A → $Q_{7:4}$	None	Copy A, RAM to Q
		3C	0011 | 1100	RAM(B) → $Q_{3:0}$		
CQMA		33	0011 | 0011	$Q_{7:4}$ → RAM(B)	None	Copy Q to RAM, A
		2C	0010 | 1100	$Q_{3:0}$ → A		
LD	r	–5	00 | r | 0101 (r=0:3)	RAM(B) → A Br⊕r → Br	None	Load RAM into A, Exclusive-OR Br with r
LDD	r,d	23	0010 | 0011	RAM(r,d) → A	None	Load A with RAM pointed to directly by r,d
		––	0 | r | d			
LQID		BF	1011 | 1111	ROM($PC_{10:8}$,A,M) → Q SB → SC	None	Load Q Indirect (Note 3)
LID		33	0011 | 0011	ROM($PC_{10:18}$, A,M)7:4 → RAM(B)		Load M,A Indirect (Note 3)
		19	0001 | 1001	ROM($PC_{10:18}$,A,A1)3:0 → A		
RMB	0	4C	0100 | 1100	0 → $RAM(B)_0$	None	Reset RAM Bit
	1	45	0100 | 0101	0 → $RAM(B)_1$		
	2	42	0100 | 0010	0 → $RAM(B)_2$		
	3	43	0100 | 0011	0 → $RAM(B)_3$		

Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
MEMORY REFERENCE INSTRUCTIONS (Continued)						
SMB	0	4D	0100 | 1101	$1 \rightarrow RAM(B)_0$	None	Set RAM Bit
	1	47	0100 | 0111	$1 \rightarrow RAM(B)_1$		
	2	46	0100 | 0110	$1 \rightarrow RAM(B)_2$		
	3	4B	0100 | 1011	$1 \rightarrow RAM(B)_3$		
DRS		33	0011 | 0011	(RAM(B)-1)1 → RAM(B)		Decrement RAM, Skip if less than 0
		IE	0001 | 1110	Skip if RAM(B)<0		
AIBD	y	33	0011 | 0011			
		4Y	0100 | y	(Bd + y) → Bd	None	Add immediate to Bd
STII	y	7–	0111 | y	y → RAM(B) Bd + 1 → Bd	None	Store Memory Immediate 1 and Increment Bd
X	r	–6	00 | r | 0110 (r=0:3)	RAM(B) ⟷ A Br ⊕ r → Br	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	r,d	23	0010 | 0011	RAM(r,d) ⟷ A	None	Exchange A with RAM Pointed to Directly by r,d
		– –	1 | r | d			
XDS	r	–7	00 | r | 0111 (r=0:3)	RAM(B) ⟷ A Bd–1 → Bd Br ⊕ r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd. Exclusive-OR Br with r
XIS	r	–4	00 | r | 0100 (r=0:3)	RAM(B) ⟷ A Bd+1 → Bd Br ⊕ r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd, Exclusive-OR Br with r
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	0101 | 0000	A → Bd	None	Copy A to Bd
CBA		4E	0100 | 1110	Bd → A	None	Copy Bd to A
LBI	r,d	– –	00 | r | (d–1) (r=0:3: d=0,9:15) or	r,d → B	Skip until not a LBI	Load B Immediate with r,d (Note 6)
		33	0011 | 0011			
		– –	1 | r | d (any r, any d)			
LEI	y	33	0011 | 0011	y → EN	None	Load EN Immediate (Note 7)
		6–	0110 | y			
LUEI	y	33	0011 | 0011	$y \rightarrow EN_{7:4}$	None	Load Upper EN Immediate (Note 7)
		7y	0111 | y			
XABR		12	0001 | 0010	A ⟷ Br	None	Exchange A with Br (Note 8)
TEST INSTRUCTIONS						
SKC		20	0010 | 0000		C="1"	Skip if C is True
SKE		21	0010 | 0001		A=RAM(B)	Skip if A Equals RAM
SKGZ		33	0011 | 0011		$G_{3:0}=0$	Skip if G is Zero (all 4 bits)
		21	0010 | 0001			
SKGBZ		33	0011 | 0011	1st byte		Skip if G Bit is Zero
	0	01	0000 | 0001	2nd byte	$G_0=0$	
	1	11	0001 | 0001		$G_1=0$	
	2	03	0000 | 0011		$G_2=0$	
	3	13	0001 | 0011		$G_3=0$	

Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
TEST INSTRUCTIONS (Continued)						
SKMBZ	0	01	0000 | 0001		$RAM(B)_0 = 0$	Skip if RAM Bit is Zero
	1	11	0001 | 0001		$RAM(B)_1 = 0$	
	2	03	0000 | 0011		$RAM(B)_2 = 0$	
	3	13	0001 | 0011		$RAM(B)_3 = 0$	
SKT		41	0100 | 0001		A time-base counter carry has occurred since last test	Skip on Timer (Note 3)
INPUT/OUTPUT INSTRUCTIONS						
ING		33	0011 | 0011	G → A	None	Input G Ports to A
		2A	0010 | 1010			
ININ		33	0011 | 0011	IN → A	None	Input IN Inputs to A (Note 2)
		28	0010 | 1000			
INIL		33	0011 | 0011	IL_3, CKO, CMP, IL_0 → A	None	Input IL Latches to A (Note 3)
		29	0010 | 1001			
INL		33	0011 | 0011	$L_{7:4}$ → RAM(B)	None	Input L Ports to RAM,A
		2E	0010 | 1110	$L_{3:0}$ → A		
OBD		33	0011 | 0011	Bd → D	None	Output Bd to D Outputs
		3E	0011 | 1110			
OGI	y	33	0011 | 0011	y → G	None	Output to G Ports Immediate
		5–	0101 | y			
OMG		33	0011 | 0011	RAM(B) → G	None	Output RAM to G Ports
		3A	0011 | 1010			
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO (Note 3)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicitly defined (e.g., Br and Bd are explicitly defined). Bits are numbered 0 to N where 0 signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit A register.

Note 2: The ININ instruction is not available on the 24-pin packages since these devices do not contain the IN inputs.

Note 3: For additional information on the operation of the XAS, JID, LQID, LID, INIL, and SKT instructions, see below.

Note 4: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 5: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 6: LBI is a single-byte instruction if d = 0, 9, 10, 11, 12, 13, 14, or 15. The machine code for the lower 4 bits equals the binary value of the "d" data *minus 1*, e.g., to load the lower four bits of B(Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Note 7: Machine code for operand field y for LEI and LUEI instruction should equal the binary value to be latched into EN, where a "1" or "0" in each bit of EN corresponds with the selection or deselection of a particular function associated with each bit. (See Functional Description, EN Register.)

Note 8: For 2K ROM devices, A ⟷ Br (0 → A3). For 1K ROM devices, A ⟷ Br (0,0 → A3, A2).

Note 9: Do not use CTMA instruction when dual-clock option is selected and part is running from D_0 clocks.

Description of Selected Instructions

XAS INSTRUCTION

XAS (Exchange A with SIO) copies C to the SKL latch and exchanges the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. If SIO is selected as a shift register, an XAS instruction can be performed once every 4 instruction cycles to effect a continuous data stream.

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 11-bit word PC10:PC8,A,M. LQID can be used for table lookup or code conversion such as BCD to seven-segment. The LQID instruction "pushes" the stack (PC+1 → SA → SB → SC) and replaces the least significant 8 bits of the PC as follows: A → PC(7:4), RAM(B) → PC(3:0), leaving PC(10), PC(9) and PC(8) unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SC → SB → SA → PC), restoring the saved value of PC to continue sequential program execution. Since LQID pushes SB → SC, the previous contents of SC are lost.

Note: LQID uses 2 instruction cycles if executed, one if skipped.

LID INSTRUCTION

LID (Load A,M Indirect) has the same operation as LQID except that the ROM contents pointed to by PC(10:8), A,M are loaded back into the M and A rather than the Q latch. Therefore, the data in A and M that is used as the ROM address will be overwritten by the ROM data that is fetched. However, using LID instruction will save the contents of the Q latch which may be used as L port output data and not lose integrity. NOTE: LID is a two-byte instruction and therefore takes three instruction cycle times if executed, two if skipped.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower 8 bits of the ROM address register PC with the contents of ROM addressed by the 11-bit word, PC10:8,A,M. PC10,PC9 and PC8 are not affected by JID.

Note: JID uses 2 instruction cycles if executed, one if skipped.

SKT INSTRUCTION

The SKT (Skip On Timer) instruction tests the state of the T counter overflow latch (see internal logic, above), executing the next program instruction if the latch is not set. If the latch has been set since the previous test, the next program instruction is skipped and the latch is reset. The features associated with this instruction allow the processor to generate its own time-base for real-time processing, rather than relying on an external input signal.

IT INSTRUCTION

The IT (idle till timer) instruction halts the processor and puts it in an idle state until the time-base counter overflows. This idle state reduces current drain since all logic (except the oscillator and time base counter) is stopped. IT instruction is not allowed if the T counter is mask-programmed as an external event counter (option #31 = 1). IT instruction is not allowed if the T counter is mask programmed (option #31 = 2) or software selected (EN4 = 1) as an external event counter.

INIL INSTRUCTION

INIL (Input IL Latches to A) inputs two latches, IL3 and IL0 into A3 and A0 respectively, CKO into A2 (if CKO is mask programmed as a general purpose input) or a logic "1" into A2, and the comparator output latch into A1 if EN5 is set. If EN5 is reset, then a logic "0" is loaded into A1. The IL3 and IL0 latches are set if a low-going pulse ("1" to "0") has occurred on the IN3 and IN0 inputs since the last INIL instruction, provided the input pulse stays low for at least two instruction cycles. Execution of an INIL resets these latches after loading them into A3 and A0 to allow them to respond to subsequent low-going pulses on the IN3 and IN0 lines. IL latches are cleared on reset, and are not available on the COP447C/448C/427C/428C.

INSTRUCTION SET NOTES

a. The first word of a program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, they are still fetched from the program memory. Thus program paths take the same number of cycles whether instructions are skipped or executed except for JID, and LQID.

c. The ROM is organized into pages of 64 words each. The Program Counter is a 11-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID, LID or LQID is the last word of a page, it operates as if it were in the next page. For example: a JP located in the last word of a page will jump to a location in the next page. Also, a JID, LID or LQID located in the last word of every fourth page (i.e. hex address 0FF, 1FF, 2FF, 3FF, 4FF, etc.) will access data in the next group of four pages.

Note: The COP427C/COP428C/COP429C needs only 10 bits to address its ROM. Therefore, the eleventh bit (P10) is ignored.

Power Dissipation

The lowest power drain is when the clock is stopped. As the frequency increases so does current. Current is also lower at lower operating voltages. Therefore, the user should run at the lowest speed and voltage that his application will allow. The user should take care that all pins swing to full supply levels to insure that outputs are not loaded down and that inputs are not at some intermediate level which may draw current. Any input with a slow rise or fall time will draw additional current. A crystal or resonator generated clock input will draw additional current. For example, a 500 kHz crystal input will typically draw 100 μA more than a square wave input. An R/C oscillator will draw even more current since the input is a slow rising signal.

If using an external squarewave oscillator, the following equation can be used to calculate operating current drain.

$I_{CO} = I_Q + V \times 40 \times Fi + V \times 1400 \times Fi/Dv$

where I_{CO} = chip operating current drain in microamps

I_Q quiescent leakage current (from curve)

Fi CKI frequency in MegaHertz

V chip V_{CC} in volts

Dv divide by option selected

Power Dissipation (Continued)

For example at 5 volts V_{CC} and 400 kHz (divide by 4)

$I_{CO} = 20 + 5 \times 40 \times 0.4 + 5 \times 1400 \times 0.4/4$

$I_{CO} = 20 + 80 + 700 = 800\ \mu A$

At 2.4 volts V_{CC} and 30 kHz (divide by 4)

$I_{CO} = 6 + 2.4 \times 40 \times 0.03 + 2.4 \times 1400 \times 0.03/4$

$I_{CO} = 6 + 2.88 + 25.2 = 34.08\ \mu A$

If an IT instruction is executed, the chip goes into the IDLE mode until the timer overflows. In IDLE mode, the current drain can be calculated from the following equation:

$I_{ci} = I_Q + V \times 40 \times Fi$

For example, at 5 volts V_{CC} and 400 kHz

$I_{ci} = 20 + 5 \times 40 \times 0.4 = 100\ \mu A$

The total average current will then be the weighted average of the operating current and the idle current:

$$I_{ta} = I_{CO} \times \frac{To}{To+Ti} + I_{ci} \times \frac{Ti}{To+Ti}$$

where: I_{ta} = total average current
I_{CO} = operating current
I_{ci} = idle current
To = operating time
Ti = idle time

I/O OPTIONS

Outputs have the following optional configurations, illustrated in *Figure 11*:

A. Push-Pull—A CMOS push-pull buffer with an N-Channel device to ground in conjunction with a P-Channel device to V_{CC}, compatible with CMOS and LSTTL.

B. Open Drain—An N-Channel device to ground only, allowing external pull up as required by the user's application. Open drain outputs can withstand voltage levels as high as 14V if the N-Channel device is "off."

C. TRI-STATE L Output—A CMOS output buffer similar to (A), which may be disabled by program control.

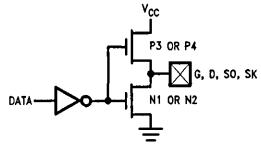
D. Open Drain TRI-STATE L Output—Same as (C) except without the P-Channel pull-up.

All inputs except CKI have the following options:

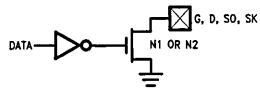
E. Input with on chip load device to V_{CC}.

F. Hi-Z input which must be driven by the user's logic. As on Open Drain outputs, Hi-Z inputs can withstand up to 14V.

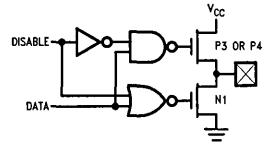
G. G-line inputs may be programmed as Schmitt trigger inputs (hysteresis) or as regular inverter inputs (no hysteresis).



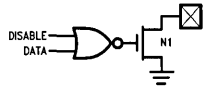
a. Push-Pull Output



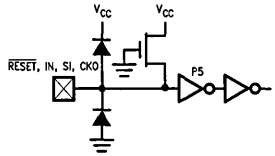
b. Open-Drain Output



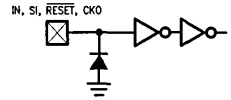
c. TRI-STATE "L" Output



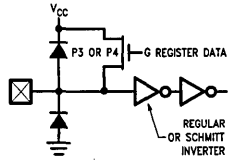
d. Open Drain TRI-STATE "L" Output



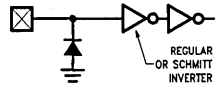
e. Input with Load



f. Hi-Z Input



g. G-Line Inputs with Push-Pull G Outputs



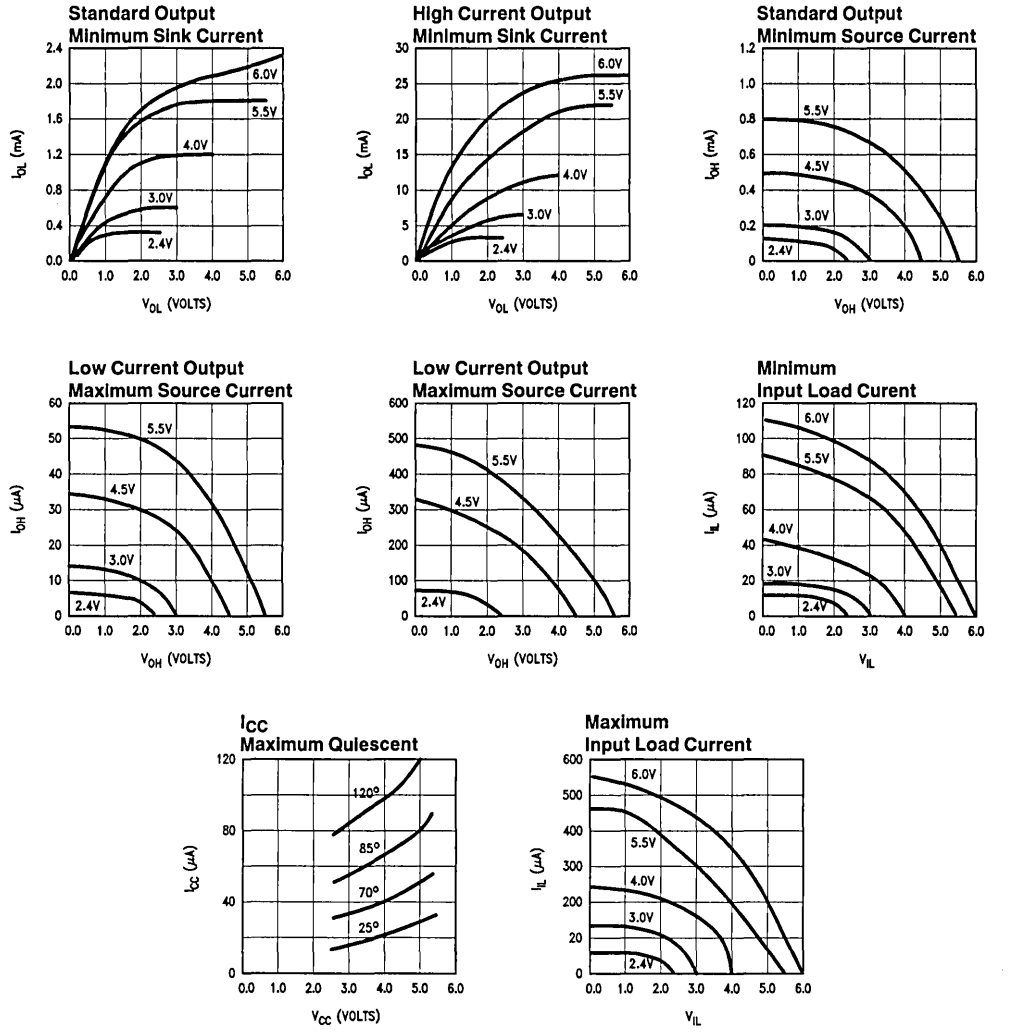
h. G-Line Inputs with Open-Drain G Outputs

TL/DD/9129–15

N1 = STD I_{OL}, N2 = HIGH I_{OL}, P3 = STD I_{OH}, P4 = LOW I_{OH}, P5 = INPUT LOAD, I_{IL} (See Option List)

FIGURE 11. Input/Output Configurations

Power Dissipation (Continued)



TL/DD/9129–16

FIGURE 12. Input/Output Characteristics

Source devices can be mask programmed to be STD current or LOW current I_{OH}. Sink devices can also be mask programmed to be HIGH or STD I_{OL} (on push-pull and open drain outputs, not TRI-STATE "L" outputs). See above DC characteristics for details.

When using either the G or L I/O ports as inputs, a pull-up device is necessary. This can be an external device or the following alternative is available: Select the low-current output option. Now, by setting the output registers to a logic "1" level, the P-Channel devices will act as the pull-up load. Note that when using the L ports in this fashion, the Q registers must be set to a logic "1" level, and the L drivers MUST BE ENABLED by an LEI instruction (see above description).

Minimum and maximum I/V curves are given in *FIGURE 12* for each of the I/O configurations to allow the designer to effectively use them.

When using either the G or L I/O ports as inputs, a pull-up device is necessary. This can be an external device or the following alternative is available: Select the low-current output option. Now, by setting the output registers to a logic "1" level, the P-channel devices will act as the pull-up load. Note that when using the L ports in this fashion the Q registers must be set to a logic "1" level and the L drivers MUST BE ENABLED by an LEI instruction (see description above).

All output drivers use one or more of three common devices numbered 1 to 3. Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figure 12* for each of these devices to allow the designer to effectively use these I/O configurations.

Option List

The COP447C/448C/449C/427C/428C/COP429C mask-programmable options are assigned numbers which correspond with the COP447C/427C pins.

The following is a list of options. The options are programmed at the same time as the ROM pattern to provide the user with the hardware flexibility to interface to various I/O components using little or no external circuitry.

PLEASE FILL OUT THE OPTION TABLE. Xerox the option data and send it in with your disk or EPROM.

Option 1 = 0: Ground Pin — no options available

Option 2: CKO Pin
- = 0: clock generator output to crystal/resonator
- = 1: HALT I/O port
- = 2: general purpose input with load device to V_{CC}
- = 3: general purpose input, high-Z
- = 4: Clock generator output to crystal/resonator with Wake-Up

Option 3: CKI input
- = 0: Crystal controlled oscillator input divide by 4
- = 1: Crystal controlled oscillator input divide by 8
- = 2: Crystal controlled oscillator input divide by 16
- = 3: Single-pin RC controlled oscillator (divide by 4)
- = 4: External oscillator input divide by 4
- = 5: External oscillator input divide by 8
- = 6: External oscillator input divide by 16
- = 7: Crystal oscillator input divide by 32
- = 8: External oscillator input divide by 32

Option 4: $\overline{\text{RESET}}$ input
- = 0: load device to V_{CC}
- = 1: Hi-Z input

Option 5: L7 Driver
- = 0: TRI-STATE push-pull output
- = 1: Low-current TRI-STATE push-pull output
- = 2: Open-drain TRI-STATE output

Option 6: L6 Driver — (same as option 5)

Option 7: L5 Driver — (same as option 5)

Option 8: L4 Driver — (same as option 5)

Option 9: IN1 input
- = 0: load device to V_{CC}
- = 1: Hi-Z input

Option 10: IN2 input — (same as option 9)

Option 11 = 0: V_{CC} Pin — no option available

Option 12: L3 Driver — (same as option 5)

Option 13: L2 Driver — (same as option 5)

Option 14: L1 Driver — (same as option 5)

Option 15: L0 Driver — (same as option 5)

Option 16: SI input — (same as option 9)

Option 17: SO Driver
- = 0: Standard push-pull output
- = 1: Low-current push-pull output
- = 2: Open-drain output, standard sink device

Option 18: SK Driver — (same as option 17)

Option 19: IN0 Input — (same as option 9)

Option 20: IN3 Input — (same as option 9)

Option 21: G0 Driver
- = 0: Push-pull output
- = 1: Push-pull output, low current source device.
- = 2: Open-drain output
- = 3: Push-pull output, high current sink device.

Option 22: G1 Driver — (same as option 21)

Option 23: G2 Driver — (same as option 21)

Option 24: G3 Driver
- = 0: Push-pull output
- = 1: Push-pull output, low current source device
- = 2: Open-drain output
- = 3: Push-pull output, high current sink device
- = 4: Open-drain output, high current sink device
- = 5: Push-pull output, high current sink; low current source (option 5 only available if malfunction detector selected)

Option 25: D3 Output — (same as option 21)

Option 26: D2 Driver
- = 0: Push-pull output
- = 1: Push-pull output, low current source device
- = 2: Open-drain output
- = 3: Push-pull output, high current sink device
- = 4: Open-drain output, high current sink device
- = 5: D0 Dual Clock, RC controlled oscillator (20 pin part only)
- = 6: D0 Dual Clock, external oscillator (20 pin part only)
- = 7: IN1 function with load device to V_{CC} (24 and 20 pin parts only)
- = 8: IN1 function, Hi-Z input (24 and 20 pin parts only)
- = 9: IN2 function, load device to V_{CC} (24 and 20 pin parts only)
- = 10: IN2 function, Hi-Z input (24 and 20 pin parts only)

Option 27: D1 Output — (same as option 21)

Option 28: D0 Output — (same as option 21)

Option 29: Internal Initialization Logic
- = 0: Normal operation
- = 1: No internal initialization logic

Option 30: Dual Clock
- = 0: Normal operation
- = 1: Dual Clock. D0 is an RC oscillator } (opt. #28 must = 2)
- = 2: Dual Clock. D0 is an ext. oscillator } (opt. #28 must = 2)
- = 3: Dual Clock. D0 function bonded out on D2 (20 pin part only)

Option 31: Timer
- = 0: T counter source software selectable with EN6
- = 1: Time-base counter
- = 2: External event counter

Option 32: Microbus
- = 0: Normal
- = 1: Microbus (opt. #31 must = 0)

Option 33: COP bonding
(1K and 2K Microcontroller)
- = 0: 28-pin package
- = 1: 24-pin package
- = 2: Same die purchased in both 24 and 28 pin version (1K Microcontroller only)
- = 3: 20-pin package
- = 4: 28- and 20-pin package
- = 5: 24- and 20-pin package
- = 6: 28-, 24- and 20-pin package

Option List (Continued)

Option 34: Wake-up from HALT control pin selection

- =0: No wake up from HALT
- =1: Wake up on L0
- =2: Wake up on L1
- =3: Wake up on L2
- =4: Wake up on L3
- =5: Wake up on L4
- =6: Wake up on L5
- =7: Wake up on L6
- =8: Wake up on L7
- =9: Wake up on all L pins (any pin grounded will make up clocks)

Option 35: G-port Input configuration selection

- =0: Normal inverter
- =1: Schmitt triggers on G-port inputs

Option 36: 12V operation

- =0: normal operation
- =1: Hi-Z inputs and open drain outputs must withstand +12V.

Option 37: T counter initialization

- =0: T counter cleared on initialization
- =1: T counter not cleared on initialization.

Option 38: Malfunction Indicator

- =0: Normal operation (no malfunction detect)
- =1: Malfunction detector enabled on G3 pin.

Option 39: COP420C compatible (for internal use only)

- =0: not compatible
- =1: Disable Br(2)

Note:—If option 33 = 2, then options 9, 10, 19, 20, and 32 must = 0. If option 33 = 3, 4, 5 or 6 then options 9, 10, 19, 20, 21, 22, 30 and 32 must = 0. If option 33 = 0, 1, 2, 4, 5 or 6, then option 26 must not = 5 or 6. If option 33 = 0, 2, 4 or 6, then option 26 must not = 5, 6, 7, 8, 9 or 10. If option 38 = 1, then option 24 must = 4 or 5, and option 35 must = 1. If option 26 = 5 or 6, then option 28 = 2, option 30 = 3, and option 33 = 3. If option 26 = 6, then option 28 = 2, option 30 = 3, and option 33 = 3. If option 26 = 7 or 8, then option 9 = 0, option 33 = 1, 3 or 5. If option 26 = 9 or 10, then option 10 = 0, option 33 = 1, 3 or 5.

The following option information is to be sent to National along with the EPROM.

OPTION DATA		OPTION DATA	
OPTION 1 VALUE = _____	IS: GROUND PIN	OPTION 20 VALUE = _____	IS: IN3 INPUT
OPTION 2 VALUE = _____	IS: CKO PIN	OPTION 21 VALUE = _____	IS: G0 DRIVER
OPTION 3 VALUE = _____	IS: CKI INPUT	OPTION 22 VALUE = _____	IS: G1 DRIVER
OPTION 4 VALUE = _____	IS: $\overline{\text{RESET}}$ INPUT	OPTION 23 VALUE = _____	IS: G2 DRIVER
OPTION 5 VALUE = _____	IS: L(7) DRIVER	OPTION 24 VALUE = _____	IS: G3 DRIVER
OPTION 6 VALUE = _____	IS: L(6) DRIVER	OPTION 25 VALUE = _____	IS: D3 DRIVER
OPTION 7 VALUE = _____	IS: L(5) DRIVER	OPTION 26 VALUE = _____	IS: D2 DRIVER
OPTION 8 VALUE = _____	IS: L(4) DRIVER	OPTION 27 VALUE = _____	IS: D1 DRIVER
OPTION 9 VALUE = _____	IS: IN1 INPUT	OPTION 28 VALUE = _____	IS: D0 DRIVER
OPTION 10 VALUE = _____	IS: IN2 INPUT	OPTION 29 VALUE = _____	IS: INT INIT LOGIC
OPTION 11 VALUE = _____	IS: VCC PIN	OPTION 30 VALUE = _____	IS: DUAL CLOCK
OPTION 12 VALUE = _____	IS: L(3) DRIVER	OPTION 31 VALUE = _____	IS: TIMER
OPTION 13 VALUE = _____	IS: L(2) DRIVER	OPTION 32 VALUE = _____	IS: MICROBUS
OPTION 14 VALUE = _____	IS: L(1) DRIVER	OPTION 33 VALUE = _____	IS: COP BONDING
OPTION 15 VALUE = _____	IS: L(0) DRIVER	OPTION 34 VALUE = _____	IS: HALT WAKEUP
OPTION 16 VALUE = _____	IS: SI INPUT	OPTION 35 VALUE = _____	IS: G PORT CONFIGURATION
OPTION 17 VALUE = _____	IS: SO DRIVER	OPTION 36 VALUE = _____	IS: 12V OPERATION
OPTION 18 VALUE = _____	IS: SK DRIVER	OPTION 37 VALUE = _____	IS: T COUNTER INIT
OPTION 19 VALUE = _____	IS: IN0 INPUT	OPTION 38 VALUE = _____	IS: MALFUNCTION INDICATOR
		OPTION 39 VALUE = _____	IS: (FOR INTERNAL USE ONLY)

National Semiconductor Corporation

COP440/COP441/COP442 and COP340/COP341/COP342 Single-Chip N-Channel Microcontrollers

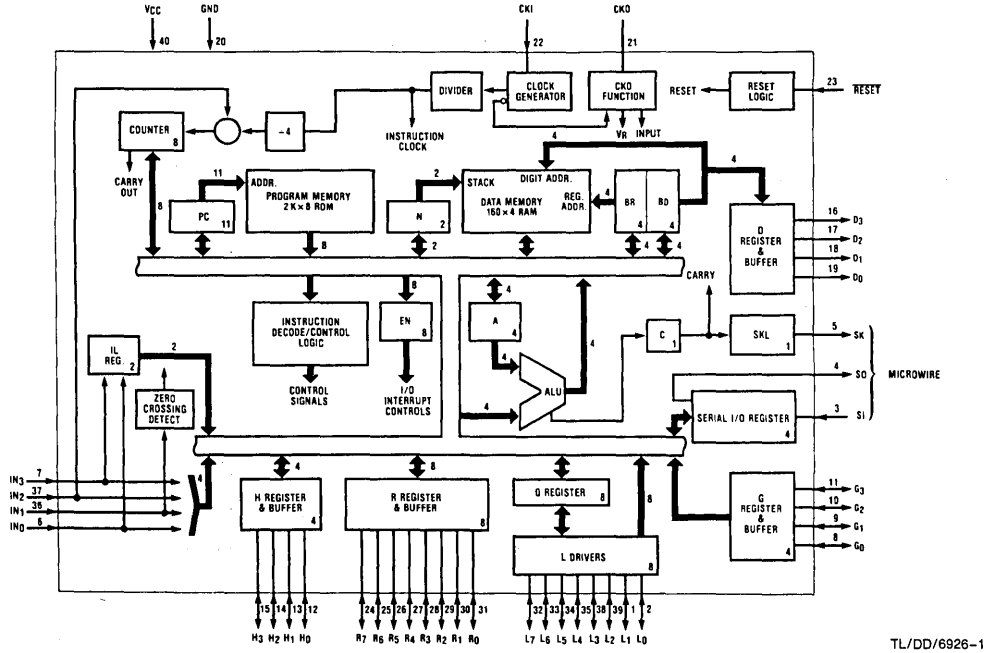
General Description

The COP440, COP441, COP442, COP340, COP341, and COP342 Single-Chip N-Channel Microcontrollers are members of the COPS™ microcontrollers family, fabricated using N-channel, silicon gate MOS technology. These are complete microcontrollers with all system timing, internal logic, ROM, RAM, and I/O necessary to implement dedicated control functions in a variety of applications. Features include single supply operation, various output configuration options, and an instruction set, internal architecture, and I/O scheme designed to facilitate keyboard input, display output, and data manipulation. The COP440 is a 40-pin chip and the COP441 is a 28-pin version of the same circuit (12 I/O lines removed). The COP442 is a 24-pin version (4 more input lines removed). The COP340, COP341, COP342 are functional equivalents of the above devices respectively, but operate with an extended temperature range (−40°C to +85°C). Standard test procedures and reliable high-density fabrication techniques provide the medium to large volume customers with a customized controller oriented processor at a low end-product cost.

Features

- Enhanced, more powerful instruction set
- 2k x 8 ROM, 160 x 4 RAM
- 35 I/O lines (COP440)
- Zero-crossing detect circuitry with hysteresis
- True multi-vectored interrupt from 4 selectable sources (plus restart)
- Four-level subroutine stack (in RAM)
- 4 μs cycle time
- Single supply operation (4.5V–6.3V)
- Programmable time-base counter
- Internal binary counter/register with MICROWIRE™ compatible serial I/O
- General purpose and TRI-STATE® outputs
- TTL/CMOS compatible in and out
- LED drive capability
- MICROBUS™ compatible
- Software/hardware compatible with other members of the COP400 family
- Extended temperature range devices COP340, COP341, COP342 (−40°C to +85°C)
- Compatible dual CPU device available (COP2440 series)

Block Diagram



TL/DD/6926–1

FIGURE 1

COP440/COP441/COP442 Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Zero-Crossing Detect Pin Relative to GND	−1.2V to +15V
Voltage at Any Other Pin Relative to GND	−0.5V to +7V
Ambient Operating Temperature	0°C to +70°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C
Power Dissipation	0.75W at 25°C 0.4W at 70°C
Total Source Current	150 mA
Total Sink Current	75 mA

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 6.3V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Operating Voltage (V_{CC})	(Note 3)	4.5	6.3	V
Power Supply Ripple	(Peak to Peak)		0.4	V
Operating Supply Current	(All Inputs and Outputs Open)			
	T_A = 0°C		44	mA
	T_A = 25°C		35	mA
	T_A = 70°C		27	mA
Input Voltage Levels				
CKI Input Levels				
Crystal Input (÷16, ÷8)				
Logic High (V_{IH})	V_{CC} = Max	3.0		V
Logic High (V_{IH})	V_{CC} = 5V ±5%	2.0		V
Logic Low (V_{IL})		−0.3	0.4	V
Schmitt Trigger Input (÷4)				
Logic High (V_{IH})		0.7 V_{CC}		V
Logic Low (V_{IL})		−0.3	0.6	V
$\overline{\text{RESET}}$ Input Levels	(Schmitt Trigger Input)			
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.6	V
Zero-Crossing Detect Input	See *Figure 7*			
Trip Point		−0.15	0.15	V
Logic High (V_{IH}) Limit			12	V
Logic Low (V_{IL}) Limit		−0.8		V
SO Input Level (Test Mode)	(Note 5)	2.0	2.5	V
All Other Inputs				
Logic High	V_{CC} = Max	3.0		V
Logic High	V_{CC} = 5V ±5%	2.0		V
Logic Low		−0.3	0.8	V
Input Levels High Trip Option				
Logic High		3.6		V
Logic Low		−0.3	1.2	V
Input Capacitance			7.0	pF
Hi-Z Input Leakage		−1.0	+1.0	µA

Note 1: Duty Cycle = $t_{WI}/(t_{WI} + t_{WO})$.

Note 2: See Figure for additional I/O Characteristics.

Note 3: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Note 4: Exercise great care not to exceed maximum device power dissipation limits when direct-driving LEDs (or sourcing similar loads) at high temperature.

Note 5: SO output "0" level must be less than 0.8V for normal operation.

COP440/COP441/COP442
DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 6.3V$ unless otherwise noted (Continued)

Parameter	Conditions	Min	Max	Units
Output Voltage Levels				
Standard Output				
TTL Operation				
Logic High (V_{OH})	$I_{OH} = -100\ \mu A$	2.4		V
Logic Low (V_{OL})	$I_{OL} = 1.6$ mA		0.4	V
CMOS Operation (Note 1)				
Logic High (V_{OH})	$I_{OH} = -10\ \mu A$	$V_{CC} - 0.4$		V
Logic Low (V_{OL})	$I_{OL} = 10\ \mu A$		0.2	V
Output Current Levels				
Standard Output Source Current	$V_{CC} = 4.5V$, $V_{OH} = 2.4V$	−100	−650	μA
LED Direct Drive Output	$V_{CC} = 6V$, $V_{OH} = 2V$			
Logic High (I_{OH})		−2.5	−17	mA
TRI-STATE Output Leakage Current		−2.5	+2.5	μA
CKO Output				
Oscillator Output Option				
Logic High	$V_{OH} = 2V$	−0.2		mA
Logic Low	$V_{OL} = 0.4V$	0.4		mA
V_R RAM Power Supply Option				
Supply Current	$V_R = 3.3V$		3.0	mA
CKI Sink Current (RC Option)	$V_{IH} = 3.5V$, $V_{CC} = 4.5V$	2.0		mA
Input Current Levels				
Zero-Crossing Detect Input				
Resistance	$V_{IH} = 1.0V$	0.9	4.6	kΩ
Input Load Source Current	$V_{IH} = 2.0V$, $V_{CC} = 4.5V$	14	230	μA
Total Sink Current Allowed				
All I/O Combined			75	mA
Each L, R Port			20	mA
Each D, G, H Port			10	mA
SO, SK			2.5	mA
Total Source Current Allowed				
All I/O Combined			150	mA
L Port			120	mA
L_7–L_4			70	mA
L_3–L_0			70	mA
Each L Pin			23	mA
All Other Output Pins			1.6	mA

Note 1: TRI-STATE and LED configurations are excluded.

COP340/COP341/COP342 Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Zero-Crossing Detect Pin Relative to GND	−1.2V to +15V
Voltage at Any Other Pin Relative to GND	−0.5V to +7V
Ambient Operating Temperature	−40°C to +85°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C
Power Dissipation	0.75W at 25°C 0.25W at 85°C
Total Source Current	150 mA
Total Sink Current	75 mA

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 5.5V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Operating Voltage (V_{CC})	(Note 3)	4.5	5.5	V
Power Supply Ripple	(Peak to Peak)		0.4	V
Operating Supply Current	(All Inputs and Outputs Open)			
	$T_A = -40°C$		54	mA
	$T_A = 25°C$		35	mA
	$T_A = 85°C$		25	mA
Input Voltage Levels				
CKI Input Levels				
Crystal Input (÷16, ÷8)	V_{CC} = Max	3.0		V
Logic High (V_{IH})		2.2		V
Logic Low (V_{IL})		−0.3	0.3	V
Schmitt Trigger Input (÷4)				
Logic High (V_{IH})		0.7 V_{CC}		V
Logic Low (V_{IL})		−0.3	0.4	V
$\overline{\text{RESET}}$ Input Levels	(Schmitt Trigger Input)			
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.4	V
Zero-Crossing Detect Input	See *Figure 7*			
Trip Point		−0.15	0.15	V
Logic High (V_{IH}) Limit			12	V
Logic Low (V_{IL}) Limit		−0.8		V
SO Input Level (Test Mode)	(Note 5)	2.2	2.4	V
All Other Inputs	V_{CC} = Max	3.0		V
Logic High		2.2		V
Logic Low		−0.3	0.6	V
Input Levels High Trip Option				
Logic High		3.6		V
Logic Low		−0.3	1.2	V
Input Capacitance			7.0	pF
Hi-Z Input Leakage		−2.0	+2.0	μA

Note 1: Duty Cycle = $t_{WI}/(t_{WI} + t_{WO})$.

Note 2: See Figure for additional I/O Characteristics.

Note 3: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Note 4: Exercise great care not to exceed maximum device power dissipation limits when direct-driving LEDs (or sourcing similar loads) at high temperature.

Note 5: SO output "0" level must be less than 0.6V for normal operation.

COP340/COP341/COP342
DC Electrical Characteristics

$-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 5.5V$ unless otherwise noted (Continued)

Parameter	Conditions	Min	Max	Units
Output Voltage Levels				
Standard Output				
TTL Operation				
Logic High (V_{OH})	$I_{OH} = -100\ \mu A$	2.4		V
Logic Low (V_{OL})	$I_{OL} = 1.6$ mA		0.4	V
CMOS Operation (Note 1)				
Logic High (V_{OH})	$I_{OH} = -10\ \mu A$	$V_{CC} - 0.5$		V
Logic Low (V_{OL})	$I_{OL} = 10\ \mu A$		0.2	V
Output Current Levels				
Standard Output Source Current	$V_{CC} = 4.5V$, $V_{OH} = 2.4V$	−100	−800	μA
LED Direct Drive Output	$V_{CC} = 5V$ (Note 4)			
Logic High (I_{OH})	$V_{OH} = 2V$	−1.5	−15	mA
TRI-STATE Output Leakage Current		−5.0	+5.0	μA
CKO Output				
Oscillator Output Option				
Logic High	$V_{OH} = 2V$	−0.2		mA
Logic Low	$V_{OL} = 0.4V$	0.4		mA
V_R RAM Power Supply Option				
Supply Current	$V_R = 3.3V$		4.0	mA
CKI Sink Current (RC Option)	$V_{CC} = 4.5V$, $V_{IH} = 3.5V$	2.0		mA
Input Current Levels				
Zero-Crossing Detect Input				
Resistance	$V_{IH} = 1.0V$	0.9	4.6	kΩ
Input Load Source Current	$V_{IH} = 2.0V$, $V_{CC} = 4.5V$	14	280	μA
Total Sink Current Allowed				
All I/O Combined			75	mA
Each L, R Port			20	mA
Each D, G, H Port			10	mA
SO, SK			2.5	mA
Total Source Current Allowed				
All I/O Combined			150	mA
L Port			120	mA
L_7–L_4			70	mA
L_3–L_0			70	mA
Each L Pin			23	mA
All Other Output Pins			1.6	mA

Note 1: TRI-STATE and LED configurations are excluded.

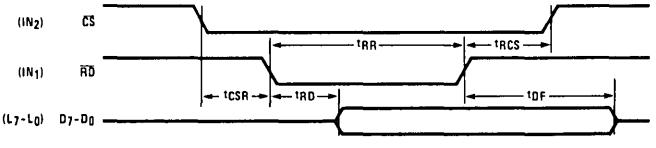
AC Electrical Characteristics

COP440/COP441/COP442: $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 6.3V$ unless otherwise noted

COP340/COP341/COP342: $-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 5.5V$ unless otherwise noted

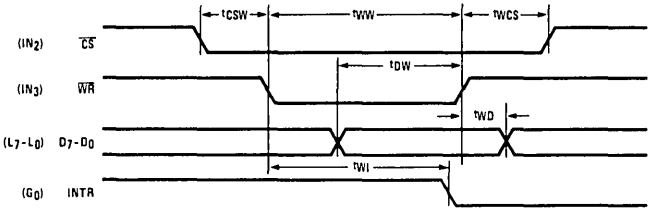
Parameter	Conditions	Min	Max	Units
Instruction Cycle Time–t_E		4.0	10	μs
CKI Frequency	÷16 Mode	1.6	4.0	MHz
	÷8 Mode	0.8	2.0	MHz
	÷4 Mode	0.4	1.0	MHz
Duty Cycle (Note 1)	f_I = 4 MHz	30	60	%
Rise Time	f_I = 4 MHz External Clock		60	ns
Fall Time	f_I = 4 MHz External Clock		40	ns
CKI Using RC *(Figure 9c)*	÷4 Mode			
Frequency	R = 15 kΩ ±5%, C = 100 pF ±10%	0.5	1.0	MHz
Instruction Execution Time–t_E (Note 1)		4.0	8.0	μs
INPUTS: *(Figure 4)*				
SI				
t_{SETUP}		0.3		μs
t_{HOLD}		300		ns
All Other Inputs				
t_{SETUP}		1.7		μs
t_{HOLD}		300		ns
OUTPUT PROPAGATION DELAY	Test Condition: C_L = 50 pF, V_{OUT} = 1.5V			
CKO				
t_{pd1}, t_{pd0}	Crystal Input		0.17	μs
t_{pd1}, t_{pd0}	Schmitt Trigger Input		0.3	μs
SO, SK				
t_{pd1}, t_{pd0}	R_L = 2.4 kΩ		1.0	μs
All Other Outputs	R_L = 5.0 kΩ		1.4	μs
MICROBUS TIMING	C_L = 100 pF, V_{CC} = 5V ±5%			
Read Operation *(Figure 2a)*	TRI-STATE Outputs			
Chip Select Stable Before $\overline{RD}$–t_{CSR}		65		ns
Chip Select Hold Time for $\overline{RD}$–t_{RCS}		20		ns
$\overline{RD}$ Pulse Width–t_{RR}		400		ns
Data Delay from $\overline{RD}$–t_{RD}			375	ns
$\overline{RD}$ to Data Floating–t_{DF}			250	ns
Write Operation *(Figure 2b)*				
Chip Select Stable Before $\overline{WR}$–t_{CSW}		65		ns
Chip Select Hold Time for $\overline{WR}$–t_{WCS}		20		ns
$\overline{WR}$ Pulse Width–t_{WW}		400		ns
Data Set-Up Time for $\overline{WR}$–t_{DW}		320		ns
Data Hold Time for $\overline{WR}$–t_{WD}		100		ns
INTR Transition Time from $\overline{WR}$–t_{WI}			700	ns

Note 1: Variation due to the device included.



TL/DD/6926-2

FIGURE 2a. MICROBUS Read Operation Timing



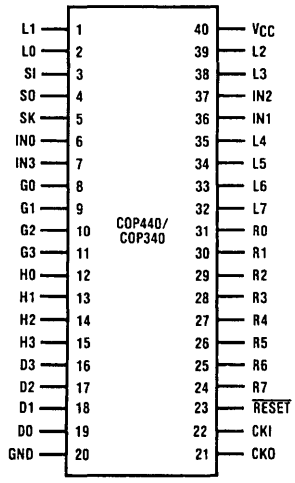
TL/DD/6926-3

FIGURE 2b. MICROBUS Write Operation Timing

1

Connection Diagrams

Dual-In-Line Package

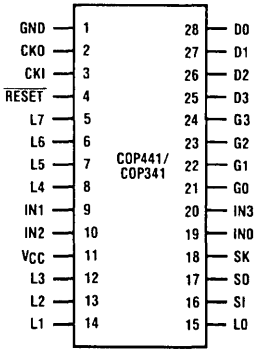


TL/DD/6926-4

Top View

Order Number COP440N or COP340N
See NS Package Number N40A

Dual-In-Line Package

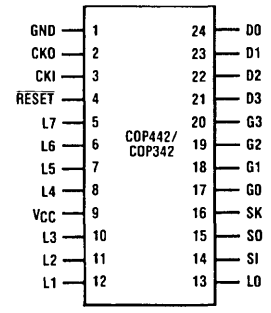


TL/DD/6926-5

Top View

Order Number COP441N or COP341N
See NS Package Number N28B

Dual-In-Line Package



TL/DD/6926-6

Top View

Order Number COP442N or COP342N
See NS Package Number N24A

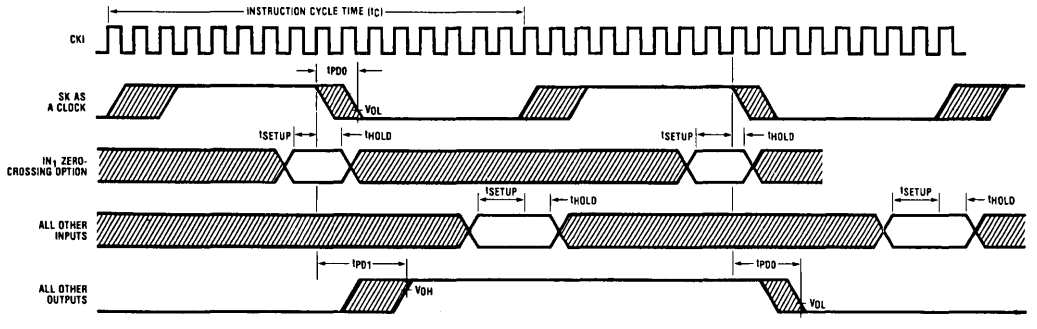
FIGURE 3

Pin Descriptions

Pin	Description
L_7–L_0	8-bit Bidirectional I/O Port with TRI-STATE
G_3–G_0	4-bit Bidirectional I/O Port
D_3–D_0	4-bit General Purpose Output Port
IN_3–IN_0	4-bit General Purpose Input Port (Not Available on COP442/COP342)
SI	Serial Input
SO	Serial Output (or General Purpose Output)
SK	Logic-Controlled Clock (or General Purpose Output)

Pin	Description
CKI	System Oscillator Input
CKO	System Oscillator Output (or General Purpose Input or RAM Power Supply)
$\overline{RESET}$	System Reset Input
V_{CC}	Power Supply
GND	Ground
H_3–H_0	4-bit Bidirectional I/O Port (COP440/COP340 Only)
R_7–R_0	8-Bit Bidirectional I/O Port with TRI-STATE (COP440/COP340 Only)

Timing Diagram



TL/DD/6926-7

FIGURE 4. Input/Output Timing Diagrams (Divide by 16 Mode)

Functional Description

The block diagram of the COP440 is shown in *Figure 1*. Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1" (greater than 2.0V). When a bit is reset, it is a logic "0" (less than 0.8V).

PROGRAM MEMORY

Program Memory consists of a 2,048 byte ROM. As can be seen by an examination of the COP440 instruction set, these words may be program instructions, constants, or ROM addressing data. Because of the special characteristics associated with the JP, JSRP, JID, LQID, and LID instructions, ROM must often be thought of as being organized into 32 pages of 64 words each.

ROM addressing is accomplished by an 11-bit PC register. Its binary value selects one of the 2,048 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 11-bit binary count value.

ROM instruction words are fetched, decoded and executed by the Instruction Decode, Control and Skip Logic circuitry.

DATA MEMORY

Data memory consists of a 640-bit RAM, organized as 10 data registers of 16 4-bit digits. RAM addressing is implemented by an 8-bit B register whose upper 4 bits (Br) select 1 of 10 (0–9) data registers and lower 4 bits (Bd) select 1 of 16 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) is usually loaded into, or from, or exchanged with the A register (accumulator), it may also be loaded into or from the Q latches, L port, R port, EN register, and T counter (internal time base counter). RAM may also be loaded from 4 bits of a ROM word. RAM addressing may also be performed directly to the lower 8 registers by the LDD and XAD instructions based upon the 7-bit contents of the operand field of these instructions. The Bd register also serves as a source register for 4-bit data sent directly to the D outputs. RAM register 8 (Br = 8) also serves as a subroutine stack. Note that it is possible, but not recommended, to alter the contents of the stack by normal data memory access commands.

INTERNAL LOGIC

The 4-bit A register (accumulator) is the source and destination register for most I/O arithmetic, logic, and data memory access operations. It can also be used to load the Br and Bd portions of the B register, N register, to load and input 4 bits of the 8-bit Q latch, EN register, or T counter, to input 4 bits of a ROM word, L or R I/O port data, to input 4-bit G, H, or IN ports, and to perform data exchanges with the SIO register. The accumulator is cleared upon reset.

A 4-bit adder performs the arithmetic and logic functions of the COP440, storing its results in A. It also outputs a carry bit to the 1-bit C register, most often employed to indicate arithmetic overflow. The C register, in conjunction with the XAS instruction and the EN register, also serves to control the SK output. C can be outputted directly to SK or can enable SK to be a sync clock each instruction cycle time. (See XAS instruction and EN register description, below).

The 8-bit T counter is a binary up counter which can be loaded to and from M and A. The input to this counter is software selectable from two sources: the first coming from a divide-by-four prescaler (from instruction cycle frequency) thus providing a 10-bit time base counter; the second coming from IN_2 input, changing the T counter into an 8-bit external event counter (see EN register below). In this mode, a low-going pulse ("1" to "0") of at least 2 instruction cycles wide will increment the counter. When the counter overflows, an overflow flag will be set (see SKT instruction below) and an interrupt signal will be sent to processor X. The T counter is cleared on reset.

Four general-purpose inputs, IN_3–IN_0, are provided; IN_1, IN_2 and IN_3 may be selected, by a mask-programmable option, as Read Strobe, Chip Select, and Write Strobe inputs, respectively, for use in MICROBUS applications; IN_1, by another mask-programmable option, can be selected as a true zero-crossing detector with the output triggering an interrupt or being interrogated by an instruction. These two mask-programmable options are mutually exclusive.

The D register provides 4 general-purpose outputs and is used as the destination register for the 4-bit contents of Bd.

The G register contents are outputs to a 4-bit general-purpose bidirectional I/O port. G_0 may be mask-programmed as an output for MICROBUS applications.

The H register contents are outputs to a 4-bit general-purpose bidirectional I/O port.

The Q register is an internal, latched, 8-bit register, used to hold data loaded to or from M and A, as well as 8-bit data from ROM. Its contents are outputted to the L I/O ports when the L drivers are enabled under program control. With the MICROBUS option selected, Q can also be loaded with the 8-bit contents of the L I/O ports upon the occurrence of a write strobe from the host CPU. Note that unlike most other COPS controllers, Q is cleared on reset.

The 8 L drivers, when enabled, output the contents of latched Q data to the L I/O ports. Also, the contents of L may be read directly into A and M. As explained above, the MICROBUS option allows L I/O port data to be latched into the Q register. The L I/O port can be directly connected to the segments of a multiplexed LED display (using the LED Direct Drive output configuration option) with Q data being outputted to the Sa–Sg and decimal point segments of the display.

The R register, when enabled, outputs to an 8-bit general-purpose, bidirectional, I/O port.

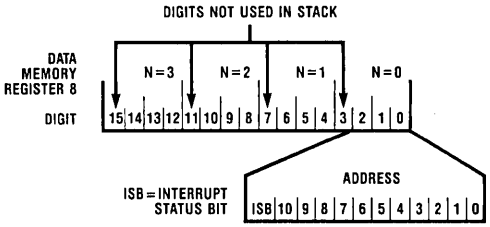
The SIO register functions as a 4-bit serial-in/serial-out shift register for MICROWIRE I/O and COPS peripherals, or as a binary counter (depending on the contents of the EN register; see EN register description, below). Its contents can be exchanged with A, allowing it to input or output a continuous serial data stream.

The XAS instruction copies the C flag into the SKL latch. In the counter mode, SK is the output of SKL; in the shift register mode, SK outputs SKL ANDed with the instruction cycle clock.

The 2-bit N register is a stack pointer to the data memory register 8 where the subroutine return address is located. It points to the next location where the address may be stored

Functional Description (Continued)

and increments by 1 after each push of the stack, and decrements by 1 before each pop. The N register can be accessed by exchanging its value with A and is cleared on reset. The stack is 4 addresses deep, 12 bits wide, and does not check for overflow or empty conditions. The RAM digit locations where the addresses are stored are shown in *Figure 5*. The LSBs of the addresses are at digits 0, 4, 8, and 12. The MSBs of digits 2, 6, 10, and 14 contain an interrupt status bit (see Interrupt description, below). The four unused digits (3, 7, 11, and 15) can be used as general data storage. When a subroutine call or interrupt occurs, an 11-bit return address and an interrupt status bit are stored in the stack. The N register is then incremented. When a RET or RETSK instruction is executed, the N register is decremented and then the return address is fetched and loaded into the program counter. The address and interrupt status bits remain in the stack, but will be overwritten when the next subroutine call or interrupt occurs.



TL/DD/6926–8

FIGURE 5. Subroutine Return Address Stack Organization

The EN register in an internal 8-bit register loaded under program control by the LEI instruction (lower 4 bits) or by the CAME instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register.

0. The least significant bit of the enable register, EN_0, selects the SIO register as either a 4-bit shift register or a 4-bit binary counter. With EN_0 set, SIO is an asynchronous binary counter, decrementing its value by one upon each low-going pulse ("1" to "0") occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output is equal to the value of EN_3. With EN_0 reset, SIO is a serial shift register left shifting 1 bit each instruction cycle time. The data present at SI goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. The SK output becomes a logic-controlled clock.
1. With EN_1 set, interrupt is enabled with EN_4 and EN_5 selecting the interrupt source. Immediately following an interrupt, EN_1 is reset to disable further interrupts.
2. With EN_2 set, the L drivers are enabled to output the data in Q to the L I/O port. Resetting EN_2 disables the L drivers, placing the L I/O port in a high-impedance input state. A special feature of the COP440 and COP441 is that the MICROBUS option will change the function of this bit to disable any writing into G_0 when EN_2 is set.
3. EN_3, in conjunction with EN_0, affects the SO output. With EN_0 set (binary counter option selected) SO will output the value loaded into EN_3. With EN_0 reset (serial shift register option selected), setting EN_3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN_3 with the serial shift register option selected disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains set to "0". Table I below provides a summary of the modes associated with EN_3 and EN_0.
4. EN_5 and EN_4 select the source of the interrupt signal.
5. The possible sources are as follows:

EN_5	EN_4	Interrupt Source
0	0	IN_1 (low-going pulse)
0	1	CKO input (if mask-programmed as an input)
1	0	Zero-crossing (or IN_1 level transition)
1	1	T counter overflows

   EN_4 determines the interrupt routine location.
6. With EN_6 set, the internal 8-bit T counter will use IN_2 as its input. With EN_6 reset, the input to the T counter is the output of a divide by four prescaler (from instruction cycle frequency), thus providing a 10-bit time-base counter.
7. With EN_7 set, the R outputs are enabled; if $EN_7 = 0$, the R outputs are disabled.

INTERRUPT

The following features are associated with the interrupt procedure and protocol and must be considered by the programmer when utilizing interrupts.

a. The interrupt, once acknowledged as explained below, pushes the next sequential program counter address (PC+1) together with an interrupt status bit, onto the program counter stack residing in data memory. Any previous contents at the bottom of the stack are lost. The program counter is set to hex address 0FF (the last word of page 3) and EN_1 is reset. If EN_4 is reset, the next program address is hex 100; if EN_4 is set, the next program address is hex 300; thus providing a different interrupt location for different interrupt sources.

TABLE I. Enable Register Modes — Bits EN_3 and EN_0

EN_3	EN_0	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL = 1, SK = Clock If SKL = 0, SK = 0
1	0	Shift Register	Input to Shift Register	Serial Out	If SKL = 1, SK = Clock If SKL = 0, SK = 0
0	1	Binary Counter	Input to Binary Counter	0	If SKL = 1, SK = 1 If SKL = 0, SK = 0
1	1	Binary Counter	Input to Binary Counter	1	If SKL = 1, SK = 1 If SKL = 0, SK = 0

Functional Description (Continued)

b. An interrupt will be acknowledged only after the following conditions are met:
   1. EN_1 has been set.
   2. For an external interrupt input, the signal pulse must be at least two instruction cycles wide.
   3. A currently executing instruction has been completed.
   4. All successive transfer of control instructions and successive LBIs have been completed (e.g., if the main program is executing a JP instruction which transfers program control to another JP instruction, the interrupt will not be acknowledged until the second JP instruction has been executed.

c. The instruction at hex address 0FF must be a NOP.

d. A CAME or LEI instruction may be put immediately before the RET instruction to re-enable interrupts.

e. If the interrupt signal source is being changed, the interrupt must be disabled prior to, or at, the same time with the change to avoid false interrupts. An interrupt may be enabled only if the interrupt source is not changing. A sample code for changing the interrupt source and enabling the interrupt is as follows:

```
CAME        ; disable interrupt & alter interrupt source
SMB  1      ; set interrupt enable bit
CAME        ; enable interrupt
```

f. An interrupt status bit is stored together with the return address in the stack. The status bit is set if an interrupt occurs at a point in the program where the next instruction is to be skipped; upon returning from the interrupt routine, this set status bit will cause the next instruction to be skipped. Subroutine and interrupt nesting inside interrupt routines are allowed. Note that this differs from the COP420/420C/420L/444L series.

MICROBUS INTERFACE
(not available in COP442, COP342)

The COP440 series has an option which allows them to be used as peripheral microprocessor devices, inputting and outputting data from and to a host microprocessor (μP). IN_1, IN_2 and IN_3 general purpose inputs become MICROBUS-compatible read-strobe, chip-select, and write-strobe lines, respectively. IN_1 becomes $\overline{RD}$—a logic "0" on this input will cause Q latch data to be enabled to the L ports for input to the μP. IN_2 becomes $\overline{CS}$—a logic "0" on this line selects the COPS processor as the μP peripheral device by enabling the operation of the $\overline{RD}$ and $\overline{WR}$ lines and allows for the selection of one of several peripheral components. IN_3 becomes $\overline{WR}$—a logic "0" on this line will write bus data from the L ports to the Q latches for input to the COPS processor. G_0 becomes INTR, a "ready" output, reset by a write pulse from the μP on the $\overline{WR}$ line, providing the "handshaking" capability necessary for asynchronous data transfer between the host CPU and the COPS processor. G_0 output can be separated from other G outputs by the EN_2 bit (see EN description above).

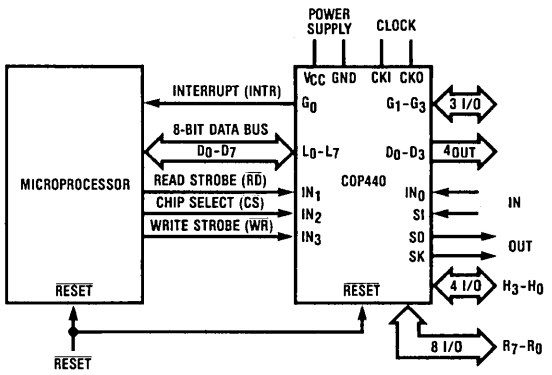
This option has been designed for compatibility with National's MICROBUS—a standard interconnect system for 8-bit parallel data transfer between MOS/LSI CPUs and interfacing devices. (See MICROBUS National Publication.) The functional and timing relationships between the COPS processor signal lines affected by this option are as specified for the MICROBUS interface, and are given in the AC electrical characteristics and shown in the timing diagrams *(Figure 2)*. Connection of the COP440 to the MICROBUS is shown in *Figure 6*.

Note: TRI-STATE outputs must be used on L port.

ZERO-CROSSING DETECTION
(not available on the COP442, COP342)

The following features are associated with the IN_1 pin: ININ and INIL instructions input the state of IN_1 to A_1; IN_1 interrupt generates an interrupt pulse when a low-going transition ("1" to "0") occurs on IN_1; zero-crossing interrupt generates an interrupt pulse when an IN_1 transition occurs (both "1" to "0" and "0" to "1").

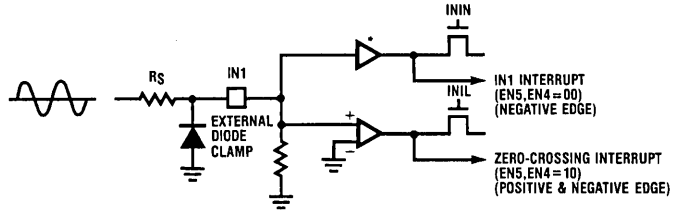
If the zero-crossing detector is mask-programmed in (see *Figure 7a*), the INIL instruction and zero-crossing interrupt will input the state of IN_1 through the true zero-crossing detector ("1" if input > 0V, "0" if input < 0V). The ININ instruction and IN_1 interrupt will then have unique logic HIGH and LOW levels depending on the IN port input level chosen. If normal (TTL) level is chosen, logic HIGH level is 3.0V (3.3V for COP340/341) and logic LOW level is 0.8V



TL/DD/6926-9

FIGURE 6. MICROBUS Option Interconnect

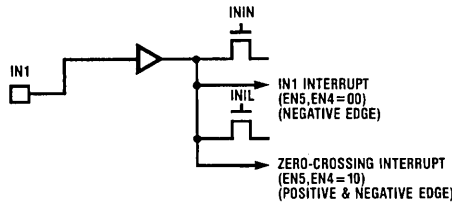
Functional Description (Continued)



TL/DD/6926-10

*Note: This input has a different set of logic HIGH and LOW levels; see above description.

a. Zero-Crossing Detect Logic Option



TL/DD/6926-11

b. IN, without Zero-Crossing Detect Logic

FIGURE 7. IN, Mask-Programmable Options

(0.6V for COP340/341); if high trip level is chosen, logic HIGH level is 5.4V and logic LOW level is 1.2V. If the zero-crossing detector is not mask-programmed in (see *Figure 7b*), IN_1 will have logic HIGH and LOW levels that are defined for the IN port (see option list).

The zero-crossing detector input contains a small hysteresis (50 mV typical) to eliminate signal noise, and is not a high impedance input but contains a resistive load to ground. Since this input can withstand a voltage range of −0.8V to +12V, an external clamping diode is needed for most input signals, as shown in *Figure 7a*, to limit the voltage below ground. An external resistor, R_S may be needed for the following two cases:

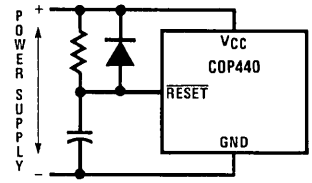
a. Input signal exceeds 12V; R_S and the internal resistor act as a voltage divider to reduce the voltage at the input pin to below 12V.

b. Signal comes from a low impedance source; when the voltage at the pin is clamped to −0.7V by the forward bias voltage of an external diode, R_S limits the current going through the diode.

INITIALIZATION

The $\overline{\text{RESET}}$ pin is configured as a Schmitt trigger input. If not used, it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the $\overline{\text{RESET}}$ input, provided it stays low for at least three instruction cycle times. The user must provide an external RC network and diode to the $\overline{\text{RESET}}$ pin as in *Figure 8*. The external POR (Power-on-Reset) delay must be greater than the internal POR. The internal POR delay is 2600 internal clock cycles.

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN, G, H, IL, L, N, Q, R, and T registers are cleared. The SK output is enabled as a SYNC output by setting the SKL latch, thus providing a clock. RAM (data memory and stack) is not cleared. The first instruction at address 0 must be a CLRA.



TL/DD/6926-12

RC ≥ 5 × power supply rise time

FIGURE 8. Power-Up Clear Circuit

OSCILLATOR

There are three basic clock oscillator configurations available, as shown by *Figure 9*.

a. Crystal Controlled Oscillator. CKI and CKO are connected to an external crystal. The cycle frequency equals the crystal frequency divided by 16 (optional by 8). Thus a 4 MHz crystal with the divide-by-16 option selected will give a 250 kHz cycle frequency (4 μs instruction cycle time).

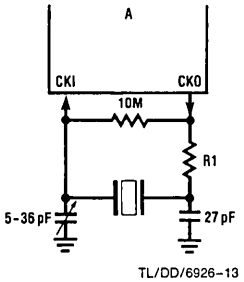
b. External Oscillator. CKI is an external clock input signal. The external frequency is divided by 16 (optional by 8 or 4) to give the cycle frequency. If the divide-by-4 option is selected, the CKI input level is the Schmitt-trigger level. CKO is now available to be used as the RAM power supply (V_R) or as a general purpose input.

c. RC Controlled Oscillator. CKI is configured as a single pin RC controlled Schmitt trigger oscillator. The cycle frequency equals the oscillation frequency divided by 4. CKO is available for non-timing functions.

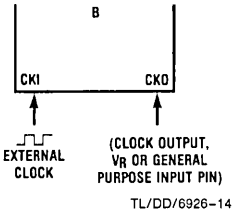
CKO PIN OPTIONS

As an option, CKO can be an oscillator output. In a crystal controlled oscillator system, this signal is used as an output to the crystal network. As another option, CKO can be an interrupt input or a general purpose input, reading into bit 2 of A (accumulator) through the INIL instruction. As another option, CKO can be a RAM power supply pin (V_R), allowing

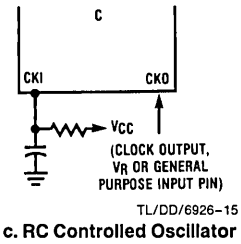
Functional Description (Continued)



a. Crystal Oscillator



b. External Oscillator



c. RC Controlled Oscillator

Crystal Oscillator

Crystal Value	R_1
4 MHz	1k
3.58 MHz	1k
2.10 MHz	2k

RC Controlled Oscillator

R (kΩ)	C (pF)	Instruction Execution Time (μs)
13	100	5.0 ±20%
6.8	220	5.3 ±23%
8.2	300	8.0 ±22%
22	100	8.2 ±17%

Note: 5 kΩ ≤ R ≤ 50 kΩ
50 pF ≤ C ≤ 360 pF

FIGURE 9. COP440/441/442 Oscillators

its connection to a standby/backup power supply to maintain the data integrity of RAM registers 0–3 with minimum power drain when the main supply is inoperative or shut down to conserve power. Using either of the two latter options is appropriate in applications where the system configuration does not require use of the CKO pin for timing functions.

RAM KEEP-ALIVE OPTION

Selecting CKO as the RAM power supply (V_R) allows the user to shut off the chip power supply (V_{CC}) and maintain data in the lower 4 registers of the RAM. To insure that RAM data integrity is maintained, the following conditions must be met:

1. $\overline{\text{RESET}}$ must go low before V_{CC} goes below spec during power-off; V_{CC} must be within spec before $\overline{\text{RESET}}$ goes high on power-up.
2. When V_{CC} is on, V_R must be within the operating voltage range of the chip, and within 1V of V_{CC}.
3. V_R must be ≥ 3.3V with V_{CC} off.

I/O OPTIONS

COP440 inputs have the following optional configurations, illustrated in *Figure 10*.

a. An on-chip depletion load device to V_{CC}.

b. A Hi-Z input which must be driven to a "1" or "0" by external components.

c. A resistive load to GND for the zero-crossing input option (IN_1 only).

COP440 outputs have the following optional configurations:

d. Standard—an enhancement mode device to ground in conjunction with a depletion-mode device to V_{CC}, compatible with TTL and CMOS input requirements. Available on SO, SK, D, G, and H outputs.

e. Open-Drain—an enhancement-mode device to ground only, allowing external pull-up as required by the user's application. Available on SO, SK, D, G, L, H, and R outputs.

f. Push-Pull—an enhancement-mode device to ground in conjunction with a depletion-mode device paralleled by an enhancement-mode device to V_{CC}. This configuration has been provided to allow for fast rise and fall times when driving capacitive loads. Available on SO and SK outputs only.

g. Standard L,R—same as **d.**, but may be disabled. Available on L and R outputs only (disabled on reset).

h. LED Direct Drive—an enhancement-mode device to ground and V_{CC} together with a depletion device to V_{CC} meeting the typical current sourcing requirements of the segments of an LED display. The sourcing devices are clamped to limit current flow. These devices may be turned off under program control (see Functional Description, EN Register), placing the output in a high-impedance state to provide required LED segment blanking for a multiplexed display. Available on L outputs only.

Note 1: When the driver is disabled, the depletion device may cause the output to settle down to an intermediate level between V_{CC} and GND. This voltage cannot be relied upon as a "1" level when reading the L inputs. The external signal must drive it to a "1" level.

Note 2: Much power is dissipated by this driver in driving an LED. Care must be taken to limit the power dissipation of the chip to within the absolute maximum ratings specified.

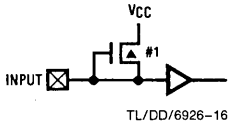
i. TRI-STATE Push-Pull—an enhancement-mode device to ground and V_{CC}. These outputs are TRI-STATE outputs, allowing for connection of these outputs to a data bus shared by other bus drivers. Available on L and R outputs only (in TRI-STATE mode on reset).

j. Push-Pull R—same as **f.**, but may be disabled. Available on R outputs only.

k. Additional depletion pull-up—a depletion load to V_{CC} with the same current sourcing capability as the input load **a.**, in addition to the output drive chosen. Available on L and R outputs only. *This device cannot be disabled*; therefore, open-drain outputs with "1" output and TRI-STATE outputs do not show high-impedance characteristics. This device is useful in applications where a pull-up with low source current is desired, e.g., reading keyboards and switches.

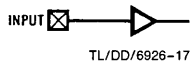
The above input and output configurations share common enhancement-mode and depletion-mode devices. Specifically, all configurations use one or more of six devices (numbered 1–6 respectively). Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figures 11* and *12* for each of these devices to allow the designer to effectively use these I/O configurations in designing a COP440 system.

Functional Description (Continued)



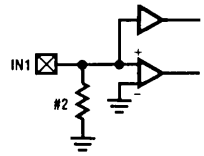
TL/DD/6926-16

a. Input with Load



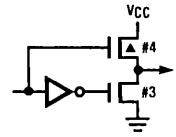
TL/DD/6926-17

b. Hi-Z Input



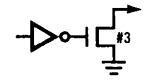
TL/DD/6926-18

c. Zero-Crossing Input



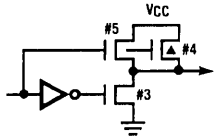
TL/DD/6926-19

d. Standard Output



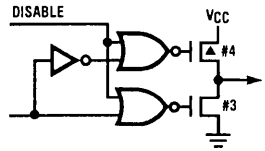
TL/DD/6926-20

e. Open-Drain Output



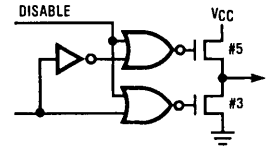
TL/DD/6926-21

f. Push-Pull Output



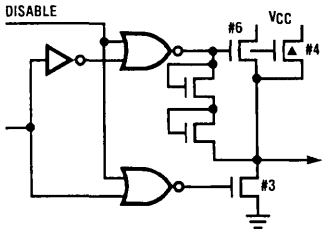
TL/DD/6926-22

g. Standard L, R Outputs



TL/DD/6926-23

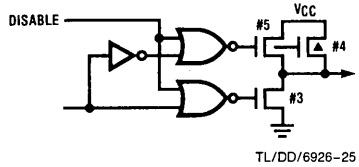
i. TRI-STATE Push-Pull (L, R) Outputs



TL/DD/6926-24

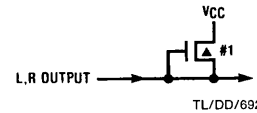
(▲ is depletion device)

h. LED (L) Outputs



TL/DD/6926-25

j. Push-Pull R Outputs



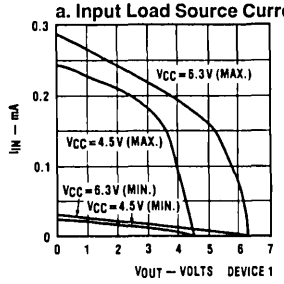
TL/DD/6926-26

k. Additional L, R Outputs Pull-Up

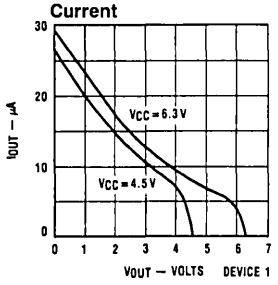
FIGURE 10. Input/Output Configurations

Typical Performance Characteristics

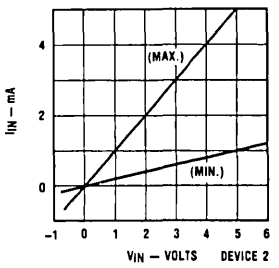
a. Input Load Source Current



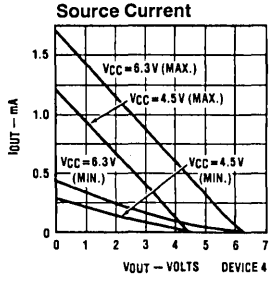
b. Input Load Minimum Source Current



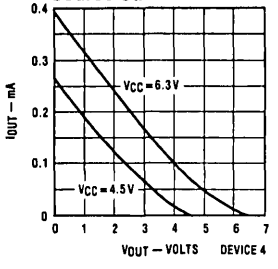
c. Zero-Crossing Detect Input Current



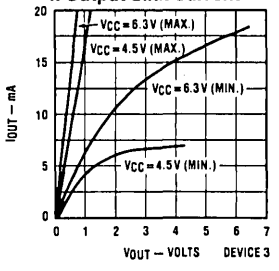
d. Standard Output Source Current



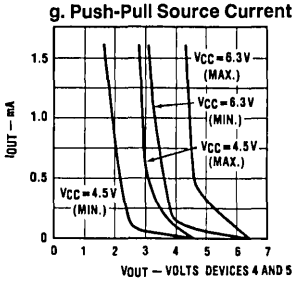
e. Standard Output Minimum Source Current



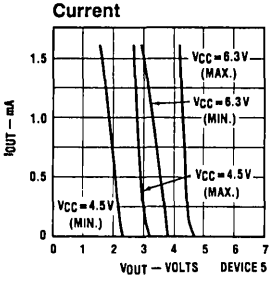
f. Output Sink Current



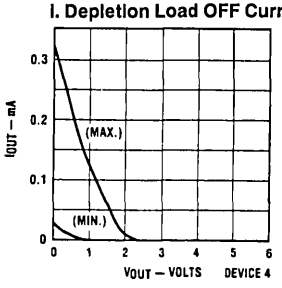
g. Push-Pull Source Current



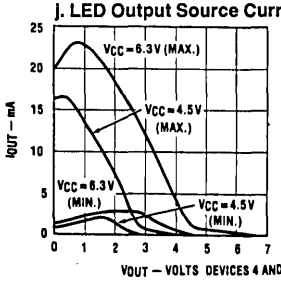
h. TRI-STATE Output Source Current



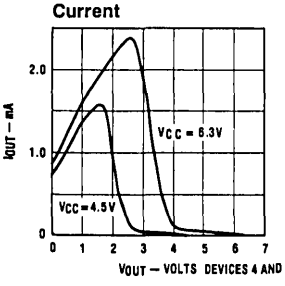
i. Depletion Load OFF Current



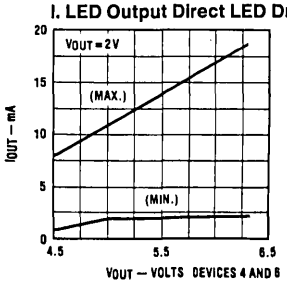
j. LED Output Source Current



k. LED Output Minimum Source Current



l. LED Output Direct LED Drive

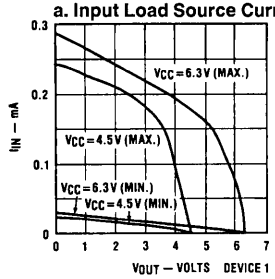


TL/DD/6926-27

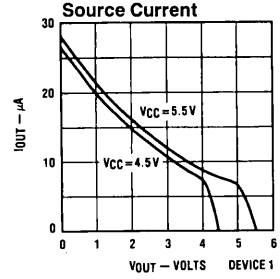
FIGURE 11. COP440/441/442 I/O Characteristics

Typical Performance Characteristics (Continued)

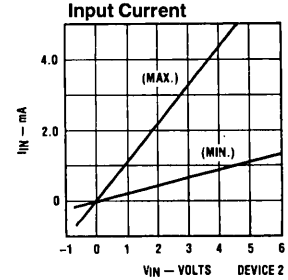
a. Input Load Source Current



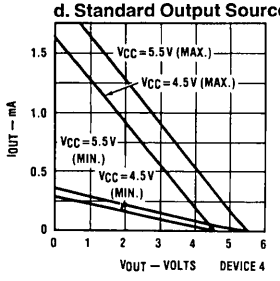
b. Input Load Minimum Source Current



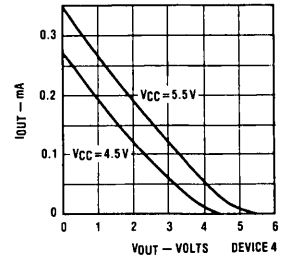
c. Zero-Crossing Detect Input Current



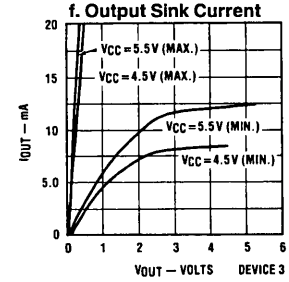
d. Standard Output Source Current



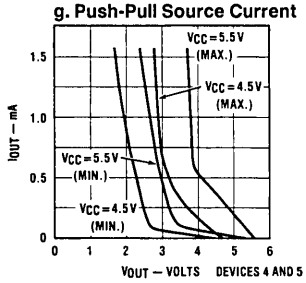
e. Standard Output Minimum Source Current



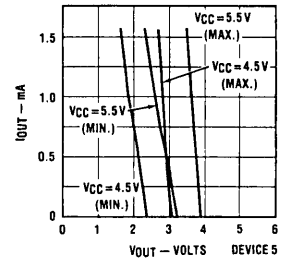
f. Output Sink Current



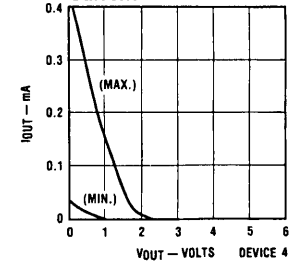
g. Push-Pull Source Current



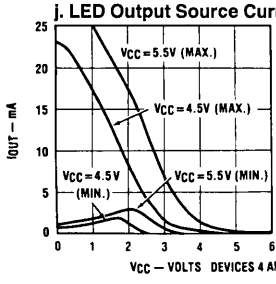
h. TRI-STATE Output Source Current



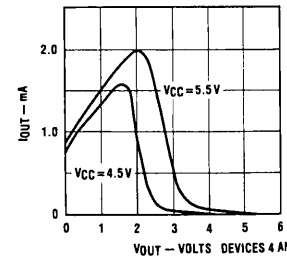
i. Depletion Load OFF Current



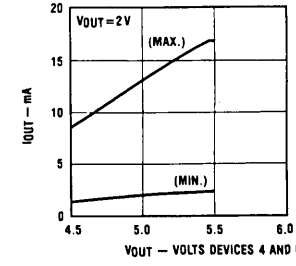
j. LED Output Source Current



k. LED Output Minimum Source Current



l. LED Output Direct LED Driver



TL/DD/6926–28

FIGURE 12. CCOP340/341/342 I/O Characteristics

Power Dissipation

In order not to damage the device by exceeding the absolute maximum power dissipation rating, the amount of power dissipated inside the chip must be carefully controlled. As an example, an application uses a COP440 in room temperature (25°C) environment with a V_{CC} power supply of 6V; IN and SI inputs have internal loads; G and D ports drive loads that may sink up to 2 mA into the chip; H port with standard output option reads switches; L port with the LED option drives a multiplexed seven-segment display; R, SO and SK drive MOS inputs that do not source or sink any current.

a. At 25°C, maximum power dissipation allowed = 750 mW

b. Power dissipation by chip except

$$I/O = I_{CC} \times V_{CC} = 35\ mA \times 6V = 210\ mW$$

c. Maximum power dissipation by IN,

$$SI = 5 \times 0.3\ mA \times 6V = 9\ mW$$

d. G and D ports are sinking current from external loads; maximum output voltage with 2 mA sink current is less than 0.4V. Power dissipation by G and D ports =

$$2\ mA \times 0.4V \times 8 = 6.4\ mW$$

e. Maximum power dissipation by H port =

$$4 \times 1.5\ mA \times 6V = 36\ mW$$

f. When the seven segments of the LED are turned on, the output voltage is about 2V, so that the segment current is 17 mA. Power dissipation by L port =

$$7 \times 17\ mA \times (6V - 2V) = 476\ mW$$

This power dissipation caused by driving LEDs is usually the highest among the various sources.

g. R, SO, and SK do not dissipate any significant amount of power because they do not need to source or sink any current.

Total power dissipation (TPD) inside the device is the sum of items b through g above.

$$TPD = 210 + 9 + 6 + 36 + 476\ mW = 737\ mW$$

This is within the 750 mW limit at room temperature. If this application has to operate at 70°C, then the power dissipation must be reduced to meet the limit at that temperature. Some ways to achieve this would be to limit the LED current or to use an external LED driver.

At 70°C the absolute maximum power dissipation rating drops to 400 mW. The user must be careful not to exceed this value.

COP440 SERIES DEVICES

If the COP440 is bonded as a 28- or 24-pin device, it becomes the COP441 or COP442, respectively, as illustrated in *Figure 3*. Note that the COP441 and COP442 do not include H and R ports. In addition, the COP442 does not include IN inputs; use of this option precludes the use of the IN options, the interrupt feature with IN as input, the zero-crossing detect option, IN_2 external event counter input, and the MICROBUS option. All other options are available.

COP340, COP341, and COP342 are extended temperature versions of the COP440, COP441, and COP442, respectively.

COP440 Series Instruction Set

Table II is a symbol table providing internal architecture, instruction operand and operation symbols used in the instruction set table.

Table III provides the mnemonic, operand, machine code, data flow, skip conditions and description associated with each instruction in the COP440 series instruction set.

TABLE II. COP440 Series Instruction Set Symbols

Symbol	Definition
INTERNAL ARCHITECTURE SYMBOLS	
A	4-bit Accumulator
B	8-bit RAM Address Register
Br	Upper 4 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry Register
D	4-bit Data Output Port
EN	8-bit Enable Register
G	4-bit Register to latch data for G I/O Port
H	4-bit Register to latch data for H I/O Port
IL	Two 1-bit Latches associated with the IN_3 or IN_0 Inputs
IN	4-bit Input Port
IN_1Z	Zero-Crossing Input
L	8-bit TRI-STATE I/O Port
M	4-bit contents of RAM Memory pointed to by B Register
N	2-bit subroutine return address stack pointer
PC	11-bit ROM Address Register (program counter)
Q	8-bit Register to latch data for L I/O Port
R	8-bit Register to latch data for R TRI-STATE I/O Port
SIO	4-bit Shift Register and Counter
SK	Logic-Controlled Clock Output
T	8-bit Binary Counter Register

Symbol	Definition
INSTRUCTION OPERAND SYMBOLS	
d	4-bit Operand Field, 0–15 binary (RAM Digit Select)
r	4-bit Operand Field, 0–9 binary (RAM Register Select)
a	11-bit Operand Field, 0–2047 binary (ROM Address)
y	4-bit Operand Field, 0–15 binary (Immediate Data)
RAM(s)	Content of RAM location addressed by s
RAM_N	Content of RAM location addressed by stack pointer N
ROM(t)	Content of ROM location addressed by t
OPERATIONAL SYMBOLS	
+	Plus
−	Minus
→	Replaces
←→	Is exchanged with
=	Is equal to
$\overline{A}$	The one's complement of A
⊕	Exclusive-OR
:	Range of values
V	OR

Instruction Set

TABLE III. COP440 Series Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC/LOGIC INSTRUCTIONS						
ASC		30	0011 | 0000	A + C + RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	0011 | 0001	A + RAM(B) → A	None	Add RAM to A
ADT		4A	0100 | 1010	A + 10_{10} → A	None	Add Ten to A
AISC	y	5–	0101 | y	A + y → A	Carry	Add immediate, Skip on Carry (y ≠ 0)
CASC		10	0001 | 0000	$\overline{A}$ + RAM(B) + C → A Carry → C	Carry	Complement and Add with Carry, Skip on Carry
CLRA		00	0000 | 0000	0 → A	None	Clear A
COMP		40	0100 | 0000	$\overline{A}$ → A	None	One's complement of A to A
NOP		44	0100 | 0100	None	None	No Operation
OR		33 1A	0011 | 0011 0001 | 1010	A ∨ M → A	None	OR RAM with A
RC		32	0011 | 0010	"0" → C	None	Reset C
SC		22	0010 | 0010	"1" → C	None	Set C
XOR		02	0000 | 0010	A ⊕ RAM(B) → A	None	Exclusive-OR RAM with A
TRANSFER OF CONTROL INSTRUCTIONS						
JID		FF	1111 | 1111	ROM ($PC_{10:8}$, A,M) → $PC_{7:0}$	None	Jump Indirect (Note 3)
JMP	a	6– – –	0110 | 0 | $a_{10:8}$ $a_{7:0}$	a → PC	None	Jump
JP	a	– –	1 | $a_{6:0}$ (pages 2,3 only) or	a → $PC_{6:0}$	None	Jump within Page (Note 4)
		– –	11 | $a_{5:0}$ (all other pages)	a → $PC_{5:0}$		
JSRP	a	– –	10 | $a_{5:0}$	PC + 1 → RAM_N N + 1 → N 00010 → $PC_{10:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 5)
JSR	a	6– – –	0110 | 1 | $a_{10:8}$ $a_{7:0}$	PC + 1 → RAM_N N + 1 → N a → PC	None	Jump to Subroutine
RET		48	0100 | 1000	N – 1 → N RAM_N → PC	None	Return from Subroutine
RETSK		49	0100 | 1001	N – 1 → N RAM_N → PC	Always Skip on Return	Return from Subroutine then Skip

Instruction Set (Continued)

TABLE III. COP440 Series Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
MEMORY REFERENCE INSTRUCTIONS						
CAME		33 1F	0011 | 0011 0001 | 1111	A → $EN_{7:4}$ RAM(B) → $EN_{3:0}$	None	Copy A, RAM to EN
CAMQ		33 3C	0011 | 0011 0011 | 1100	A → $Q_{7:4}$ RAM(B) → $Q_{3:0}$	None	Copy A, RAM to Q
CAMT		33 3F	0011 | 0011 0011 | 1111	A → $T_{7:4}$ RAM(B) → $T_{3:0}$	None	Copy A, RAM to T
CEMA		33 0F	0011 | 0011 0000 | 1111	$EN_{7:4}$ → RAM(B) $EN_{3:0}$ → A	None	Copy EN to RAM, A
CQMA		33 2C	0011 | 0011 0010 | 1100	$Q_{7:4}$ → RAM(B) $Q_{3:0}$ → A	None	Copy Q to RAM, A
CTMA		33 2F	0011 | 0011 0010 | 1111	$T_{7:4}$ → RAM(B) $T_{3:0}$ → A	None	Copy T to RAM, A
LD	r	–5	00 | r | 0101 r = 0:3	RAM(B) → A Br ⊕ r → Br	None	Load RAM into A, Exclusive-OR Br with r
LDD	r,d	23 – –	00 | 10 | 0011 0 | r | d r = 0:7	RAM(r,d) → A	None	Load A with RAM pointed to directly by r,d
LID		33 19	0011 | 0011 0001 | 1001	ROM ($PC_{10:8}$, A, M) → M, A	None	Load RAM, A Indirect
LQID		BF	1011 | 1111	ROM($PC_{10:8}$,A,M) → Q	None	Load Q Indirect (Note 3)
RMB	0 1 2 3	4C 45 42 43	0100 | 1100 0100 | 0101 0100 | 0010 0100 | 0011	0 → $RAM(B)_0$ 0 → $RAM(B)_1$ 0 → $RAM(B)_2$ 0 → $RAM(B)_3$	None	Reset RAM Bit
SMB	0 1 2 3	4D 47 46 4B	0100 | 1101 0100 | 0111 0100 | 0110 0100 | 1011	1 → $RAM(B)_0$ 1 → $RAM(B)_1$ 1 → $RAM(B)_2$ 1 → $RAM(B)_3$	None	Set RAM Bit
STII	y	7–	0111 | y	y → RAM(B) Bd + 1 → Bd	None	Store Memory Immediate and Increment Bd
X	r	–6	00 | r | 0110 r = 0:3	RAM(B) ⟷ A Br ⊕ r → Br	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	r,d	23 – –	0010 | 0011 1 | r | d r = 0:7	RAM(r,d) ⟷ A	None	Exchange A with RAM pointed to directly by r,d
XDS	r	–7	00 | r | 0111 r = 0:3	RAM(B) ⟷ A Bd – 1 → Bd Br ⊕ r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd, Exclusive-OR Br with r
XIS	r	–4	00 | r | 0100 r = 0:3	RAM(B) ⟷ A Bd + 1 → Bd Br ⊕ r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd, Exclusive-OR Br with r

Instruction Set (Continued)

TABLE III. COP440 Series Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	|0101|0000|	A → Bd	None	Copy A to Bd
CBA		4E	|0100|1110|	Bd → A	None	Copy Bd to A
LBI	r,d	--	|00|r|(d − 1)| r = 0:3, d = 0,9:15 or	r,d → B	Skip until not a LBI	Load B Immediate with r,d (Note 6)
		33	|0011|0011|			
		--	|1| r | d | r = 0:7, any d			
LEI	y	33	|0011|0011|	y → $EN_{3:0}$	None	Load lower half of EN Immediate
		6-	|0110| y |			
XABR		12	|0001|0010|	A ⟷ Br	None	Exchange A with Br
XAN		33	|0011|0011|	A ⟷ N(0,0 → A_3, A_2)	None	Exchange A with N
		0B	|0000|1011|			
TEST INSTRUCTIONS						
SKC		20	|0010|0000|		C = "1"	Skip if C is True
SKE		21	|0010|0001|		A = RAM(B)	Skip if A Equals RAM
SKGZ		33	|0011|0011|		$G_{3:0}$ = 0	Skip if G is Zero (all 4 bits)
		21	|0010|0001|			
SKGBZ		33	|0011|0011|	1st byte		Skip if G Bit is Zero
	0	01	|0000|0001|	2nd byte	G_0 = 0	
	1	11	|0001|0001|		G_1 = 0	
	2	03	|0000|0011|		G_2 = 0	
	3	13	|0001|0011|		G_3 = 0	
SKMBZ	0	01	|0000|0001|		$RAM(B)_0$ = 0	Skip if RAM Bit is Zero
	1	11	|0001|0001|		$RAM(B)_1$ = 0	
	2	03	|0000|0011|		$RAM(B)_2$ = 0	
	3	13	|0001|0011|		$RAM(B)_3$ = 0	
SKSZ		33	|0011|0011|		SIO = 0	Skip if SIO is Zero
		1C	|0001|1100|			
SKT		41	|0100|0001|		T counter carry has occurred since last test	Skip on Timer (Note 3)

Instruction Set (Continued)

TABLE III. COP440 Series Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
INPUT/OUTPUT INSTRUCTIONS						
CAMR		33 3D	0011 | 0011 0011 | 1101	A → $R_{7:4}$ RAM(B) → $R_{3:0}$	None	Output A, RAM to R Port
ING		33 2A	0011 | 0011 0010 | 1010	G → A	None	Input G Port to A
INH		33 2B	0011 | 0011 0010 | 1011	H → A	None	Input H Port to A
ININ		33 28	0011 | 0011 0010 | 1000	IN → A	None	Input IN Inputs to A (Note 2)
INIL		33 29	0011 | 0011 0010 | 1001	IL_3, CKO, IN_1Z, IL_0 → A	None	Input IL Latches to A (Note 3)
INL		33 2E	0011 | 0011 0010 | 1110	$L_{7:4}$ → RAM(B) $L_{3:0}$ → A	None	Input L Port to RAM, A
INR		33 2D	0011 | 0011 0010 | 1101	$R_{7:4}$ → RAM(B) $R_{3:0}$ → A	None	Input R Port to RAM,A
OBD		33 3E	0011 | 0011 0011 | 1110	Bd → D	None	Output Bd to D Port
OGI	y	33 5–	0011 | 0011 0101 | y	y → G	None	Output to G Port Immediate
OMG		33 3A	0011 | 0011 0011 | 1010	RAM(B) → G	None	Output RAM to G Port
OMH		33 3B	0011 | 0011 0011 | 1011	RAM(B) → H	None	Output RAM to H Port
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO (Note 3)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicitly defined (e.g., Br and Bd are explicitly defined). Bits are numbered 0 to N where 0 signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit A register.

Note 2: The ININ instruction is not available on the 24-pin COP442/COP342 since this device does not contain the IN inputs.

Note 3: For additional information on the operation of the XAS, JID, LQID, INIL, and SKT instructions, see below.

Note 4: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 5: A JSRP transfers program control to subroutine page 2 (00010 is loaded into the upper 5 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 6: LBI is a single-byte instruction if d = 0, 9, 10, 11, 12, 13, 14, or 15. The machine code for the lower 4 bits equals the binary value of the "d" data *minus 1*, e.g., to load the lower four bits of B (Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Description of Selected Instructions

The following information is provided to assist the user in understanding the operation of several unique instructions and to provide notes useful to programmers in writing COP440 programs.

XAS INSTRUCTION

XAS (Exchange A with SIO) exchanges the 4-bit contents of the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. An XAS instruction will also affect the SK output. (See Functional Description, EN register, above). If SIO is selected as a shift register, an XAS instruction must be performed once every 4 instruction cycles to effect a continuous data stream.

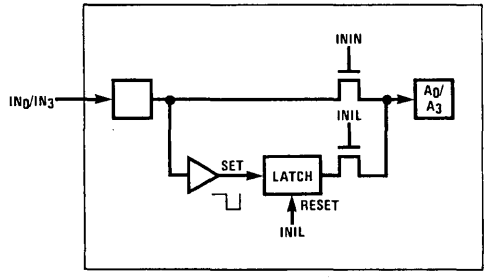
JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower 8 bits of the ROM address register PC with the *contents* of ROM addressed by the 11-bit word, $PC_{10:8}$, A, M. PC_{10}, PC_9 and PC_8 are not affected by this instruction.

Note that JID requires 2 instruction cycles if executed, 1 instruction cycle time if skipped.

INIL INSTRUCTION

INIL (Input IL Latches to A) inputs 2 latches, IL_3 and IL_0, CKO and IN_1 into A (see *Figure 13*). The IL_3 and IL_0 latches are set if a low-going pulse ("1" to "0") has occurred on the IN_3 and IN_0 inputs since the last INIL instruction, provided the input pulse stays low for at least two instruction cycles. Execution of an INIL inputs IL_3 and IL_0 into A3 and A0 respectively, and resets these latches to allow them to respond to subsequent low-going pulses on the IN_3 and IN_0 lines. If CKO is mask-programmed as a general purpose input, an INIL will input the state of CKO into A2. If CKO has not been so programmed, a "1" will be placed in A2. Unlike the COP420/420C/420L/444L series, INIL will input IN_1 into A1.



TL/DD/6926-29

FIGURE 13. INIL Hardware Implementation

If zero-crossing detect is selected, the IN_1 input will go through the detection logic, thus allowing the user to interrogate the input, sending a "1" if the input is above 0V and a "0" if it is below 0V. INIL is useful in recognizing pulses of short duration or pulses which occur too often to be read conveniently by an ININ instruction. It is also useful in checking the status of the zero-crossing detect input. The general purpose inputs IN_3–IN_0 are input to A upon execution of an ININ instruction, and the IN_1 input does not go through zero-crossing logic so that it has the same logic level as the other IN inputs for the ININ instruction (see *Figure 9*).

Note: IL latches are cleared on reset. This is different from the COP420/420C/420L/444L series.

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 11-bit word PC_{10}:PC_8, A, M. LQID can be used for table lookup or code conversion such as BCD to seven-segment. Note that LQID takes two instruction cycles if executed and one instruction cycle if skipped. Unlike most other COPS processors, this instruction does not push the stack.

LID INSTRUCTION

LID (Load Indirect) loads M and A with the contents of ROM pointed to by the 11-bit word PC_{10}:PC_8, A, M. Note that LID takes three instruction cycles if executed and two if skipped.

SKT INSTRUCTION

The SKT (Skip On Timer) instruction tests the state of the T counter (see internal logic, above) overflow latch, executing the next program instruction if the latch is not set. If the latch has been set since the previous test, the next program instruction is skipped and the latch is reset. The features associated with this instruction allow the processor to generate its own time-base for real-time processing, rather than relying on an external input signal.

INSTRUCTION SET NOTES

a. The first word of a COP440 program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, they are still fetched from program memory. Thus program paths take the same number of cycle times whether instructions are skipped or executed, except for LID, LQID, and JID.

c. The ROM is organized into 32 pages of 64 words each. The Program Counter is an 11-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID, LQID, or LID instruction is the last word of a page, the instruction operates as if it were in the next page. For example: a JP located in the last word of a page will jump to a location in the next page. Also, a LQID or JID located in the last word of page 3, 7, 11, 15, 19, 23, 27, or 31 will access data in the next group of four pages.

Option List

The COP440 mask-programmable options are assigned numbers which correspond with the COP440 pins.

Option 1: L_1 I/O Port (see note below)
- = 0: Standard output
- = 1: Open-drain output
- = 2: LED direct drive output
- = 3: TRI-STATE output
- = 4: same as 0 with extra load device to V_{CC}
- = 5: same as 1 with extra load device to V_{CC}
- = 6: same as 2 with extra load device to V_{CC}
- = 7: same as 3 with extra load device to V_{CC}

Option 2: L_0 I/O Port
(same as Option 1)

Option 3: SI Input
- = 0: Input with load device to V_{CC}
- = 1: Hi-Z Input

Option 4: SO Output
- = 0: Standard output
- = 1: Open-drain output
- = 2: Push-pull output

Option 5: SK Output
(same as Option 4)

Option 6: IN_0 Input
(same as Option 3)

Option 7: IN_3 Input
(same as Option 3)

Option 8: G_0 I/O Port
- = 0: Standard output
- = 1: Open-drain output

Option 9, G_1 I/O Port
(same as Option 8)

Option 10: G_2 I/O Port
(same as Option 8)

Option 11: G_3 I/O Port
(same as Option 8)

Option 12: H_0 I/O Port
(same as Option 8)

Option 13: H_1 I/O Port
(same as Option 8)

Option 14: H_2 I/O Port
(same as Option 8)

Option 15: H_3 I/O Port
(same as Option 8)

Option 16: D_3 Output
(same as Option 8)

Option 17: D_2 Output
(same as Option 8)

Option 18: D_1 Output
(same as Option 8)

Option 19: D_0 Output
(same as Option 8)

Option 20: GND—No options available

Option 21: CKO Pin
- = 0: Oscillator output
- = 1: RAM power supply (V_R) input
- = 2: General purpose input with load device to V_{CC}
- = 3: General purpose Hi-Z input

Option 22: CKI Input
- = 0: Crystal input divided by 16
- = 1: Crystal input divided by 8
- = 2: Single-pin RC controlled oscillator (÷4)
- = 3: Schmitt trigger clock input (÷4)

Option 23: $\overline{\text{RESET}}$ Input
(same as Option 3)

Option 24: R_7 I/O Port (see note below)
- = 0: Standard output
- = 1: Open-drain output
- = 2: Push-pull output
- = 3: TRI-STATE output
- = 4: same as 0 with extra load device to V_{CC}
- = 5: same as 1 with extra load device to V_{CC}
- = 6: same as 2 with extra load device to V_{CC}
- = 7: same as 3 with extra load device to V_{CC}

Option 25: R_6 I/O Port
(same as Option 24)

Option 26: R_5 I/O Port
(same as Option 24)

Option 27: R_4 I/O Port
(same as Option 24)

Option 28: R_3 I/O Port
(same as Option 24)

Option 29: R_2 I/O Port
(same as Option 24)

Option 30: R_1 I/O Port
(same as Option 24)

Option 31: R_0 I/O Port
(same as Option 24)

Option 32: L_7 I/O Port
(same as Option 1)

Option 33: L_6 I/O Port
(same as Option 1)

Option 34: L_5 I/O Port
(same as Option 1)

Option 35: L_4 I/O Port
(same as Option 1)

Option 36: IN_1 Input
- = 0: Input with load device to V_{CC}
- = 1: Hi-Z Input
- = 2: Zero-crossing detect input (Option 41 = 0)

Option 37: IN_2 Input
(same as Option 3)

Option 38: L_3 I/O Port
(same as Option 1)

Option 39: L_2 I/O Port
(same as Option 1)

Option 40: V_{CC}—no options available

Option List (Continued)

Option 41: COP Function
= 0: Normal
= 1: MICROBUS option

Option 42: IN Input Levels
= 0: Standard TTL input levels ("0" = 0.8V, "1" = 2.0V)
= 1: Higher voltage input levels ("0" = 1.2V, "1" = 3.6V)

Option 43: G Input Levels
(same as Option 42)

Option 44: L Input Levels
(same as Option 42)

Option 45: CKO Input Levels
(same as Option 42)

Option 46: SI Input Levels
(same as Option 42)

Option 47: R Input Levels
(same as Option 42)

Option 48: H Input Levels
(same as Option 42)

Option 49: No option available

Option 50: COP Bonding
= 0: COP440 (40-pin device)
= 1: COP441 (28-pin device)
= 2: COP442 (24-pin device)
= 3: COP440 and COP441
= 4: COP440 and COP442
= 5: COP440, COP441, and COP442
= 6: COP441 and COP442

COP440 Option Table

The following options information is to be sent to National along with the EPROM.

OPTION 1 VALUE = _____ IS: L_1 I/O PORT
OPTION 2 VALUE = _____ IS: L_0 I/O PORT
OPTION 3 VALUE = _____ IS: SI INPUT
OPTION 4 VALUE = _____ IS: SO OUTPUT
OPTION 5 VALUE = _____ IS: SK OUTPUT
OPTION 6 VALUE = _____ IS: IN_0 INPUT
OPTION 7 VALUE = _____ IS: IN_3 INPUT
OPTION 8 VALUE = _____ IS: G_0 I/O PORT
OPTION 9 VALUE = _____ IS: G_1 I/O PORT
OPTION 10 VALUE = _____ IS: G_2 I/O PORT
OPTION 11 VALUE = _____ IS: G_3 I/O PORT
OPTION 12 VALUE = _____ IS: H_0 I/O PORT
OPTION 13 VALUE = _____ IS: H_1 I/O PORT
OPTION 14 VALUE = _____ IS: H_2 I/O PORT
OPTION 15 VALUE = _____ IS: H_3 I/O PORT
OPTION 16 VALUE = _____ IS: D_3 OUTPUT
OPTION 17 VALUE = _____ IS: D_2 OUTPUT
OPTION 18 VALUE = _____ IS: D_1 OUTPUT
OPTION 19 VALUE = _____ IS: D_0 OUTPUT
OPTION 20 VALUE = _____ IS: GROUND PIN
OPTION 21 VALUE = _____ IS: CKO PIN
OPTION 22 VALUE = _____ IS: CKI INPUT
OPTION 23 VALUE = _____ IS: RESET INPUT
OPTION 24 VALUE = _____ IS: R_7 I/O PORT
OPTION 25 VALUE = _____ IS: R_6 I/O PORT
OPTION 26 VALUE = _____ IS: R_5 I/O PORT
OPTION 27 VALUE = _____ IS: R_4 I/O PORT
OPTION 28 VALUE = _____ IS: R_3 I/O PORT
OPTION 29 VALUE = _____ IS: R_2 I/O PORT
OPTION 30 VALUE = _____ IS: R_1 I/O PORT
OPTION 31 VALUE = _____ IS: R_0 I/O PORT
OPTION 32 VALUE = _____ IS: L_7 I/O PORT
OPTION 33 VALUE = _____ IS: L_6 I/O PORT
OPTION 34 VALUE = _____ IS: L_5 I/O PORT
OPTION 35 VALUE = _____ IS: L_4 I/O PORT
OPTION 36 VALUE = _____ IS: IN_1 INPUT
OPTION 37 VALUE = _____ IS: IN_2 INPUT
OPTION 38 VALUE = _____ IS: L_3 I/O PORT
OPTION 39 VALUE = _____ IS: L_2 I/O PORT
OPTION 40 VALUE = _____ IS: V_{CC}
OPTION 41 VALUE = _____ IS: COP FUNCTION
OPTION 42 VALUE = _____ IS: IN INPUT LEVELS
OPTION 43 VALUE = _____ IS: G INPUT LEVELS
OPTION 44 VALUE = _____ IS: L INPUT LEVELS
OPTION 45 VALUE = _____ IS: CKO INPUT LEVELS
OPTION 46 VALUE = _____ IS: SI INPUT LEVELS
OPTION 47 VALUE = _____ IS: R INPUT LEVELS
OPTION 48 VALUE = _____ IS: H INPUT LEVELS
OPTION 49 VALUE = _____ IS: NO OPTION
OPTION 50 VALUE = _____ IS: COP BONDING

Note on L and R I/O Port Options

If L and R I/O Ports are used as inputs, the following must be observed:

a. Open-Drain output (selection 1) is allowed only if external pull-up is provided.

b. If L and R output ports are disabled when reading, an external pull-up is required unless selections 4, 5, 6, or 7 are chosen.

c. If L output port is enabled, selections 3 and 7 are not allowed.

d. If R output port is enabled, selections 2, 3, 6, and 7 are not allowed.

Test Mode (Non-Standard Operation)

The SO output has been configured to provide for standard test procedures for the custom-programmed COP440. With SO forced to logic "1", two test modes are provided, depending upon the value of SI:

a. RAM and Internal Logic Test Mode (SI = 1)

b. ROM Test Mode (SI = 0)

These special test modes should not be employed by the user; they are intended for manufacturing test only.

National Semiconductor Corporation

COP444L/COP445L/COP344L/COP345L Single-Chip N-Channel Microcontrollers

General Description

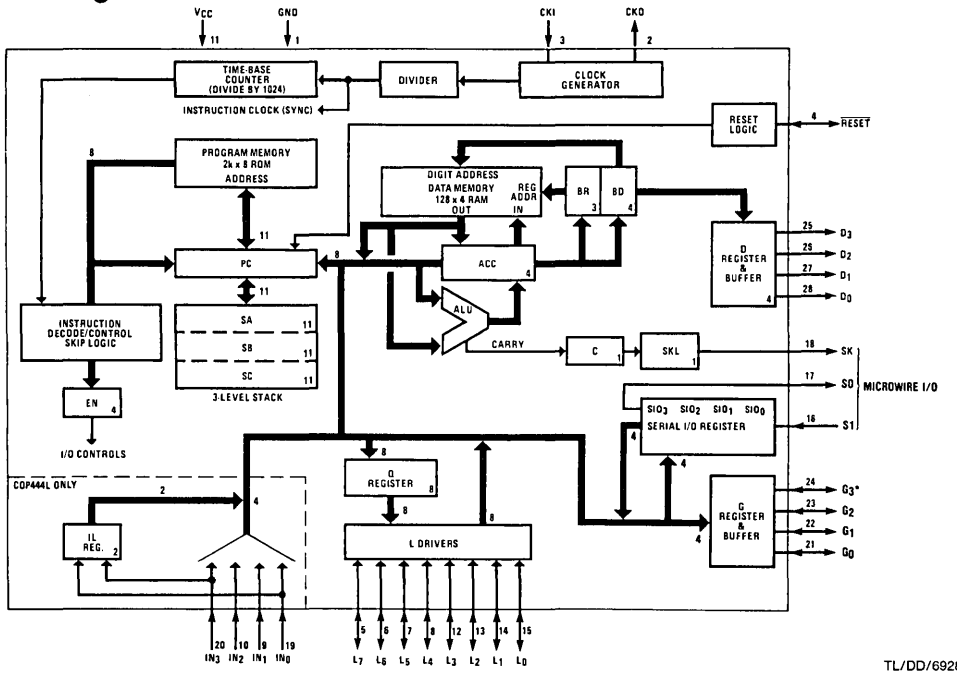
The COP444L, COP445L, COP344L, and COP345L Single-Chip N-Channel Microcontrollers are members of the COPS™ family, fabricated using N-channel, silicon gate MOS technology. These controller oriented processors are complete microcomputers containing all system timing, internal logic, ROM, RAM, and I/O necessary to implement dedicated control functions in a variety of applications. Features include single supply operation, a variety of output configuration options, with an instruction set, internal architecture and I/O scheme designed to facilitate keyboard input, display output and BCD data manipulation. The COP445L is identical to the COP444L, but with 19 I/O lines instead of 23. They are an appropriate choice for use in numerous human interface control environments. Standard test procedures and reliable high-density fabrication techniques provide the medium to large volume customers with a customized controller oriented processor at a low end-product cost.

The COP344L and COP345L are exact functional equivalents, but extended temperature range versions of the COP444L and COP445L respectively.

Features

- Low cost
- Powerful instruction set
- 2k x 8 ROM, 128 x 4 RAM
- 23 I/O lines (COP444L)
- True vectored interrupt, plus restart
- Three-level subroutine stack
- 15 μs instruction time
- Single supply operation (4.5–6.3V)
- Low current drain (11 mA max.)
- Internal time-base counter for real-time processing
- Internal binary counter register with MICROWIRE™ serial I/O capability
- General purpose and TRI-STATE® outputs
- LSTTL/CMOS compatible in and out
- Direct drive of LED digit and segment lines
- Software/hardware compatible with other members of COP400 family
- Extended temperature range devices COP344L/COP345L (−40°C to +85°C)
- Wider supply range (4.5–9.5V) optionally available

Block Diagram



TL/DD/6928–1

FIGURE 1

COP444L/COP445L

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin Relative to GND	−0.5V to +10V
Ambient Operating Temperature	0°C to +70°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 seconds)	300°C
Power Dissipation	0.75 Watt at 25°C
	0.4 Watt at 70°C
Total Source Current	120 mA
Total Sink current	120 mA

Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolulte maximum ratings.

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 9.5V$ unless otherwise noted.

Parameter	Conditions	Min	Max	Units
Standard Operating Voltage (V_{CC})	(Note 1)	4.5	6.3	V
Optional Operating Voltage (V_{CC})		4.5	9.5	V
Power Supply Ripple	Peak to Peak		0.5	V
Operating Supply Current	All Inputs and Outputs Open		13	mA
Input Voltage Levels				
CKI Input Levels				
Crystal Input (÷32, ÷16, ÷8)				
Logic High (V_{IH})	V_{CC} = Max.	3.0		V
Logic High (V_{IH})	V_{CC} = 5V ±5%	2.0	0.4	V
Logic Low (V_{IL})		−0.3		
Schmitt Trigger Input (÷4)				
Logic High (V_{IH})		0.7 V_{CC}		V
Logic Low (V_{IL})		−0.3	0.6	V
$\overline{\text{RESET}}$ Input Levels	Schmitt Trigger Input			
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.6	V
SO Input Level (Test Mode)	(Note 3)	2.0	2.5	V
All Other Inputs				
Logic High	V_{CC} = Max.	3.0		V
Logic High	With TTL Trip Level Options	2.0		V
Logic Low	Selected, V_{CC} = 5V ±10%	−0.3	0.8	V
Logic High	With High Trip Level Options	3.6		V
Logic Low	Selected	−0.3	1.2	V
Input Capacitance			7	pF
Hi-Z Input Leakage		−1	+1	µA
Output Voltage Levels				
LSTTL Operation	V_{CC} = 5V ±5%			
Logic High (V_{OH})	I_{OH} = −25 µA	2.7		V
Logic Low (V_{OL})	I_{OL} = 0.36 mA		0.4	V
CMOS Operation (Note 2)				
Logic High	I_{OH} = −10 µA	V_{CC}−1		V
Logic Low	I_{OL} = +10µA		0.2	V

Note 1: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Note 2: TRI-STATE and LED configurations are excluded.

Note 3: SO output "0" level must be less than 0.8V for normal operation.

COP444L/COP445L (Continued)

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 9.5V$ unless otherwise noted. (Continued)

Parameter	Conditions	Min	Max	Units
Output Current Levels				
Output Sink Current				
SO and SK Outputs (I_{OL})	$V_{CC} = 9.5V, V_{OL} = 0.4V$	1.8		mA
	$V_{CC} = 6.3V, V_{OL} = 0.4V$	1.2		mA
	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.9		mA
L_0–L_7 Outputs and Standard	$V_{CC} = 9.5V, V_{OL} = 0.4V$	0.4		mA
G_0–G_3, D_0–D_3 Outputs (I_{OL})	$V_{CC} = 6.3V, V_{OL} = 0.4V$	0.4		mA
	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.4		mA
G_0–G_3 and D_0–D_3 Outputs with	$V_{CC} = 9.5V, V_{OL} = 1.0V$	15		mA
High Current Options (I_{OL})	$V_{CC} = 6.3V, V_{OL} = 1.0V$	11		mA
	$V_{CC} = 4.5V, V_{OL} = 1.0V$	7.5		mA
G_0–G_3 and D_0–D_3 Outputs with	$V_{CC} = 9.5V, V_{OL} = 1.0V$	30		mA
Very High Current Options (I_{OL})	$V_{CC} = 6.3V, V_{OL} = 1.0V$	22		mA
	$V_{CC} = 4.5V, V_{OL} = 1.0V$	15		mA
CKI (Single-pin RC oscillator)	$V_{CC} = 4.5V, V_{IH} = 3.5V$	2		mA
CKO	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.2		mA
Output Source Current				
Standard Configuration,	$V_{CC} = 9.5V, V_{OH} = 2.0V$	−140	−800	μA
All Outputs (I_{OH})	$V_{CC} = 6.3V, V_{OH} = 2.0V$	−75	−480	μA
	$V_{CC} = 4.5V, V_{OH} = 2.0V$	−30	−250	μA
Push-Pull Configuration	$V_{CC} = 9.5V, V_{OH} = 4.75V$	−1.4		mA
SO and SK Outputs (I_{OH})	$V_{CC} = 6.3V, V_{OH} = 2.4V$	−1.4		mA
	$V_{CC} = 4.5V, V_{OH} = 1.0V$	−1.2		mA
LED Configuration, L_0–L_7				
Outputs, Low Current	$V_{CC} = 9.5V, V_{OH} = 2.0V$	−1.5	−18	mA
Drivers Option (I_{OH})	$V_{CC} = 6.0V, V_{OH} = 2.0V$	−1.5	−13	mA
LED Configuration, L_0–L_7				
Outputs, High Current	$V_{CC} = 9.5V, V_{OH} = 2.0V$	−3.0	−35	mA
Driver Option (I_{OH})	$V_{CC} = 6.0V, V_{OH} = 2.0V$	−3.0	−25	mA
TRI-STATE Configuration,	$V_{CC} = 9.5V, V_{OH} = 5.5V$	−0.75		mA
L_0–L_7 Outputs, Low	$V_{CC} = 6.3V, V_{OH} = 3.2V$	−0.8		mA
Current Driver Option (I_{OH})	$V_{CC} = 4.5V, V_{OH} = 1.5V$	−0.9		mA
TRI-STATE Configuration,	$V_{CC} = 9.5V, V_{OH} = 5.5V$	−1.5		mA
L_0–L_7 Outputs, High	$V_{CC} = 6.3V, V_{OH} = 3.2V$	−1.6		mA
Current Driver Option (I_{OH})	$V_{CC} = 4.5V, V_{OH} = 1.5V$	−1.8		mA
Input Load Source Current	$V_{CC} = 5.0V, V_{IL} = 0V$	−10	−140	μA
CKO Output				
RAM Power Supply Option				
Power Requirement	$V_R = 3.3V$		3.0	mA
TRI-STATE Output Leakage Current		−2.5	+2.5	μA
Total Sink Current Allowed				
All Outputs Combined			120	mA
D, G Ports			120	mA
L_7–L_4			4	mA
L_3–L_0			4	mA
All Other Pins			1.5	mA
Total Source Current Allowed				
All I/O Combined			120	mA
L_7–L_4			60	mA
L_{3-L0}			60	mA
Each L Pin			30	mA
All Other Pins			1.5	mA

1

COP344L/COP345L

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin Relative to GND	−0.5V to +10V
Ambient Operating Temperature	−40°C to +85°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 seconds)	300°C
Power Dissipation	0.75 Watt at 25°C 0.25 Watt at 85°C
Total Source Current	120 mA
Total Sink Current	120 mA

Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.

DC Electrical Characteristics $-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 7.5V$ unless otherwise noted.

Parameter	Conditions	Min	Max	Units
Standard Operating Voltage (V_{CC})	(Note 1)	4.5	5.5	V
Optional Operating Voltage (V_{CC})		4.5	7.5	V
Power Supply Ripple	Peak to Peak		0.5	V
Operating Supply Current	All Inputs and Outputs Open		15	mA
Input Voltage Levels				
CKI Input Levels				
Crystal Input				
Logic High (V_{IH})	V_{CC} = Max.	3.0		V
Logic High (V_{IH})	V_{CC} = 5V ±5%	2.2	0.3	V
Logic Low (V_{IL})		−0.3		
Schmitt Trigger Input				
Logic High (V_{IH})		0.7 V_{CC}		V
Logic Low (V_{IL})		−0.3	0.4	V
$\overline{\text{RESET}}$ Input Levels	Schmitt Trigger Input			
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.4	V
SO Input Level (Test Mode)		2.2	2.5	V
All Other Inputs				
Logic High	V_{CC} = Max.	3.0		V
Logic High	With TTL Trip Level Options	2.2		V
Logic Low	Selected, V_{CC} = 5V ±5%	−0.3	0.6	V
Logic High	With High Trip Level Options	3.6		V
Logic Low	Selected	−0.3	1.2	V
Input Capacitance			7	pF
Hi-Z Input Leakage		−2	+2	µA
Output Voltage Levels				
LSTTL Operation	V_{CC} = 5V ±10%			
Logic High (V_{OH})	I_{OH} = −20 µA	2.7		V
Logic Low (V_{OL})	I_{OL} = 0.36 mA		0.4	V
CMOS Operation (Note 2)				
Logic High	I_{OH} = −10 µA	V_{CC}−1		V
Logic Low	I_{OL} = +10 µA		0.2	V

Note 1: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Note 2: TRI-STATE and LED configurations are excluded.

Note 3: SO output "0" level must be less than 0.6V for normal operation.

COP344L/COP345L (Continued)

DC Electrical Characteristics

$-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 7.5V$ unless otherwise noted. (Continued)

Parameter	Conditions	Min	Max	Units
Output Current Levels				
Output Sink Current				
SO and SK Outputs (I_{OL})	$V_{CC} = 7.5V, V_{OL} = 0.4V$	1.4		mA
	$V_{CC} = 5.5V, V_{OL} = 0.4V$	1.0		mA
	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.8		mA
L_0–L_7 Outputs, and Standard	$V_{CC} = 7.5V, V_{OL} = 0.4V$	0.4		mA
G_0–G_3, D_0–D_3 Outputs (I_{OL})	$V_{CC} = 5.5V, V_{OL} = 0.4V$	0.4		mA
	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.4		mA
G_0–G_3 and D_0–D_3 Outputs with	$V_{CC} = 7.5V, V_{OL} = 1.0V$	12		mA
High Current Options (I_{OL})	$V_{CC} = 5.5V, V_{OL} = 1.0V$	9		mA
	$V_{CC} = 4.5V, V_{OL} = 1.0V$	7		mA
G_0–G_3 and D_0–D_3 Outputs with	$V_{CC} = 7.5V, V_{OL} = 1.0V$	24		mA
Very High Current Options (I_{OL})	$V_{CC} = 5.5V, V_{OL} = 1.0V$	18		mA
	$V_{CC} = 4.5V, V_{OL} = 1.0V$	14		mA
CKI (Single-Pin RC Oscillator)	$V_{CC} = 4.5V, V_{IH} = 3.5V$	2		mA
CKO	$V_{CC} = 4.5V, V_{OL} = 0.4V$	0.2		mA
Output Source Current				
Standard Configuration,	$V_{CC} = 7.5V, V_{OH} = 2.0V$	−100	−900	μA
All Outputs (I_{OH})	$V_{CC} = 5.5V, V_{OH} = 2.0V$	−55	−600	μA
	$V_{CC} = 4.5V, V_{OH} = 2.0V$	−28	−350	μA
Push-Pull Configuration	$V_{CC} = 7.5V, V_{OH} = 3.75V$	−0.85		mA
SO and SK Outputs (I_{OH})	$V_{CC} = 5.5V, V_{OH} = 2.0V$	−1.1		mA
	$V_{CC} = 4.5V, V_{OH} = 1.0V$	−1.2		mA
LED Configuration, L_0–L_7	$V_{CC} = 7.5V, V_{OH} = 2.0V$	−1.4	−27	mA
Outputs, Low Current	$V_{CC} = 6.0V, V_{OH} = 2.0V$	−1.4	−17	mA
Driver Option(I_{OH})	$V_{CC} = 5.5V, V_{OH} = 2.0V$	−0.7	−15	mA
LED Configuration, L_0–L_7	$V_{CC} = 7.5V, V_{OH} = 2.0V$	−2.7	−54	mA
Outputs, High Current	$V_{CC} = 6.0V, V_{OH} = 2.0V$	−2.7	−34	mA
Driver Option (I_{OH})	$V_{CC} = 5.5V, V_{OH} = 2.0V$	−1.4	−30	mA
TRI-STATE Configuration,	$V_{CC} = 7.5V, V_{OH} = 4.0V$	−0.7		mA
L_0–L_7 Outputs, Low	$V_{CC} = 5.5V, V_{OH} = 2.7V$	−0.6		mA
Current Driver Option (I_{OH})	$V_{CC} = 4.5V, V_{OH} = 1.5V$	−0.9		mA
TRI-STATE Configuration,	$V_{CC} = 7.5V, V_{OH} = 4.0V$	−1.4		mA
L_0–L_7 Outputs, High	$V_{CC} = 5.5V, V_{OH} = 2.7V$	−1.2		mA
Current Driver Option (I_{OH})	$V_{CC} = 4.5V, V_{OH} = 1.5V$	−1.8		mA
Input Load Source Current	$V_{CC} = 5.0V, V_{IL} = 0V$	−10	−200	μA
CKO Output				
RAM Power Supply Option	$V_R = 3.3V$		4.0	mA
Power Requirement				
TRI-STATE Output Leakage Current		−5	+5	μA
Total Sink Current Allowed				
All Outputs Combined			120	mA
D, G Ports			120	mA
L_7–L_4			4	mA
L_3–L_0			4	mA
All Other Pins			1.5	mA
Total Source Current Allowed				
All I/O Combined			120	mA
L_7–L_4			60	mA
L_3–L_0			60	mA
Each L Pin			30	mA
All Other Pins			1.5	mA

AC Electrical Characteristics

COP444L/445L: $0°C \leq T_A \leq 70°C$, $4.5V \leq V_{CC} \leq 9.5V$ unless otherwise noted.

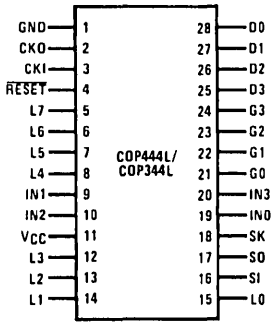
COP344L/345L: $-40°C \leq T_A \leq +85°C$, $4.5V \leq V_{CC} \leq 7.5V$ unless otherwise noted.

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time—t_C		16	40	μs
CKI				
Input Frequency—f_I	÷32 Mode	0.8	2.0	MHz
	÷16 Mode	0.4	1.0	MHz
	÷8 Mode	0.2	0.5	MHz
	÷4 Mode	0.1	0.25	MHz
Duty Cycle		30	60	%
Rise Time	$f_I = 2$ MHz		120	ns
Fall Time			80	ns
CKI Using RC (÷4)	R = 56 kΩ ±5% C = 100 pF ±10%			
Instruction Cycle Time (Note 1)		16	28	μs
CKO as SYNC Input				
t_{SYNC}		400		ns
INPUTS:				
IN_3–IN_0, G_3–G_0, L_7–L_0				
t_{SETUP}		8.0		μs
t_{HOLD}		1.3		μs
SI				
t_{SETUP}		2.0		μs
t_{HOLD}		1.0		μs
OUTPUT PROPAGATION DELAY	Test Condition: $C_L = 50$ pF, $R_L = 20$ kΩ, $V_{OUT} = 1.5V$			
SO, SK Outputs				
t_{pd1}, t_{pd0}			4.0	μs
All Other Outputs				
t_{pd1}, t_{pd0}			5.6	μs

Note 1: Variation due to the device included.

Connection Diagrams

Dual-In-Line

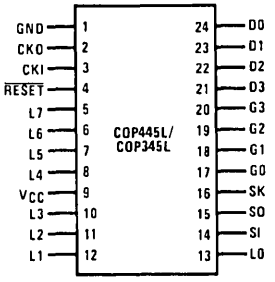


TL/DD/6928-2

Top View

Order Number COP444L/N or COP344L/N
See NS Package Number N28B

Dual-In-Line



TL/DD/6928-3

Top View

Order Number COP445L/N, COP445L/M, COP345L/N or COP345L/M
See NS Package Number N24A or M24B

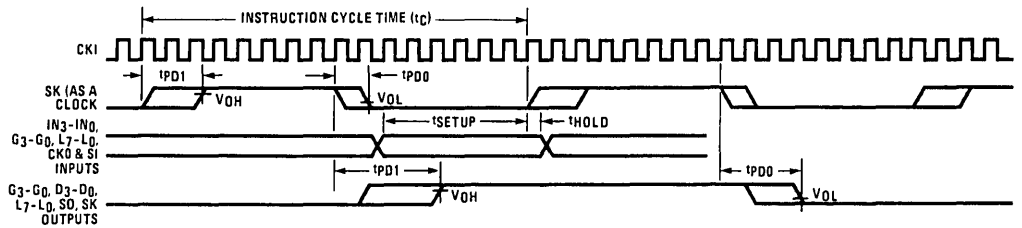
FIGURE 2

Pin Descriptions

Pin	Description
L_7–L_0	8 bidirectional I/O ports with TRI-STATE
G_3–G_0	4 bidirectional I/O ports
D_3–D_0	4 general purpose outputs
IN_3–IN_0	4 general purpose inputs (COP444L only)
SI	Serial input (or counter input)
SO	Serial output (or general purpose output)
SK	Logic-controlled clock (or general purpose output)

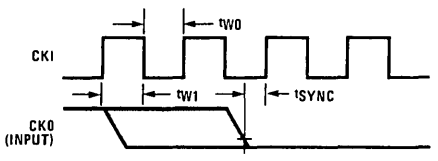
Pin	Description
CKI	System oscillator input
CKO	System oscillator output (or general purpose input, RAM power supply, or SYNC input)
$\overline{RESET}$	System reset input
V_{CC}	Power supply
GND	Ground

Timing Diagrams



TL/DD/6928-4

FIGURE 3a. Input/Output Timing Diagrams (Crystal Divide-by-16 Mode)



TL/DD/6928-5

FIGURE 3b. Synchronization Timing

Functional Description

A block diagram of the COP444L is given in *Figure 1.* Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1" (greater than 2 volts). When a bit is reset, it is a logic "0" (less than 0.8 volts).

All functional references to the COP444L/COP445L also apply to the COP344L/COP345L.

PROGRAM MEMORY

Program Memory consists of a 2048 byte ROM. As can be seen by an examination of the COP444L/445L instruction set, these words may be program instructions, program data or ROM addressing data. Because of the special characteristics associated with the JP, JSRP, JID, and LQID instructions, ROM must often be thought of as being organized into 32 pages of 64 words each.

ROM addressing is accomplished by a 11-bit PC register. Its binary value selects one of the 2048 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 11-bit binary count value. Three levels of subroutine nesting are implemented by the 11-bit subroutine save registers, SA, SB, and SC, providing a last-in, first-out (LIFO) hardware subroutine stack.

ROM instruction words are fetched, decoded and executed by the Instruction Decode, Control and Skip Logic circuitry.

DATA MEMORY

Data memory consists of a 512-bit RAM, organized as 8 data registers of 16 4-bit digits. RAM addressing is implemented by a 7-bit B register whose upper 3 bits (Br) select 1 of 8 data registers and lower 4 bits (Bd) select 1 of 16 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) is usually loaded into or from, or exchanged with, the A register (accumulator), it may also be loaded into or from the Q latches or loaded from the L ports. RAM addressing may also be performed directly by the LDD and XAD instructions based upon the 7-bit contents of the operand field of these instructions. The Bd register also serves as a source register for 4-bit data sent directly to the D outputs.

INTERNAL LOGIC

The 4-bit A register (accumulator) is the source and destination register for most I/O, arithmetic, logic and data memory access operations. It can also be used to load the Br and Bd portions of the B register, to load and input 4 bits of the 8-bit Q latch data, to input 4 bits of the 8-bit L I/O port data and to perform data exchanges with the SIO register.

A 4-bit adder performs the arithmetic and logic functions, storing its results in A. It also outputs a carry bit to the 1-bit C register, most often employed to indicate arithmetic overflow. The C register, in conjunction with the XAS instruction and the EN register, also serves to control the SK output. C can be outputted directly to SK or can enable SK to be a sync clock each instruction cycle time. (See XAS instruction and EN register descriptor, below.)

Four general-purpose inputs, IN_3–IN_0, are provided.

The D register provides 4 general-purpose outputs and is used as the destination register for the 4-bit contents of Bd. The D outputs can be directly connected to the digits of a multiplexed LED display.

The G register contents are outputs to 4 general-purpose bidirectional I/O ports. G I/O ports can be directly connected to the digits of a multiplexed LED display.

The Q register is an internal, latched, 8-bit register, used to hold data loaded to or from M and A, as well as 8-bit data from ROM. Its contents are output to the L I/O ports when the L drivers are enabled under program control. (See LEI instruction.)

The 8 L drivers, when enabled, output the contents of latched Q data to the L I/O ports. Also, the contents of L may be read directly into A and M. L I/O ports can be directly connected to the segments of a multiplexed LED display (using the LED Direct Drive output configuration option) with Q data being outputted to the Sa–Sg and decimal point segments of the display.

The SIO register functions as a 4-bit serial-in/serial-out shift register or as a binary counter depending on the contents of the EN register. (See EN register description, below.) Its contents can be exchanged with A, allowing it to input or output a continuous serial data stream. SIO may also be used to provide additional parallel I/O by connecting SO to external serial-in/parallel-out shift registers.

The XAS instruction copies C into the SKL latch. In the counter mode, SK is the output of SKL; in the shift register mode, SK outputs SKL ANDed with the clock.

The EN register is an internal 4-bit register loaded under program control by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register (EN_3–EN_0).

1. The least significant bit of the enable register, EN_0, selects the SIO register as either a 4-bit shift register or a 4-bit binary counter. With EN_0 set, SIO is an asynchronous binary counter, *decrementing* its value by one upon each low-going pulse ("1" to "0") occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output is equal to the value of EN_3. With EN_0 reset, SIO is a serial shift register shifting left each instruction cycle time. The data present at SI goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. (See 4 below.) The SK output becomes a logic-controlled clock.
2. With EN_1 set the IN_1 input is enabled as an interrupt input. Immediately following an interrupt, EN_1 is reset to disable further interrupts.
3. With EN_2 set, the L drivers are enabled to output the data in Q to the L I/O ports. Resetting EN_2 disables the L drivers, placing the L I/O ports in a high-impedance input state.
4. EN_3, in conjunction with EN_0, affects the SO output. With EN_0 set (binary counter option selected) SO will output the value loaded into EN_3. With EN_0 reset (serial shift register option selected), setting EN_3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN_3 with the serial shift register option selected disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains reset to "0". The table below provides a summary of the modes associated with EN_3 and EN_0.

Functional Description (Continued)

Enable Register Modes—Bits EN_3 and EN_0

EN_3	EN_0	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL = 1, SK = CLOCK If SKL = 0, SK = 0
1	0	Shift Register	Input to Shift Register	Serial Out	If SKL = 1, SK = CLOCK If SKL = 0, SK = 0
0	1	Binary Counter	Input to Binary Counter	0	If SKL = 1, SK = 1 If SKL = 0, SK = 0
1	0	Binary Counter	Input to Binary Counter	1	If SKL = 1, SK = 1 If SKL = 0, SK = 0

INTERRUPT

The following features are associated with the IN_1 interrupt procedure and protocol and must be considered by the programmer when utilizing interrupts.

a. The interrupt, once acknowledged as explained below, pushes the next sequential program counter address (PC+1) onto the stack, pushing in turn the contents of the other subroutine-save registers to the next lower level (PC + 1 → SA → SB → SC). Any previous contents of SC are lost. The program counter is set to hex address 0FF (the last word of page 3) and EN_1 is reset.

b. An interrupt will be acknowledged only after the following conditions are met:

1. EN_1 has been set.
2. A low-going pulse ("1" to "0") at least two instruction cycles wide occurs on the IN_1 input.
3. A currently executing instruction has been completed
4. All successive transfer of control instructions and successive LBIs have been completed (e.g., if the main program is executing a JP instruction which transfers program control to another JP instruction, the interrupt will not be acknowledged until the second JP instruction has been executed.

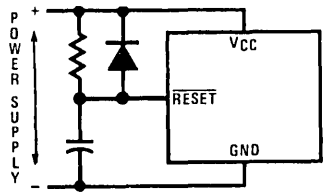
c. Upon acknowledgement of an interrupt, the skip logic status is saved and later restored upon popping of the stack. For example, if an interrupt occurs during the execution of ASC (Add with Carry, Skip on Carry) instruction which results in carry, the skip logic status is saved and program control is transferred to the interrupt servicing routine at hex address 0FF. At the end of the interrupt routine, a RET instruction is executed to "pop" the stack and return program control to the instruction following the original ASC. *At this time*, the skip logic is enabled and skips this instruction because of the previous ASC carry. Subroutines and LQID instructions should not be nested within the interrupt service routine, since their popping the stack will enable any previously saved main program skips, interfering with the orderly execution of the interrupt routine.

d. The first instruction of the interrupt routine at hex address 0FF must be a NOP.

e. A LEI instruction can be put immediately before the RET to re-enable interrupts.

INITIALIZATION

The Reset Logic will initialize (clear) the device upon power-up if the power supply rise time is less than 1 ms and greater than 1 μs. If the power supply rise time is greater than 1 ms, the user use provide an external RC network and diode to the $\overline{RESET}$ pin as shown below. If the RC network is not used, the $\overline{RESET}$ pin must be pulled up to V_{CC} either by the internal load or by an external resistor (≥40 kΩ) to V_{CC}. The $\overline{RESET}$ pin is configured as a Schmitt trigger input. Initialization will occur whenever a logic "0" is applied to the $\overline{RESET}$ input, provided it stays low for at least three instruction cycle times.



TL/DD/6928–6

RC ≥ 5 x Power Supply Rise Time (R ≥ 40k)

Power-Up Clear Circuit

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN, and G registers are cleared. The SK output is enabled as a SYNC output, providing a pulse each instruction cycle time. *Data Memory (RAM) is not cleared upon initialization.* The first instuction at address 0 must be a CLRA.

OSCILLATOR

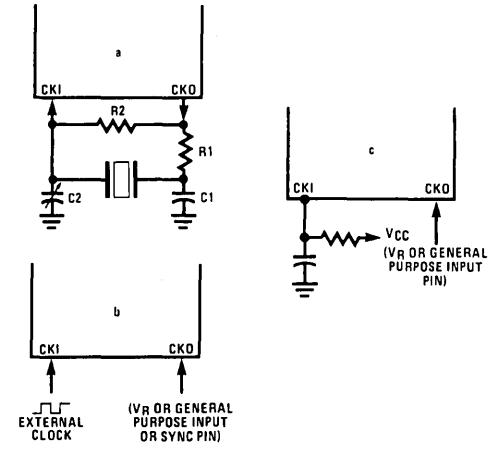
There are four basic clock oscillator configurations available as shown by *Figure 4.*

a. **Crystal Controlled Oscillator.** CKI and CKO are connected to an external crystal. The instruction cycle time equals the crystal frequency divided by 32 (optional by 16 or 8).

b. **External Oscillator.** CKI is an external clock input signal. The external frequency is divided by 32 (optional by 16 or 8) to give the instruction cycle time. CKO is now available to be used as the RAM power supply (V_R), as a general purpose input.

c. **RC Controlled Oscillator.** CKI is configured as a single pin RC controlled Schmitt trigger oscillator. The instruction cycle equals the oscillation frequency divided by 4. CKO is available as the RAM power supply (V_R) or as a general purpose input.

Functional Description (Continued)



TL/DD/6928–7

Crystal Oscillator

Crystal Value	Component Values			
	R1 (Ω)	R2 (Ω)	C1 (pF)	C2 (pF)
455 kHz	4.7k	1M	220	220
2.097 MHz	1k	1M	30	6–36

RC Controlled Oscillator

R (kΩ)	C (pF)	Instruction Cycle Time (μs)
51	100	19 ± 15%
82	56	19 ± 13%

NOTE: 200 kΩ ≥ R ≥ 25 kΩ
360 pF ≥ C ≥ 50 pF

FIGURE 4. COP444L/445L Oscillator

CKO PIN OPTIONS

In a crystal controlled oscillator system, CKO is used as an output to the crystal network. As an option CKO can be a general purpose input, read into bit 2 of A (accumulator) upon execution of an INIL instruction. As another option, CKO can be a RAM power supply pin (V_R), allowing its connection to a standby/backup power supply to maintain the integrity of RAM data with minimum power drain when the main supply is inoperative or shut down to conserve power. Using either option is appropriate in applications where the COP444L/445L system timing configuration does not require use of the CKO pin.

I/O OPTIONS

COP444L/445L outputs have the following optional configurations, illustrated in *Figure 5*.

a. Standard—an enhancement mode device to ground in conjunction with a depletion-mode device to V_{CC}, compatible with LSTTL and CMOS input requirements. Available on SO, SK, and all D and G outputs.

b. Open-Drain—an enhancement-mode device to ground only, allowing external pull-up as required by the user's application. Available on SO, SK, and all D and G outputs.

c. Push-Pull—An enhancement-mode device to ground in conjunction with a depletion-mode device paralleled by an enhancement-mode device to V_{CC}. This configuration has been provided to allow for fast rise and fall times when driving capacitive loads. Available on SO and SK outputs only.

d. Standard L—same as **a.**, but may be disabled. Available on L outputs only.

e. Open Drain L—same as **b.**, but may be disabled. Available on L outputs only.

f. LED Direct Drive—an enhancement-mode device to ground and to V_{CC}, meeting the typical current sourcing requirements of the segments of an LED display. The sourcing device is clamped to limit current flow. These devices may be turned off under program control (See Functional Description, EN Register), placing the outputs in a high impedance state to provide required LED segment blanking for a multiplexed display. Available on L outputs only.

g. TRI-STATE Push-Pull—an enhancement-mode device to ground and V_{CC}. These outputs are TRI-STATE outputs, allowing for connection of these outputs to a data bus shared by other bus drivers. Available on L outputs only.

COP444L/COP445L inputs have the following optional configurations:

h. An on-chip depletion load device to V_{CC}.

i. A Hi-Z input which must be driven to a "1" or "0" by external components.

The above input and output configurations share common enhancement-mode and depletion-mode devices. Specifically, all configurations use one or more of six devices (numbered 1–6, respectively). Minimum and maximum current (I_{OUT} and V_{OUT} curves are given in *Figure 6* for each of these devices to allow the designer to effectively use these I/O configurations in designing a system.

The SO, SK outputs can be configured as shown in **a.**, **b.**, or **c.** The D and G outputs can be configured as shown in **a.** or **b.** Note that when inputting data to the G ports, the G outputs should be set to "1". The L outputs can be configured in **d.**, **e.**, **f.** or **g.**

An important point to remember if using configuration **d.** or **f.** with the L drivers is that even when the L drivers are disabled, the depletion load device will source a small amount of current (see *Figure 6*, device 2); however, when the L-lines are used as inputs, the disabled depletion device can *not* be relied on to source sufficient current to pull an input to logic "1".

RAM KEEP-ALIVE OPTION

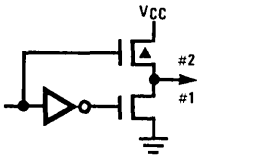
Selecting CKO as the RAM power supply (V_R) allows the user to shut off the chip power supply (V_{CC}) and maintain data in the lower four (Br = 0, 1, 2, 3) registers of RAM. To insure that RAM data integrity is maintained, the following conditions *must* be met:

1. $\overline{\text{RESET}}$ must go low before V_{CC} goes low during power off; V_{CC} must go high before $\overline{\text{RESET}}$ goes high on power-up.
2. V_R must be within the operating range of the chip, and equal to $V_{CC} \pm 1V$ during normal operation.
3. V_R must be ≥ 3.3V with V_{CC} off.

Functional Description (Continued)

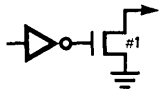
COP445L

If the COP444L is bonded as a 24-pin device, it becomes the COP455L, illustrated in *Figure 2*, COP444L/445L Connection Diagrams. Note that the COP445L does not contain the four general purpose IN inputs (IN_3–IN_0). Use of this option precludes, of course, use of the IN options and the interrupt feature, which uses IN_1. All other options are available for the COP445L.



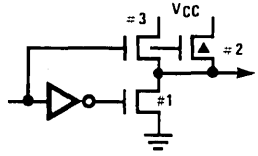
TL/DD/6928-9

a. Standard Output



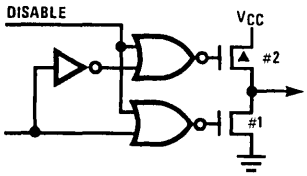
TL/DD/6928-10

b. Open-Drain Output



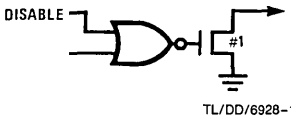
TL/DD/6928-11

c. Push-Pull Output



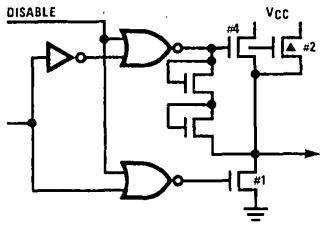
TL/DD/6928-12

d. Standard L Output



TL/DD/6928-13

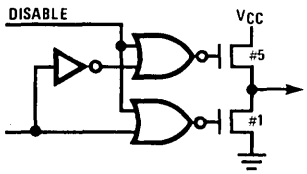
e. Open-Drain L Output



TL/DD/6928-14

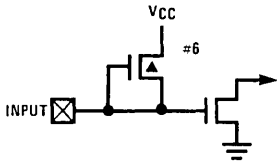
(▲ is Depletion Device)

f. LED (L Output)



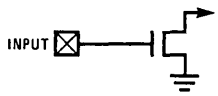
TL/DD/6928-15

g. TRI-STATE Push-Pull (L Output)



TL/DD/6928-16

h. Input with Load



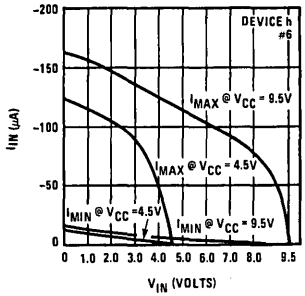
TL/DD/6928-17

i. Hi-Z Input

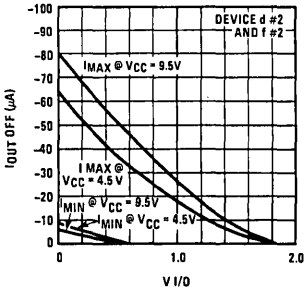
FIGURE 5. Output Configuration

Typical Performance Characteristics

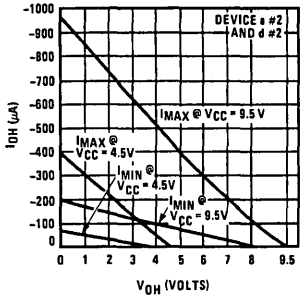
Current for Inputs with Load Device



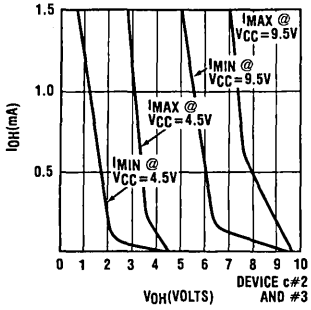
Input Current for L_0 through L_7 when Output Programmed Off by Software



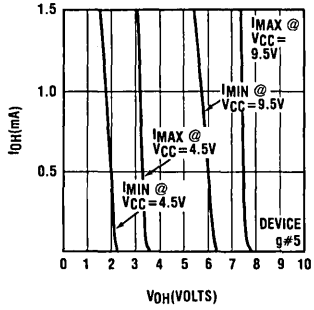
Source Current for Standard Output Configuration



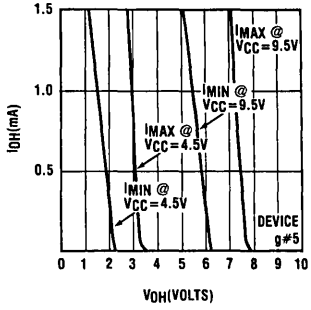
Source Current for SO and SK in Push-Pull Configuration



Source Current for L_0 through L_7 in TRI-STATE Configuration (High Current Option)



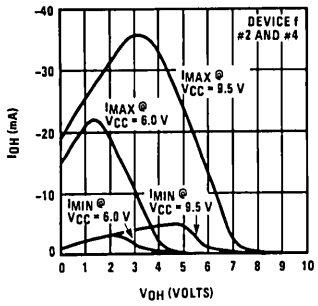
Source Current for L_0 through L_7 in TRI-STATE configuration (Low Current Option)



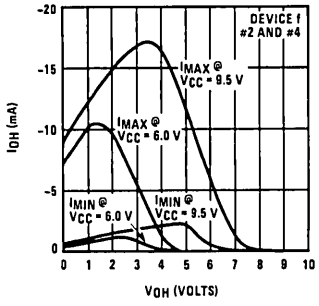
TL/DD/6928–18

Typical Performance Characteristics (Continued)

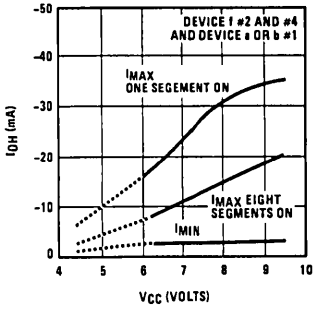
LED Output Source Current (for High Current LED Option)



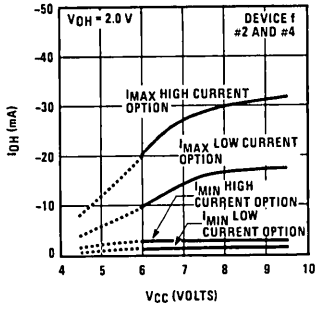
LED Output Source Current (for Low Current LED Option)



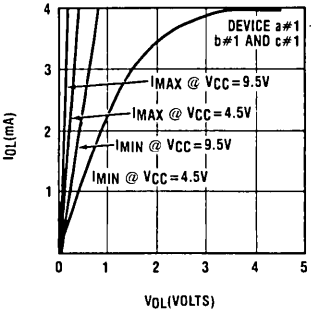
LED Output Direct Segment Drive High Current Options on L_0–L_7 Very High Current Options on D_0–D_3 or G_0–G_3



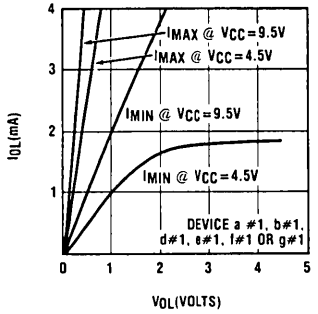
LED Output Direct Segment Drive



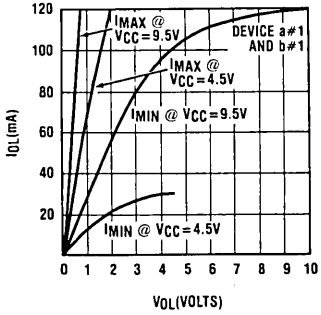
Output Sink Current for SO and SK



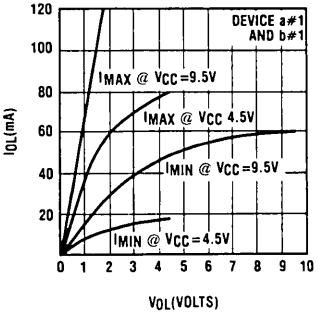
Output Sink Current for L_0–L_7 and Standard Drive Option for D_0–D_3 and G_0–G_3



Output Sink Current G_0–G_3 and D_0–D_3 with Very High Current Option



Output Sink Current for G_0–G_3 and D_0–D_3 (for High Current Option

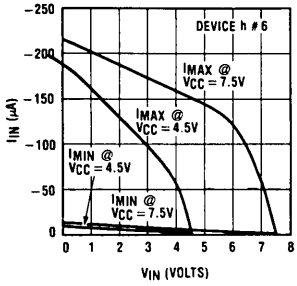


TL/DD/6928–19

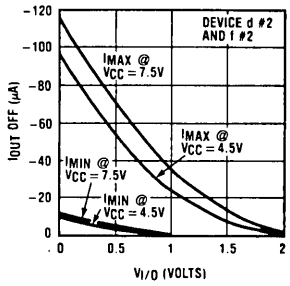
FIGURE 6a. COP444L/COP445L Input/Output Characteristics

Typical Performance Characteristics (Continued)

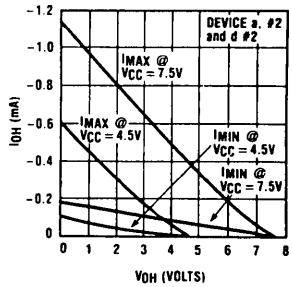
Input Current IN_0–IN_3



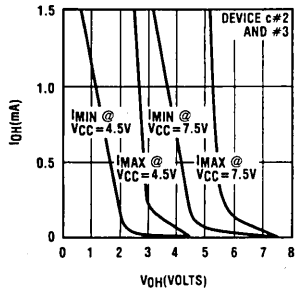
Input Current for L0–L7 when Output Programmed Off by Software



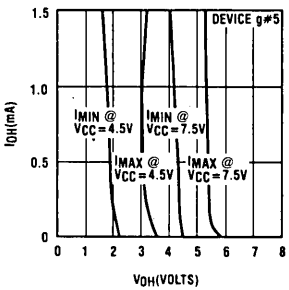
Source Current for Standard Output Configuration



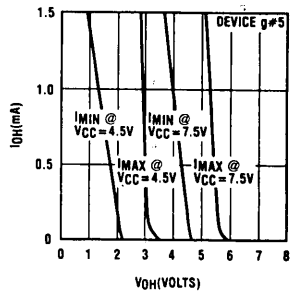
Source Current for SO and SK in Push-Pull Configuration



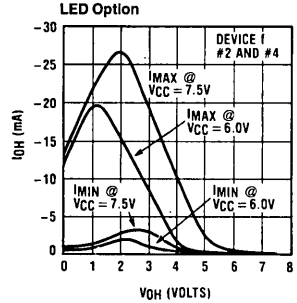
Source Current for L0–L7 in TRI-STATE Configuration (High Current Option)



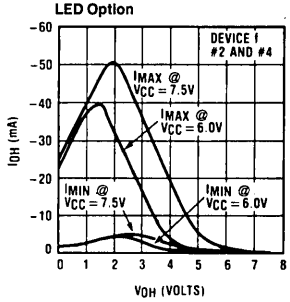
Source Current for L0–L7 in TRI-STATE Configuration (Low Current Option)



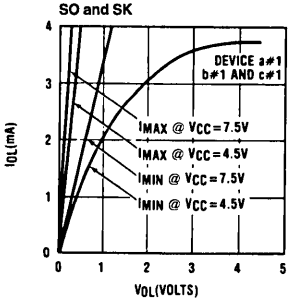
LED Output Source Current (for Low Current LED Option



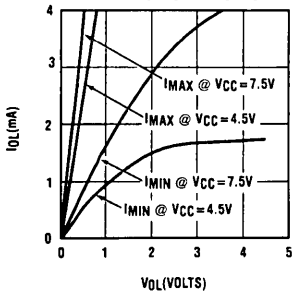
LED Output Source Current (for High Current LED Option



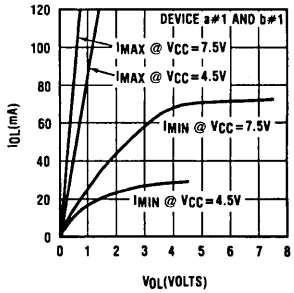
Output Sink Current for SO and SK



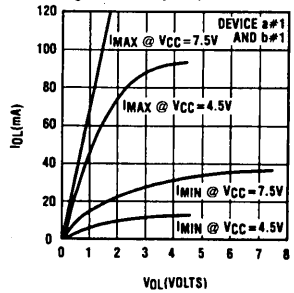
Output Sink Current for L_0–L_7 and Standard Drive Option for D_0–D_3 and G_0–G_3



Output Sink Current G_0–G_3 and D_0–D_3 with Very High Current Option



Output Sink Current for G_0–G_3 and D_0–D_3 (for High Current Option)



TL/DD/6928–20

Fiare 6b. COP344L/COP345L Input/Output Characteristics

COP444L/COP445L/COP344L/COP345L Instruction Set

Table I is a symbol table providing internal architecture, instruction operand and operational symbols used in the instruction set table.

Table II provides the mnemonic, operand, machine code, data flow, skip conditions and description associated with each instruction in the COP444L/445L instruction set.

TABLE I. COP444L/445L/344L/345L Instruction Table Symbols

Symbol	Definition
INTERNAL ARCHITECTURE SYMBOLS	
A	4-bit Accumulator
B	6-bit RAM Address Register
Br	Upper 3 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry Register
D	4-bit Data Output Port
EN	4-bit Enable Register
G	4-bit Register to latch data for G I/O Port
IL	Two 1-bit latches associated with the IN_3 or IN_0 inputs
IN	4-bit Input Port
L	8-bit TRI-STATE I/O Port
M	4-bit contents of RAM Memory pointed to by B Register
PC	11-bit ROM Address Register (program counter)
Q	8-bit Register to latch data for L I/O Port
SA	11-bit Subroutine Save Register A
SB	11-bit Subroutine Save Register B
SC	11-bit Subroutine Save Register C
SIO	4-bit Shift Register and Counter
SK	Logic-Controlled Clock Output

Symbol	Definition
INSTRUCTION OPERAND SYMBOLS	
d	4-bit Operand Field, 0-15 binary (RAM Digit Select)
r	3-bit Operand Field, 0-7 binary (RAM Register Select)
a	11-bit Operand Field, 0-2047 binary (ROM Address)
y	4-bit Operand Field, 0-15 binary (Immediate Data)
RAM(s)	Contents of RAM location addressed by s
ROM(t)	Contents of ROM location addressed by t
OPERATIONAL SYMBOLS	
+	Plus
−	Minus
→	Replaces
↔	Is exchanged with
=	Is equal to
$\overline{A}$	The one's complement of A
⊕	Exclusive-OR
:	Range of values

TABLE II. COP444L/445L Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	|0011|0000|	A + C + RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	|0011|0001|	A + RAM(B) → A	None	Add RAM to A
ADT		4A	|0100|1010|	A + 10_{10} → A	None	Add Ten to A
AISC	y	5−	|0101| y |	A + y → A	Carry	Add Immediate, Skip on Carry (y ≠ 0)
CASC		10	|0001|0000|	$\overline{A}$ + RAM(B) + C → A Carry → C	Carry	Complement and Add with Carry, Skip on Carry
CLRA		00	|0000|0000|	0 → A	None	Clear A
COMP		40	|0100|0000|	$\overline{A}$ → A	None	Ones complement of A to A
NOP		44	|0100|0100|	None	None	No Operation
RC		32	|0011|0010|	"0" → C	None	Reset C
SC		22	|0010|0010|	"1" → C	None	Set C

Instruction Set (Continued)

TABLE II. COP444L/445L Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
TRANSFER OF CONTROL INSTRUCTIONS						
XOR		02	0000 | 0010	A ⊕ RAM(B) → A	None	Exclusive-OR RAM with A
JID		FF	1111 | 1111	ROM ($PC_{10:8}$, A,M) → $PC_{7:0}$	None	Jump Indirect (Note 3)
JMP	a	6–	0110 | 0 | $a_{10:8}$	a → PC	None	Jump
		– –	$a_{7:0}$			
JP	a	– –	1 | $a_{6:0}$ (pages 2,3 only)	a → $PC_{6:0}$	None	Jump within Page (Note 4)
			or			
		– –	11 | $a_{5:0}$ (all other pages)	a → $PC_{5:0}$		
JSRP	a	– –	10 | $a_{5:0}$	PC + 1 → SA → SB → SC 00010 → $PC_{10:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 5)
JSR	a	6–	0110 | 1 | $a_{10:8}$	PC + 1 → SA → SB → SC a → PC	None	Jump to Subroutine
		– –	$a_{7:0}$			
RET		48	0100 | 1000	SC → SB → SA → PC	None	Return from Subroutine
RETSK		49	0100 | 1001	SC → SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
MEMORY REFERENCE INSTRUCTIONS						
CAMQ		33	0011 | 0011	A → $Q_{7:4}$ RAM(B) → $Q_{3:0}$	None	Copy A, RAM to Q
		3C	0011 | 1100			
CQMA		33	0011 | 0011	$Q_{7:4}$ → RAM(B) $Q_{3:0}$ → A	None	Copy Q to RAM, A
		2C	0010 | 1100			
LD	r	–5	00 | r | 0101 (r = 0:3)	RAM(B) → A Br ⊕ r → Br	None	Load RAM into A Exclusive-OR Br with r
LDD	r,d	23	0010 | 0011	RAM(r,d) → A	None	Load A with RAM pointed to directly by r,d
		– –	0 | r | d			
LQID		BF	1011 | 1111	ROM($PC_{10:8}$,A,M) → Q SB → SC	None	Load Q Indirect (Note 3)
RMB	0	4C	0100 | 1100	0 → $RAM(B)_0$	None	Reset RAM Bit
	1	45	0100 | 0101	0 → $RAM(B)_1$		
	2	42	0100 | 0010	0 → $RAM(B)_2$		
	3	43	0100 | 0011	0 → $RAM(B)_3$		
SMB	0	4D	0100 | 1101	1 → $RAM(B)_0$	None	Set RAM Bit
	1	47	0100 | 1101	1 → $RAM(B)_1$		
	2	46	0100 | 0110	1 → $RAM(B)_2$		
	3	4B	0100 | 1011	1 → $RAM(B)_3$		
STII	y	7–	0111 | y	y → RAM(B) Bd + 1 → Bd	None	Store Memory Immediate and Increment Bd
X	r	–6	00 | r | 0110 (r = 0:3)	RAM(B) ⟷ A Br ⊕ r → Br	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	r,d	23	0010 | 0011	RAM(r,d) ⟷ A	None	Exchange A with RAM pointed to directly by r,d
		– –	1 | r | d			

Instruction Set (Continued)

TABLE II. COP444L/445L Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
MEMORY REFERENCE INSTRUCTIONS (Continued)						
XDS	r	–7	|00|r|0111| (r = 0:3)	RAM(B) ⟷ A Bd – 1 → Bd Br ⊕ r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd, Exclusive-OR Br with r
XIS	r	–4	|00|r|0100| (r = 0:3)	RAM(B) ⟷ A Bd + 1 → Bd Br ⊕ r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd, Exclusive-OR Br with r
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	|0101|0000|	A → Bd	None	Copy A to Bd
CBA		4E	|0100|1110|	Bd → A	None	Copy Bd to A
LBI	r,d	– –	|00|r|(d - 1)| (r = 0:3; d = 0, 9:15)	r,d → B	Skip until not a LBI	Load B Immediate with r,d (Note 6)
			or			
		33	|0011|0011|			
		– –	|1|r|d| any r, any d)			
LEI	y	33	|0001|0011|	y → EN	None	Load EN Immediate (Note 7)
		6–	|0110|y|			
XABR		12	|0001|0010|	A ⟷ Br (0 → A_3)	None	Exchange A with Br
TEST INSTRUCTIONS						
SKC		20	|0010|0000|		C = "1"	Skip if C is True
SKE		21	|0010|0001|		A = RAM(B)	Skip if A Equals RAM
SKGZ		33	|0011|0011|		$G_{3:0}$ = 0	Skip if G is Zero (all 4 bits)
		21	|0010|0001|			
SKGBZ		33	|0011|0011|	1st byte		Skip if G Bit is Zero
	0	01	|0000|0001|	2nd byte	G_0 = 0	
	1	11	|0001|0001|		G_1 = 0	
	2	03	|0000|0011|		G_2 = 0	
	3	13	|0001|0011|		G_3 = 0	
SKMBZ	0	01	|0000|0001|		$RAM(B)_0$ = 0	Skip if RAM Bit is Zero
	1	11	|0001|0001|		$RAM(B)_1$ = 0	
	2	03	|0000|0011|		$RAM(B)_2$ = 0	
	3	13	|0001|0011|		$RAM(B)_3$ = 0	
SKT		41	|0100|0001|		A time-base counter carry has occurred since last test	Skip on Timer (Note 3)
INPUT/OUTPUT INSTRUCTIONS						
ING		33	|0011|0011|	G → A	None	Input G Ports to A
		2A	|0010|1010|			
ININ		33	|0011|0011|	IN → A	None	Input IN Inputs to A (Note 2)
		28	|0010|1000|			
INIL		33	|0011|0011|	IL_3, CKO, "0", IL_0 → A	None	Input IL Latches to A (Note 3)
		29A	|0010|1001|			

Instruction Set (Continued)

TABLE II. COP444L/445L Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
INPUT/OUTPUT INSTRUCTIONS (Continued)						
INL		33	0011 | 0011	$L_{7:4} \rightarrow$ RAM(B)	None	Input L Ports to RAM, A
		2E	0010 | 1110	$L_{3:0} \rightarrow$ A		
OBD		33	0011 | 0011	Bd $\rightarrow$ D	None	Output Bd to D Outputs
		3E	0011 | 1110			
OGI	y	33	0011 | 0011	y $\rightarrow$ G	None	Output to G Ports Immediate
		5–	0101 | t			
OMG		33	0011 | 0011	RAM(B) $\rightarrow$ G	None	Output RAM to G Ports
		3A	0011 | 1010			
XAS		4F	0100 | 1111	A $\longleftrightarrow$ SIO, C $\rightarrow$ SKL	None	Exchange A with SIO (Note 3)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicitly defined (e.g., Br and Bd are explicitly defined). Bits are numbered 0 to N where 0 signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit A register.

Note 2: The ININ instruction is not available on the 24-pin COP445L or COP345L since these devices do not contain the IN inputs.

Note 3: For additional information on the operation of the XAS, JID, LQID, INIL, and SKT instructions, see below.

Note 4: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 5: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 6: LBI is a single-byte instruction if d = 0, 9, 10, 11, 12, 13, 14 or 15. The machine code for the lower 4 bits equals the binary value of the "d" data *minus 1*, e.g., to load the lower four bits of B (Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Note 7: Machine code for operand field y for LEI instruction should equal the binary value to be latched into EN, where a "1" or "0" in each bit of EN corresponds with the selection or deselection of a particular function associated with each bit. (See Functional Description, EN Register.)

Description of Selected Instructions

The following information is provided to assist the user in understanding the operation of several unique instructions and to provide notes useful to programmers in writing COP444L/445L programs.

XAS INSTRUCTION

XAS (Exchange A with SIO) exchanges the 4-bit contents of the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. An XAS instruction will also affect the SK output. (See Functional Description, EN Register, above.) If SIO is selected as a shift register, an XAS instruction must be performed once every 4 instruction cycles to effect a continuous data stream.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower 8 bits of the ROM address register PC with the *contents* of ROM addressed by the 11-bit word, $PC_{10:8}$, A, M. PC_{10}, PC_9 and PC_8 are not affected by this instruction.

Note that JID requires 2 instruction cycles to execute.

INIL INSTRUCTION

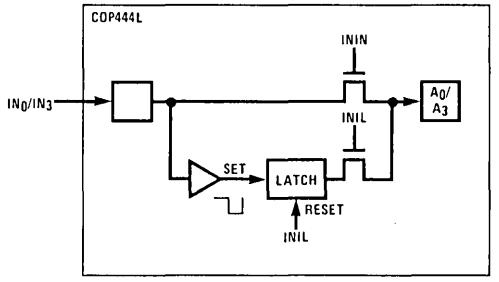
INIL (Input IL Latches to A) inputs 2 latches, IL_3 and IL_0 (see *Figure 7*) and CKO into A. The IL_3 and IL_0 latches are set if a low-going pulse ("1" to "0") has occurred or the IN_3 and IN_0 inputs since the last INIL instruction, provided the input pulse stays low for at least two instruction times. Execution of an INIL inputs IL_3 and IL_0 into A3 and A0 respectively, and resets these latches to allow them to respond to subsequent low-going pulses on the IN_3 and IN_0 lines. If CKO is mask programmed as a general purpose input, an INIL will input the state of CKO into A2. If CKO has not been so programmed, a "1" will be placed in A2. A "0" is always placed in A1 upon the execution of an INIL. The general purpose inputs IN_3–IN_0 are input to A upon execution of an ININ instruction. (See Table II, ININ instruction.) INIL is useful in recognizing pulses of short duration or pulses which occur too often to be read conveniently by an ININ instruction.

Note: IL latches are not cleared on reset; IL_3–IL_0 *not* input on 445L

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 11-bit word PC_{10}, PC_9, PC_8 A, M. LQID can be used for table lookup or code conversion such as BCD to seven-segment. The LQID instruction "pushes" the stack (PC + 1 $\rightarrow$ SA $\rightarrow$ SB $\rightarrow$ SC) and replaces the least significant 8 bits of PC as follows: A $\rightarrow$ $PC_{7:4}$, RAM(B) $\rightarrow$ $PC_{3:0}$, leaving PC_{10}, PC_9 and PC_8 unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SC $\rightarrow$ SB $\rightarrow$ SA $\rightarrow$ PC), restoring the saved value of PC to continue sequential program execution. Since LQID pushes SB $\rightarrow$ SC, the previous contents

Description of Selected Instructions (Continued)



TL/DD/6928–21

FIGURE 7. INIL Hardware Implementation

of SC are lost. Also, when LQID pops the stack, the previously pushed contents of SB are left in SC. The net result is that the contents of SB are placed in SC (SB → SC). Note that LQID takes two instruction cycle times to execute.

SKT INSTRUCTION

The SKT (Skip On Timer) instruction tests the state of an internal 10-bit time-base counter. This counter divides the instruction cycle clock frequency by 1024 and provides a latched indication of counter overflow. The SKT instruction tests this latch, executing the next program instruction if the latch is not set. If the latch has been set since the previous test, the next program instruction is skipped and the latch is reset. The features associated with this instruction, therefore, allow the COP444L/445L to generate its own time-base for real-time processing rather than relying on an external input signal.

For example, using a 2.097 MHz crystal as the time-base to the clock generator, the instruction cycle clock frequency will be 65 kHz (crystal frequency ÷32) and the binary counter output pulse frequency will be 64 Hz. For time-of-day or similar real-time processing, the SKT instruction can call a routine which increments a "seconds" counter every 64 ticks.

INSTRUCTION SET NOTES

a. The first word of a COP444L/445L program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, one instruction cycle time is devoted to skipping each byte of the skipped instruction. Thus all program paths except JID and LQID take the same number of cycle times whether instructions are skipped or executed. JID and LQID instructions take 2 cycles if executed and 1 cycle if skipped.

c. The ROM is organized into 32 pages of 64 words each. The Program Counter is an 11-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID or LQID instruction is located in the last word of a page, the instruction operates as if it were in the next page. For example: a JP located in the last work of a page will jump to a location in the next page. Also, a LQID or JID located in the last word of page 3, 7, 11, 15, 19, 23 or 27 will access data in the next group of four pages.

Option List

The COP444L/445L mask-programmable options are assigned numbers which correspond with the COP*444L* pins.

The following is a list of COP444L options. When specifying a COP445L chip, Options 9, 10, 19, and 20 must all be set to zero. The options are programmed at the same time as the ROM pattern to provide the user with the hardware flexibility to interface to various I/O components using little or no external circuitry.

Option 1 = 0: Ground Pin—no options available

Option 2: CKO Output
- = 0: clock generator ouput to crystal/resonator (0 not allowable value if option 3 = 3)
- = 1: pin is RAM power supply (V_R) input
- = 2: general purpose input, load device to V_{CC}
- = 3: general purpose input, Hi-Z

Option 3: CKI Input
- = 0: oscillator input divided by 32 (2 MHz max.)
- = 1: oscillator input divided by 16 (1 MHz max.)
- = 2: oscillator input divided by 8 (500 kHz max.)
- = 3: single-pin RC controlled oscillator divided by 4
- = 4: oscillator input divided by 4 (Schmitt)

Option 4: $\overline{\text{RESET}}$ Input
- = 0: load device to V_{CC}
- = 1: Hi-Z input

Option 5: L_7 Driver
- = 0: Standard output
- = 1: Open-drain output
- = 2: High current LED direct segment drive output
- = 3: High current TRI-STATE push-pull output
- = 4: Low-current LED direct segment drive output
- = 5: Low-current TRI-STATE push-pull output

Option 6: L_6 Driver
same as Option 5

Option 7: L_5 Driver
same as Option 5

Option 8: L_4 Driver
same as Option 5

Option 9: IN_1 Input
- = 0: load device to V_{CC}
- = 1: Hi-Z input

Option 10: IN_2 Input
same as Option 9

Option 11: V_{CC} pin Operating Voltage

	COP44XL	COP34XL
= 0:	+4.5V to +6.3V	+4.5V to +5.5V
= 1:	+4.5V to +9.5V	+4.5V to +7.5V

Option 12: L_3 Driver
same as Option 5

Option 13: L_2 Driver
same as Option 5

Option 14: L_1 Driver
same as Option 5

Option 15: L_0 Driver
same as Option 5

Option 16: SI Input
same as Option 9

Option List (Continued)

Option 17: SO Driver
- = 0: standard output
- = 1: open-drain output
- = 2: push-pull output

Option 18: SK Driver
same as Option 17

Option 19: IN_0 Input
same as Option 9

Option 20: IN_3 Input
same as Option 9

Option 21: G_0 I/O Port
- = 0: very-high current standard output
- = 1: very-high current open-drain output
- = 2: high current standard output
- = 3: high current open-drain output
- = 4: standard LSTTL output (fanout = 1)
- = 5: open-drain LSTTL output (fanout = 1)

Option 22: G_1 I/O Port
same as Option 21

Option 23: G_2 I/O Port
same as Option 21

Option 24: G_3 I/O Port
same as Option 21

Option 25: D_3 Output
same as Option 21

Option 26: D_2 Output
same as Option 21

Option 27: D_1 Output
same as Option 21

Option 28: D_0 Output
same as Option 21

Option 29: L Input Levels
- = 0: standard TTL input levels ("0" = 0.8V, "1" = 2.0V)
- = 1: higher voltage input levels ("0" = 1.2V, "1" = 3.6V)

Option 30: IN Input Levels
same as Option 29

Option 31: G Input Levels
same as Option 29

Option 32: SI Input Levels
same as Option 29

Option 33: RESET Input
- = 0: Schmitt trigger input levels
- = 1: standard TTL input levels
- = 2: higher voltage input levels

Option 34: CKO Input Levels (CKO = input; Option 2 = 2, 3)
same as Option 29

Option 35: COP Bonding
- = 0: COP444L (28-pin device)
- = 1: COP445L (24-pin device)
- = 2: both 28- and 24-pin versions

Option 36: Internal Initialization Logic
- = 0: normal operation
- = 1: no internal initialization logic

COP444L Option Table

The following option information is to be seNt to National along with the EPROM.

	OPTION DATA	
OPTION 1	VALUE= _____	IS: GROUND PIN
OPTION 2	VALUE= _____	IS: CKO PIN
OPTION 3	VALUE= _____	IS: CKI PIN
OPTION 4	VALUE= _____	IS: RESET INPUT
OPTION 5	VALUE= _____	IS: L(7) DRIVER
OPTION 6	VALUE= _____	IS: L(6) DRIVER
OPTION 7	VALUE= _____	IS: L(5) DRIVER
OPTION 8	VALUE= _____	IS: L(4) DRIVER
OPTION 9	VALUE= _____	IS: IN1 INPUT
OPTION 10	VALUE= _____	IS: IN2 INPUT
OPTION 11	VALUE= _____	IS: VCC PIN
OPTION 12	VALUE= _____	IS: L(3) DRIVER
OPTION 13	VALUE= _____	IS: L(2) DRIVER
OPTION 14	VALUE= _____	IS: L(1) DRIVER
OPTION 15	VALUE= _____	IS: L(0) DRIVER
OPTION 16	VALUE= _____	IS: SI INPUT
OPTION 17	VALUE= _____	IS: SO DRIVER
OPTION 18	VALUE= _____	IS: SK DRIVER
OPTION 19	VALUE= _____	IS: IN0 INPUT
OPTION 20	VALUE= _____	IS: IN3 INPUT

	OPTION DATA	
OPTION 21	VALUE= _____	IS: G0 I/O PORT
OPTION 22	VALUE= _____	IS: G1 I/O PORT
OPTION 23	VALUE= _____	IS: G2 I/O PORT
OPTION 24	VALUE= _____	IS: G3 I/O PORT
OPTION 25	VALUE= _____	IS: D3 OUTPUT
OPTION 26	VALUE= _____	IS: D2 OUTPUT
OPTION 27	VALUE= _____	IS: D1 OUTPUT
OPTION 28	VALUE= _____	IS: D0 OUTPUT
OPTION 29	VALUE= _____	IS: L INPUT LEVELS
OPTION 30	VALUE= _____	IS: IN INPUT LEVELS
OPTION 31	VALUE= _____	IS: G INPUT LEVELS
OPTION 32	VALUE= _____	IS: SI INPUT LEVELS
OPTION 33	VALUE= _____	IS: RESET INPUT
OPTION 34	VALUE= _____	IS: CKO INPUT LEVELS
OPTION 35	VALUE= _____	IS: COP BONDING
OPTION 36	VALUE= _____	IS: INTERNAL INITIALIZATION LOGIC

Typical Applications

TEST MODE (NON-STANDARD OPERATION)

The SO output has been configured to provide for standard test procedures for the custom-programmed COP444L. With SO forced to logic "1", two test modes are provided, depending upon the value of SI:

a. RAM and Internal Logic Test Mode (SI = 1)

b. ROM Test Mode (SI = 0)

These special test modes should not be employed by the user; they are intended for manufacturing test only.

APPLICATION # 1: COP444L GENERAL CONTROLLER

Figure 8 shows an interconnect diagram for a COP444L used as a general controller. Operation of the system is as follows:

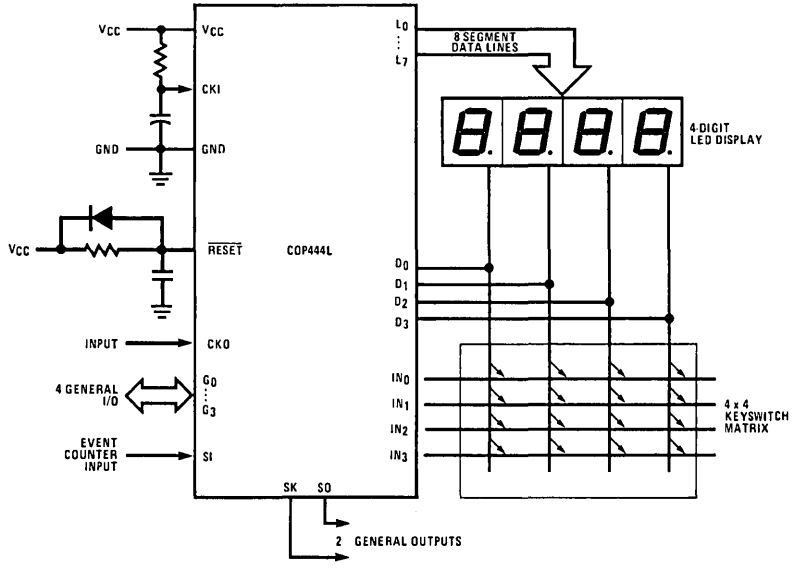
1. The L_7–L_0 outputs are configured as LED Direct Drive outputs, allowing direct connection to the segments of the display
2. The D_3–D_0 outputs drive the digits of the multiplexed display directly and scan the columns of the 4 x 4 keyboard matrix.
3. The IN_3–IN_0 inputs are used to input the 4 rows of the keyboard matrix. Reading the IN lines in conjunction with the current value of the D outputs allows detection, debouncing, and decoding of any one of the 16 keyswitches.
4. CKI is configured as a single-pin oscillator input allowing system timing to be controlled by a single-pin RC network. CKO is therefore available for use as a general-purpose input.
5. SI is selected as the input to a binary counter input. With SIO used as a binary counter, SO and SK can be used as general purpose outputs.
6. The 4 bidirectional G I/O ports (G_3–G_0) are available for use as required by the user's application.
7. Normal reset operation is selected.

COP444L EVALUATION (See COP Note 4)

The 444L-EVAL is a pre-programmed COP444L, containing several routines which facilitate user familiarization and evaluation of the COP444L operating characteristics. It may be used as an up/down counter or timer, interfacing to any combination of (1) an LED digit or lamps, (2) 4-digit LED Display Controller, (3) a 4-digit VF Display Controller, and/or (4) a 4-digit LCD Display Controller. Alternatively, it may be used as a simple music synthesizer.

SAMPLE CIRCUITS

1. By making only the oscillator, power supply and "L7" connections, *(Figure 9)* an approximate 1 Hz square wave will be produced at output "D1." This output may be observed with an oscilloscope, or connected to additional TTL or CMOS circuitry.
2. By making the indicated connections to a small LED digit (NSA1541A, NSA1166, or equiv.—larger digits will be proportionately dimmer), the counter actions may be observed. Place the "up/down" switch in the "up" (open) position and apply a TTL-compatible signal at the "counter-input." Placing the "up/down" switch in the "down" (closed) position causes the count to decrement on each high-to-low input transition.
3. All 4 digits of the counter may be displayed by connecting a standard display controller (COP470 for VF, COP472 for LCD, MM5450 for LED) as shown in *Figure 9*.



TL/DD/6928-22

FIGURE 8. COP444L Keyboard/Display Interface

Typical Applications (Continued)

Any combination of the single LED digit and display controllers may be used simultaneously, and will display the same data.

4. The simple counter described above becomes a timer when the 1 Hz output is connected to the "counter input." Up or down counting may be used with input frequencies up to 1 kHz. Improved timing accuracies may be obtained by subsituting the 2.097 MHz crystal oscillator circuit of *Figure 4a* for the RC network shown in *Figure 9*, or by connecting a more stable external frequency to the "counter input" in place of the 1 Hz signal.
5. An "entertaining" use of the 444L-EVAL is as a simple music synthesizer (or electronic organ). By attaching a simple switch matrix (or keyboard), a speaker or piezo-ceramic transducer, and grounding "L7", the user can play "music" *(Figure 10)*. Three modes of operation are available: Play a note, play one of four stored tunes, or record a tune for subsequent replay.

   a. *Play A Note*

   Twelve keys, representing the 12 notes in one octave, are labeled "C" through "B"; depressing a key causes a square wave of the corresponding frequency to be outputted to the speaker. Depressing "LShift" or "UShift" causes the next note to be shifted to the next lower octave (one-half frequency) or the next upper octave (double frequency), respectively.

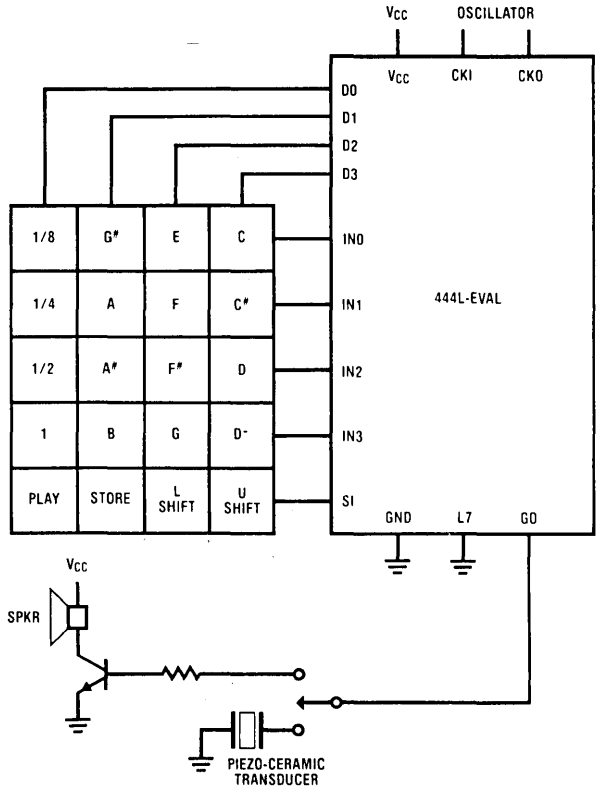
   b. *Play Stored Tune*

   Depressing "Play" followed by "1/8", "1/4", "1/2", or "1" will cause one of 4 stored tunes to be played.

   c. *Record Tune*

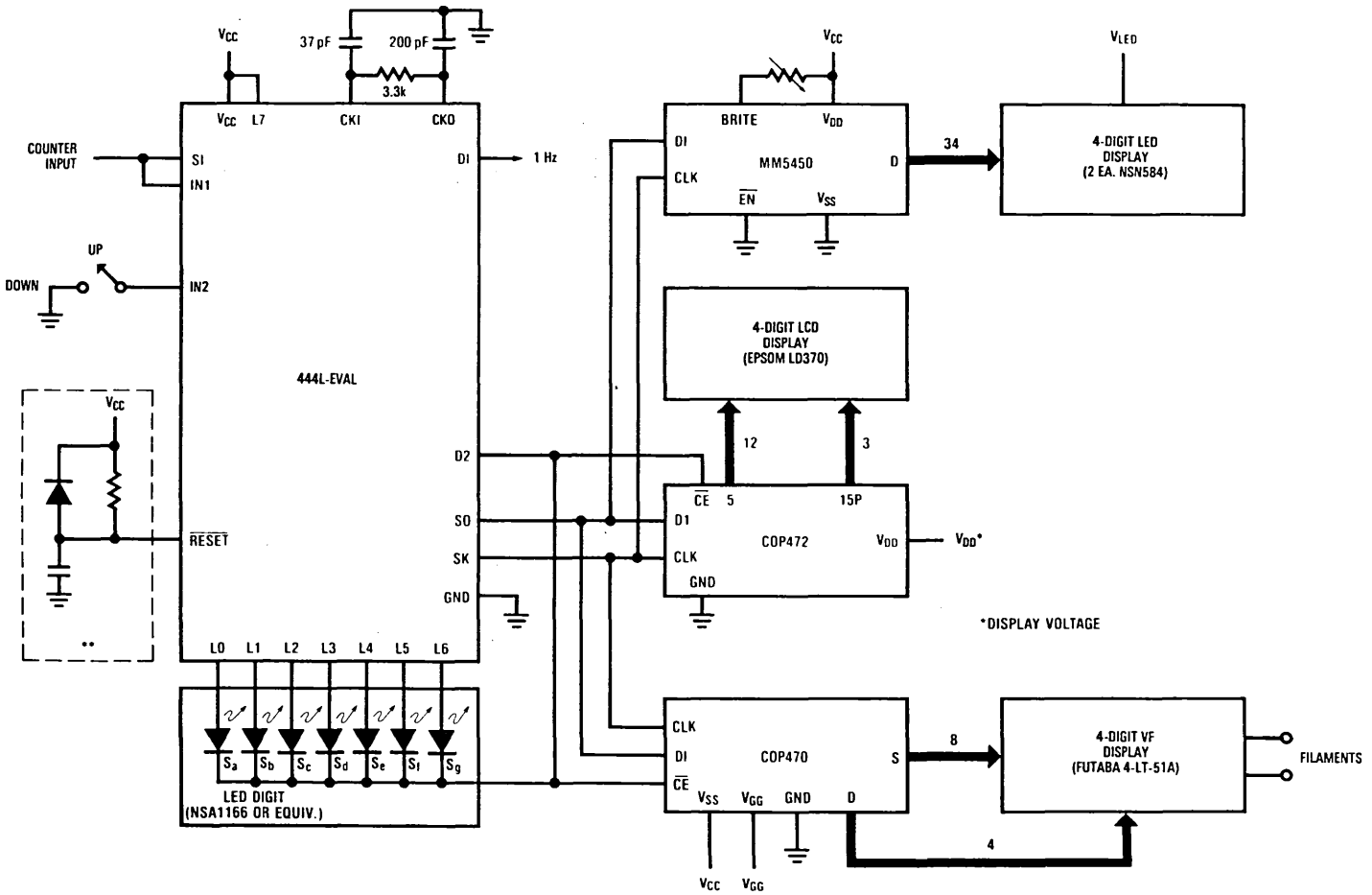
   Any combination of notes and rests up to a total of 48 may be stored in RAM for later replay. To store a note, press the appropriate note key, followed by the duration of the note (1/8-note, 1/4-note, 1/2-note, whole (1)-note, followed by "Store"; a rest is stored by selecting the duration and pressing "Store." When the tune is complete, press "Play" followed by "Store"; the tune will be played for immediate audition. Subsequent depression of "Play" and "Store" will replay the last stored tune.

Note: The accuracy of the tones produced is a function of the oscillator accuracy and stability; the crystal oscillator is recommended.



TL/DD/6928-23

FIGURE 9. Counter/Timer



** See "Initialization"

FIGURE 10. Music Synthesizer

National Semiconductor Corporation

COP401L ROMless N-Channel Microcontroller

General Description

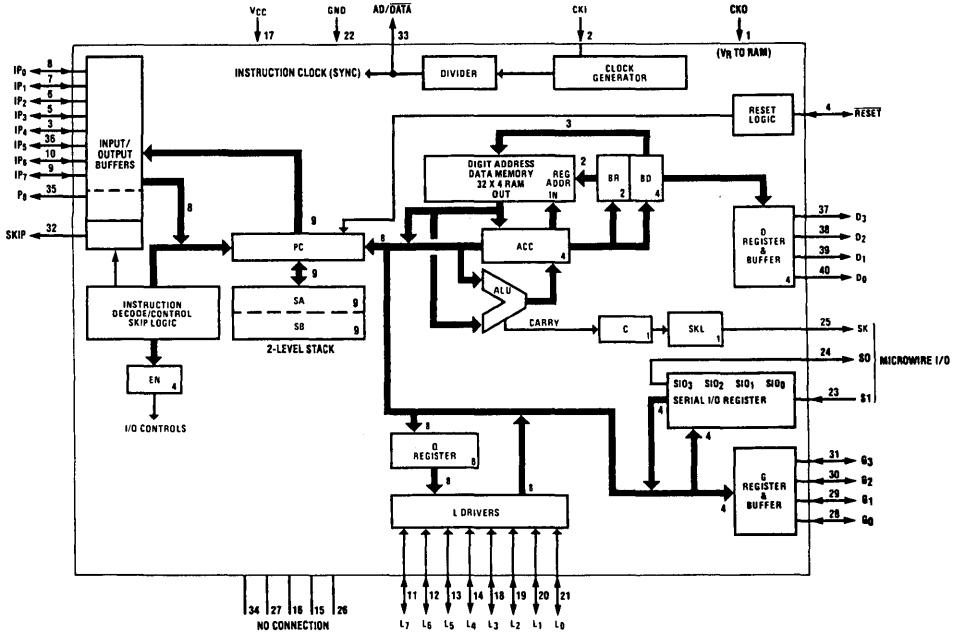
The COP401L ROMless Microcontroller is a member of the COPS™ family of microcontrollers, fabricated using N-channel, silicon gate MOS technology. The COP401L contains CPU, RAM, I/O and is identical to a COP410L device except the ROM has been removed and pins have been added to output the ROM address and to input the ROM data. In a system the COP401L will perform exactly as the COP410L. This important benefit facilitates development and debug of a COP program prior to masking the final part.

The COP401L is intended for emulation only, not intended for volume production. Use COP402 or COP404L for volume production.

Features

- Circuit equivalent of COP410L
- Low cost
- Powerful instruction set
- 512 x 8 ROM, 32 x 4 RAM
- Separate RAM power supply pin for RAM keep-alive applications
- Two-level subroutine stack
- 15 μs instruction time
- Single supply operation (4.5–9.5V)
- Low current drain (8 mA max.)
- Internal binary counter register with serial I/O
- MICROWIRE™ compatible serial I/O
- General purpose outputs
- LSTTL/CMOS compatible in and out
- Direct drive of LED digit and segment lines
- Software/hardware compatible with other members of COP400 family
- Pin-for-pin compatible with COP402 and COP404L

Block Diagram



TL/DD/6913-1

FIGURE 1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at any Pin Relative to GND	−0.5V to +10V
Ambient Operating Temperature	0°C to +70°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temp. (Soldering, 10 sec.)	300°C
Power Dissipation	0.75W at 25°C
	0.4W at 70°C
Total Source Current	120 mA
Total Sink Current	120 mA

Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 9.5V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Operating Voltage (V_{CC})	(Note 2)	4.5	9.5	V
Power Supply Ripple	Peal to Peak		0.5	V
Operating Supply Current	All Inputs and Outputs Open		8	mA
Input Voltage Levels				
CKI Input Levels				
Crystal Input				
Logic High (V_{IH})		2.0		V
Logic Low (V_{IL})		−0.3	0.4	V
$\overline{\text{RESET}}$ Input Levels	Schmitt Trigger Input			
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.6	V
IP0–IP7 Input Levels				
Logic High	V_{CC} = 9.5V	2.4		V
Logic High	V_{CC} = 5V ±5%	2.0		V
Logic Low		−0.3	0.8	V
All Other Inputs				
Logic High	V_{CC} = 9.5V	3.0		V
Logic High	V_{CC} = 5V ±5%	2.0		V
Logic Low		−0.3	0.8	V
Input Capacitance			7	pF
Output Voltage Levels				
LSTTL Operation	V_{CC} = 5V ±10%			
Logic High (V_{OH})	I_{OH} = −25 μA	2.7		V
Logic Low (V_{OL})	I_{OL} = 0.36 mA		0.4	V
IP0–IP7, P8, SKIP	(Note 1)			
Logic Low	I_{OL} = 1.6 mA		0.4	V
Output Current Levels				
Output Sink Current				
SO and SK Outputs (I_{OL})	V_{CC} = 9.5V, V_{OL} = 0.4V	1.8		mA
	V_{CC} = 4.5V, V_{OL} = 0.4V	0.9		mA
L_0–L_7 and G_0–G_3 Outputs	V_{CC} = 9.5V, V_{OL} = 0.4V	0.8		mA
	V_{CC} = 4.5V, V_{OL} = 0.4V	0.4		mA
D_0–D_3 Outputs	V_{CC} = 9.5V, V_{OL} = 1.0V	30		mA
	V_{CC} = 4.5V, V_{OL} = 1.0V	15		mA
CKO				
RAM Power Supply Input	V_R = 3.3V		1.5	mA

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 9.5V$ unless otherwise noted (Continued)

Parameter	Conditions	Min	Max	Units
Output Source Current				
D_0–D_3, G_0–G_3 Outputs (I_{OH})	$V_{CC} = 9.5V$, $V_{OH} = 2.0V$	−140	−800	μA
	$V_{CC} = 4.5V$, $V_{OH} = 2.0V$	−30	−250	μA
SO and SK Outputs (I_{OH})	$V_{CC} = 9.5V$, $V_{OH} = 4.75V$	−1.4		mA
	$V_{CC} = 4.5V$, $V_{OH} = 1.0V$	−1.2		mA
L_0–L_7 Outputs	$V_{CC} = 9.5V$, $V_{OH} = 2.0V$	−3.0	−35	mA
	$V_{CC} = 6.0V$, $V_{OH} = 2.0V$	−0.3	−25	mA
Input Load Source Current (I_{IL})	$V_{CC} = 5.0V$, $V_L = 0V$	−10	−140	μA
Total Sink Current Allowed				
All Outputs Combined			120	mA
D Port			100	mA
L_7–L_4, G Port			4	mA
L_3–L_0			4	mA
All Other Pins			1.8	mA
Total Source Current Allowed				
All I/O Combined			120	mA
L_7–L_4			60	mA
L_3–L_0			60	mA
Each L Pin			25	mA
All Other Pins			1.5	mA

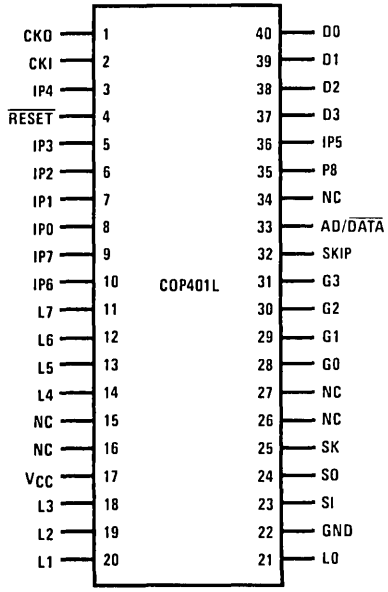
AC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 9.5V$ unless otherwise specified.

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time		15	40	μs
CKI				
Input Frequency f_I	(÷32 Mode)	0.8	2.1	MHz
Duty Cycle		30	60	%
Rise Time			120	ns
Fall Time	$f_I = 2.097$ MHz		80	ns
INPUTS:				
SI, IP7–IP0				
t_{SETUP}		2.0		μs
t_{HOLD}		1.0		μs
G_3–G_0, L_7–L_0				
t_{SETUP}		8.0		μs
t_{HOLD}		1.3		μs
OUTPUT PROPAGATION DELAY	Test Condition: C_L = pF, $V_{OUT} = 1.5V$			
SO, SK Outputs	$R_L = 20\ k\Omega$			
t_{pd1}, t_{pd0}			4.0	μs
D_3–D_0, G_3–G_0, L_7–L_0	$R_L = k\Omega$			
t_{pd1}, t_{pd0}			5.6	μs
IP7–IP0, P8, SKIP	$R_L = 5\ k\Omega$			
t_{pd1}, t_{pd0}			7.2	μs

Note 1: Pull-up resistors required.

Note 2: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Connection Diagram



TL/DD/6913-2

Order Number COP401L/N
NS Package Number N40A

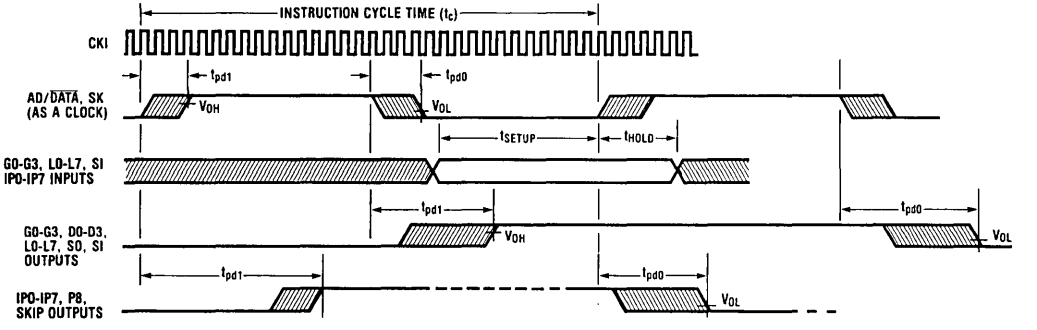
FIGURE 2

Pin Descriptions

Pin	Description
L_7–L_0	8 bidirectional I/O ports with LED segment drive
G_3–G_0	4 bidirectional I/O ports
D_3–D_0	4 general purpose outputs
SI	Serial input (or counter input)
SO	Serial output (or general purpose output)
SK	Logic-controlled clock (or general purpose output)
AD/$\overline{\text{DATA}}$	Address Out/data in flag

Pin	Description
CKI	System oscillator input
CKO	RAM power supply input
$\overline{\text{RESET}}$	System reset input
V_{CC}	Power supply
GND	Ground
IP7–IP0	8 bidirectional ROM address and data ports
P8	Most significant ROM address bit output
SKIP	Instruction skip output

Timing Diagram



TL/DD/6913-3

FIGURE 3. Input/Output

Functional Description

A block diagram of the COP401L is given in *Figure 1*. Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1" greater than 2 volts). When a bit is reset, it is a logic "0" (less than 0.8 volts).

PROGRAM MEMORY

Program Memory consists of a 512-byte external memory. As can be seen by an examination of the COP401L instruction set, these words may be program instructions, program data or ROM addressing data. Because of the special characteristics associated with the JP, JSRP, JID and LQID instructions, ROM must often be thought of as being organized into 8 pages of 64 words each.

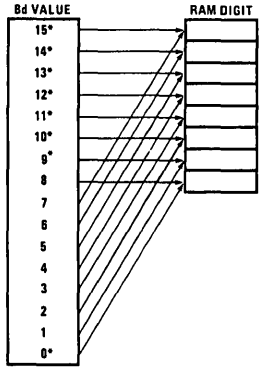
ROM addressing is accomplished by a 9-bit PC register. Its binary value selects one of the 512 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transer of control instruction, the PC register is loaded with the next sequential 9-bit binary count value. Two levels of subroutine nesting are implemented by the 9-bit subroutine save registers, SA and SB, providing a last-in, first-out (LIFO) hardware subroutine stack.

ROM instruction words are fetched, decoded and executed by the instruction Decode, Control and Skip Logic circuitry.

DATA MEMORY

Data memory consists of a 128-bit RAM, organized as 4 data registers of 8 4-bit digits. RAM addressing is implemented by a 6-bit B register whose upper 2 bits (Br) select 1 of 4 data registers and lower 3 bits of the 4-bit Bd select 1 of 8 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) is usually loaded into or from, or exchanged with, the A register (accumulator), it may also be loaded into the Q latches or loaded from the L ports. RAM addressing may also be performed directly by the XAD 3, 15 instruction. The Bd register also serves as a source register for 4-bit data sent directly to the D outputs.

The most significant bit of Bd is not used to select a RAM digit. Hence each physical digit of RAM may be selected by two different values of Bd as shown in *Figure 4* below. The skip condition for XIS and XDS instructions will be true if Bd changes between 0 and 15, but NOT between 7 and 8 (see Table 3).



TL/DD/6913-4

FIGURE 4. RAM Digit Address to Physical RAM Digit Mapping

INTERNAL LOGIC

The 4-bit A register (accumulator) is the source and destination register for most I/O, arithmetic, logic and data memory access operations. It can also be used to load the Bd portion of the B register, to load 4 bits of the 8-bit Q latch data, to input 4 bits of the 8-bit L I/O port data and to perform data exchanges with the SIO register.

A 4-bit adder performs the arithmetic and logic functions of the COP401L, storing its results in A. It also outputs a carry bit to the 1-bit C register, most often employed to indicate arithmetic overflow. The C register, in conjunction with the XAS instruction and the EN register, also serves to control the SK output. C can be outputted directly to SK or can enable SK to be a sync clock each instruction cycle time. (See XAS instruction and EN register description, below.)

The G register contents are outputs to 4 general-purpose bidirectional I/O ports.

The Q register is an internal, latched, 8-bit register, used to hold data loaded from M and A, as well as 8-bit data from ROM. Its contents are output to the L I/O ports when the L drivers are enabled under program control. (See LEI instruction.)

The 8 L drivers, when enabled, output the contents of latched Q data to the L I/O ports. Also, the contents of L may be read directly into A and M. L I/O ports can be directly connected to the segments of a multiplexed LED display (using the LED Direct Drive output configuration option) with Q data being outputted to the Sa–Sg and decimal point segments of the display.

The SIO register functions as a 4-bit serial-in/serial-out shift register or as a binary counter depending on the contents of the EN register. (See EN register description, below.) Its contents can be exchanged with A, allowing it to input or output a continuous serial data stream. SIO may also be used to provide additional parallel I/O by connecting SO to external serial-in/parallel-out shift register.

The XAS instruction copies C into the SKL Latch. In the counter mode, SK is the output of SKL in the shift register mode, SK outputs SKL ANDed with internal instruction cycle clock.

The EN register is an internal 4-bit register loaded under program control by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register (EN_3–EN_0).

1. The least significant bit of the enable register, EN_0, selects the SIO register as either a 4-bit shift register or a 4-bit binary counter. With EN_0 set, SIO is an asynchronous binary counter, *decrementing* its value by one upon each low-going pulse ("1" to "0") occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output is equal to the value of EN_3. With EN_0 reset, SIO is a serial shift register shifting left each instruction cycle time. The data present at SI goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. (See 4 below.) The SK output becomes a logic-controlled clock.
2. EN_1 is not used. It has no effect on COP401L operation.

Functional Description (Continued)

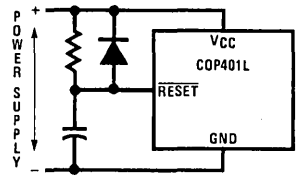
TABLE I. Enable Register Modes—Bits EN_3 and EN_0

EN_3	EN_0	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL = 1, SK = Clock If SKL = 0, SK = 0
1	0	Shift Register	Input to Shift Register	Serial Out	If SKL = 1, SK = Clock If SKL = 0, SK = 0
0	1	Binary Counter	Input to Binary Counter	0	If SKL = 1, SK = 1 If SKL = 0, SK = 0
1	1	Binary Counter	Input to Binary Counter	1	If SKL = 1, SK = 1 If SKL = 0, SK = 0

3. With EN_2 set, the L drivers are enabled to output the data in Q to the L I/O ports. Resetting EN_2 disables the L drivers, placing the L I/O ports in a high-impedance input state.
4. EN_3, in conjunction with EN_0, affects the SO output. With EN_0 set (binary counter option selected) SO will output the value loaded into EN_3. With EN_0 reset (serial shift register option selected), setting EN_3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN_3 with the serial shift register option selected disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains reset to "0". Table I provides a summary of the modes associated with EN_3 and EN_0.

INITIALIZATION

The Reset Logic will initialize (clear) the device upon power-up if the power supply rise time is less than 1 ms and greater than 1 μs. If the power supply rise time is greater than 1 ms, the user must provide an external RC network and diode to the $\overline{\text{RESET}}$ pin as shown below *(Figure 5)*. The $\overline{\text{RESET}}$ pin is configured as a Schmitt trigger input. If not used it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the $\overline{\text{RESET}}$ input, provided it stays low for at least three instruction cycle times.



TL/DD/6913–5

RC ≥ Power Supply Rise Time

FIGURE 5. Power-Up Clear Circuit

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN, and G registers are cleared. The SK output is enabled as a SYNC output, providing a pulse each instruction cycle time. *Data Memory (RAM) is not cleared upon initialization.* The first instruction at address 0 must be a CLRA.

EXTERNAL MEMORY INTERFACE

The COP401L is designed for use with an external Program Memory. This memory may be implemented using any devices having the following characteristics:

1. random addressing
2. TTL-compatible TRI-STATE® outputs
3. TTL-compatible inputs
4. access time = 5 μs max.

Typically these requirements are met using bipolar or MOS PROMs.

During operation, the address of the next instruction is sent out on P8 and IP7 through IP0 during the time that AD/$\overline{\text{DATA}}$ is high (logic "1" = address mode). Address data on the IP lines is stored into an external latch on the high-to-low transition of the AD/$\overline{\text{DATA}}$ line; P8 is a dedicated address output, and does not need to be latched. When AD/$\overline{\text{DATA}}$ is low (logic "0" = data mode), the output of the memory is gated onto IP7 through IP0, forming the input bus. Note that the AD/$\overline{\text{DATA}}$ output has a period of one instruction time, a duty cycle of approximately 50%, and specifies whether the IP lines are used for address output or instruction input.

OSCILLATOR

CKI is an external clock input signal. The external frequency is divided by 32 to give the instruction cycle time. The divide-by-32 configuration was chosen to make the COP 401L compatible with the COP404L and the COPS™ Development System. However, the ÷32 configuration is not available on the COP410L/COP411L. It is therefore possible to exactly emulate the system **speed** (cycle time), but **not** possible to drive the 401L with the system clock during emulation.

Functional Description (Continued)

CKO (RAM POWER)

CKO is configured as a RAM power supply pin (V_R), allowing its connection to a standby/backup power supply to maintain the integrity of RAM data with minimum power drain when the main supply is inoperative or shut down to conserve power. This pin must be connected to V_{CC} if the power backup feature is not used. To insure that RAM integrity is maintained, the following conditions must be met:

1. $\overline{\text{RESET}}$ must go low before V_{CC} goes below spec during power-off; V_{CC} must be within spec before $\overline{\text{RESET}}$ goes high on power-up.
2. During normal operation, V_R must be within the operating range of the chip with $(V_{CC}-1) \leq V_R \leq V_{CC}$.
3. V_R must be $\geq$ 3.3V with V_{CC} off.

INPUT/OUTPUT CONFIGURATIONS

COP401L outputs have the following configurations, illustrated in *Figure 6*:

a. Standard—an enhancement mode device to ground in conjunction with a depletion-mode device to V_{CC}, compatible with LSTTL and CMOS input requirements. (Used on D and G outputs.)

b. Open-Drain—an enhancement-mode device to ground only, allowing external pull-up as required by the user's application. (Used on IP, P and SKIP outputs.)

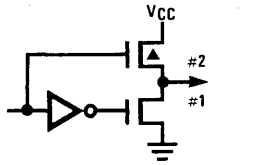
c. Push-Pull—An enhancement-mode device to ground in conjunction with a depletion-mode device paralleled enhancement-mode device to V_{CC}. This configuration has been provided to allow for fast rise and fall times when driving capacitive loads. (Used on SO and SK outputs.)

d. LED Direct Drive—an enhancement-mode device to ground and to V_{CC}, meeting the typical current sourcing requirements of the segments of an LED display. The sourcing device is clamped to limit current flow. These devices may be turned off under program control (See Functional Description, EN Register), placing the outputs in a high-impedance state to provide required LED segment blanking for a multiplexed display. (Used on L outputs.)

COP401L inputs have an on-chip depletion load device to V_{CC}.

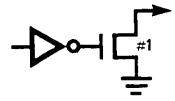
The above input and output configurations share common enhancement-mode and depletion-mode devices. Specifically, all configurations use one or more of five devices (numbered 1–5, respectively). Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figure 7* for each of these devices to allow the designer to effectively use these I/O configurations in designing a system.

An important point to remember is that even when the L drivers are disabled, the depletion load device will source a small amount of current (see *Figure 7*, Device 2); however, when the L-lines are used as inputs, the disabled depletion device can *not* be relied on to source sufficient current to pull an input to a logic "1".



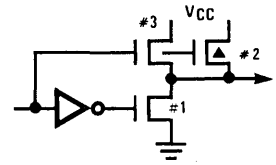
TL/DD/6913-6

a. Standard Output



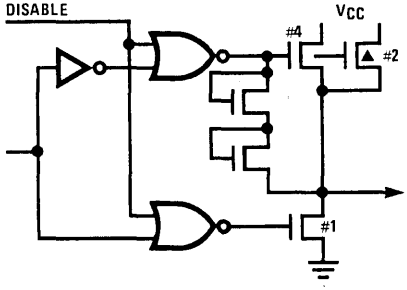
TL/DD/6913-7

b. Open-Drain Output



TL/DD/6913-8

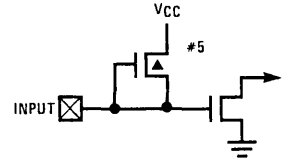
c. Push-Pull Output



TL/DD/6913-9

(▲ is Depletion Device)

d. L Output (LED)



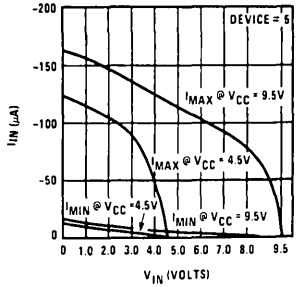
TL/DD/6913-10

e. Input with Load

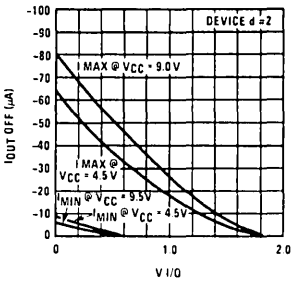
FIGURE 6. Output Configurations

Typical Performance Characteristics

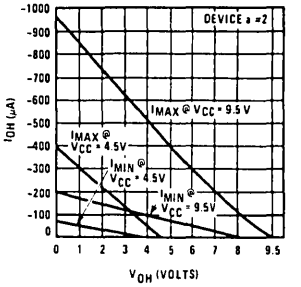
Current for Inputs with Load Device



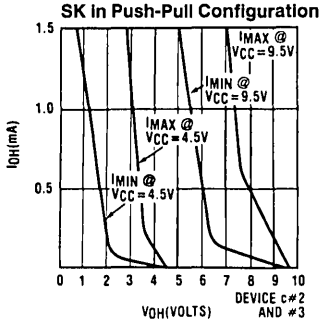
Input Current for L_0 through L_7 when Output Programmed Off by Software



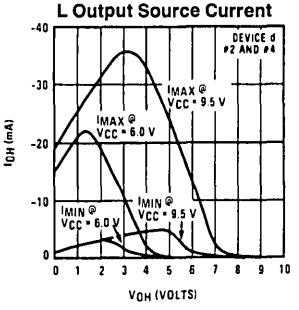
Source Current for Standard Output Configuration



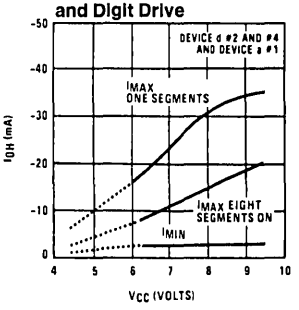
Source Current for SO and SK in Push-Pull Configuration



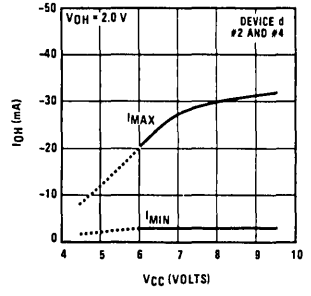
L Output Source Current



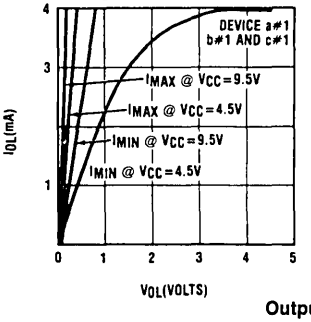
LED Output Direct Segment and Digit Drive



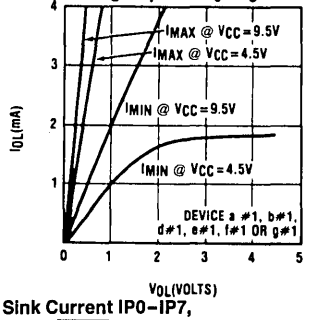
LED Output Direct Segment Drive



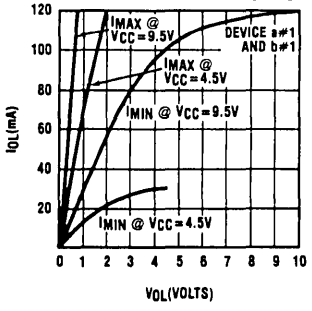
Outut Sink Current for SO and SK



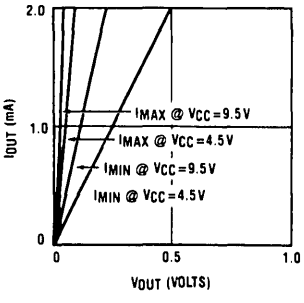
Output Sink Current for L_0 through L_7 and G_0–G_3



Output Sink Current D_0–D_3



Output Sink Current IP0–IP7, P8, SKIP, AD/$\overline{\text{DATA}}$



TL/DD/69134–11

FIGURE 7. I/O Characteristics

COP401L Instruction Set

Table II is a symbol table providing internal architecture, instruction operand and operational symbols used in the instruction set table.

Table III provides the mnemonic, operand, machine code, data flow, skip conditions, and description associated with each instruction in the COP401L instruction set.

TABLE II. COP401L Instruction Set Table Symbols

Symbol	Definition
INTERNAL ARCHITECTURE SYMBOLS	
A	4-bit Accumulator
B	6-bit RAM Address Register
Br	Upper 2 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry Register
D	4-bit Data Output Port
EN	4-bit Enable Register
G	4-bit Register to latch data for G I/O Port
L	8-bit TRI-STATE I/O Port
M	4-bit contents of RAM Memory pointed to by B Register
PC	9-bit ROM Address Register (program counter)
Q	8-bit Register to latch data for L I/O Port
SA	9-bit Subroutine Save Register A
SB	9-bit Subroutine Save Register B
SIO	4-bit Shift Register and Counter
SK	Logic-Controlled Clock Output

Symbol	Definition
INSTRUCTION OPERAND SYMBOLS	
d	4-bit Operand Field, 0-15 binary (RAM Digit Select)
r	2-bit Operand Field, 0-3 binary (RAM Register Select)
a	9-bit Operand Field, 0-511 binary (ROM Address)
y	4-bit Operand Field, 0-15 binary (Immediate Data)
RAM(s)	Contents of RAM location addressed by s
ROM(t)	Contents of ROM location addressed by t
OPERATIONAL SYMBOLS	
+	Plus
−	Minus
→	Replaces
↔	Is exchanged with
=	Is equal to
$\overline{A}$	The one's complement of A
⊕	Exclusive-OR
:	Range of values

TABLE III. COP401L Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	0011 | 0000	A + C + RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	0011 | 0001	A + RAM(B) → A	None	Add RAM to A
AISC	y	5−	0101 | y	A + y → A	Carry	Add immediate, Skip on Carry (y ≠ 0)
CLRA		00	0000 | 0000	0 → A	None	Clear A
COMP		40	0100 | 0000	$\overline{A}$ → A	None	One's complement of A to A
NOP		44	0100 | 0100	None	None	No Operation
RC		32	0011 | 0010	"0" → C	None	Reset C
SC		22	0010 | 0010	"1" → C	None	Set C
XOR		02	0000 | 0010	A ⊕ RAM(B) → A	None	Exclusive-OR RAM with A

COP410L Instruction Set (Continued)

TABLE III. COP401L Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
TRANSFER OF CONTROL INSTRUCTIONS						
JID		FF	1111 | 1111	ROM (PC_8, A,M) → $PC_{7:0}$	None	Jump Indirect (Note 2)
JMP	a	6–	0110 | 000 | a_8	a → PC	None	Jump
		– –	$a_{7:0}$			
JP	a	– –	1 | $a_{6:0}$ (pages 2,3 only)	a → $PC_{6:0}$	None	Jump within Page (Note 3)
			or			
		– –	11 | $a_{5:0}$ (all other pages)	a → $PC_{5:0}$		
JSRP	a	– –	10 | $a_{5:0}$	PC + 1 → SA → SB	None	Jump to Subroutine Page (Note 4)
				010 → $PC_{8:6}$ a → $PC_{5:0}$		
JSR	a	6–	0110 | 100 | a_8	PC + 1 → SA → SB	None	Jump to Subroutine
		– –	$a_{7:0}$	a → PC		
RET		48	0100 | 1000	SB → SA → PC	None	Return from Subroutine
RETSK		49	0100 | 1001	SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
MEMORY REFERENCE INSTRUCTIONS						
CAMQ		33	0011 | 0011	A → $Q_{7:4}$	None	Copy A, RAM to Q
		3C	0011 | 1100	RAM(B) → $Q_{3:0}$		
LD	r	–5	00 | r | 0101	RAM(B) → A Br ⊕ r → Br	None	Load RAM into A, Exclusive-OR Br with r
LQID		BF	1011 | 1111	ROM(PC_8, A, M) → Q SA → SB	None	Load Q Indirect (Note 2)
RMB	0	4C	0100 | 1100	0 → $RAM(B)_0$	None	Reset RAM Bit
	1	45	0100 | 0101	0 → $RAM(B)_1$		
	2	42	0100 | 0010	0 → $RAM(B)_2$		
	3	43	0100 | 0011	0 → $RAM(B)_3$		
SMB	0	4D	0100 | 1101	1 → $RAM(B)_0$	None	Set RAM Bit
	1	47	0100 | 0111	1 → $RAM(B)_1$		
	2	46	0100 | 0110	1 → $RAM(B)_2$		
	3	4B	0100 | 1011	1 → $RAM(B)_3$		
STII	y	7–	0111 | y	y → RAM(B) Bd + 1 → Bd	None	Store Memory Immediate and Increment Bd
X	r	–6	00 | r | 0110	RAM(B) ⟷ A Br ⊕ r → Br	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	3,15	23	0010 | 0011	RAM(3,15) ⟷ A	None	Exchange A with RAM (3,15)
		BF	1011 | 1111			
XDS	r	–7	00 | r | 0111	RAM(B) ⟷ A Bd – 1 → Bd Br ⊕ r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd, Exclusive-OR Br with r
XIS	r	–4	00 | r | 0100	RAM(B) ⟷ A Bd + 1 → Bd Br ⊕ r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd, Exclusive-OR Br with r

COP410L Instruction Set (Continued)

TABLE III. COP401L Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	0101 | 0000	A → Bd	None	Copy A to Bd
CBA		4E	0100 | 1110	Bd → A	None	Copy Bd to A
LBI	r, d	–	00 | r | (d - 1) (d = 0, 9:15)	r, d → B	Skip until not a LBI	Load B Immediate with r, d (Note 5)
LEI	y	33 6–	0011 | 0011 0110 | y	y → EN	None	Load EN Immediate (Note 6)
TEST INSTRUCTIONS						
SKC		20	0010 | 0000		C = "1"	Skip if C is True
SKE		21	0010 | 0001		A = RAM(B)	Skip if A Equals RAM
SKGZ		33 21	0011 | 0011 0010 | 0001		$G_{3:0} = 0$	Skip if G is Zero (all 4 bits)
SKGBZ		33	0011 | 0011	1st byte		Skip if G Bit is Zero
	0	01	0000 | 0001	2nd byte	$G_0 = 0$	
	1	11	0001 | 0001		$G_1 = 0$	
	2	03	0000 | 0011		$G_2 = 0$	
	3	13	0001 | 0011		$G_3 = 0$	
SKMBZ	0	01	0000 | 0001		$RAM(B)_0 = 0$	Skip if RAM Bit is Zero
	1	11	0001 | 0001		$RAM(B)_1 = 0$	
	2	03	0000 | 0011		$RAM(B)_2 = 0$	
	3	13	0001 | 0011		$RAM(B)_3 = 0$	
INPUT/OUTPUT INSTRUCTIONS						
ING		33 2A	0011 | 0011 0010 | 1010	G → A	None	Input G Ports to A
INL		33 2E	0011 | 0011 0010 | 1110	$L_{7:4}$ → RAM(B) $L_{3:0}$ → A	None	Input L Ports to RAM, A
OBD		33 3E	0011 | 0011 0011 | 1110	Bd → D	None	Output Bd to D Outputs
OMG		33 3A	0011 | 0011 0011 | 1010	RAM(B) → G	None	Output RAM to G Ports
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO (Note 2)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicitly defined (e.g., Br and Bd are explicitly defined). Bits are numbered 0 to N where 0 signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit A register.

Note 2: For additional information on the operation of the XAS, JID, and LQID instructions, see below.

Note 3: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 4: A JSRP transfers program control to subroutine page 2 (010 is loaded into the upper 3 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 5: The machine code for the lower 4 bits of the LBI instruction equals the binary value of the "d" data *minus* 1, e.g., to load the lower four bits of B (Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Note 6: Machine code for operand field y for LEI instruction should equal the binary value to be latched into EN, where a "1" or "0" in each bit of EN corresponds with the selection or deselection of a particular function associated with each bit. (See Functional Description, EN Register.)

Description of Selected Instructions

The following information is provided to assist the user in understanding the operation of several unique instructions and to provide notes useful to programmers in writing COP401L programs.

XAS INSTRUCTION

XAS (Exchange A with SIO) exchanges the 4-bit contents of the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. An XAS instruction will also affect the SK output. (See Functional Description, EN Register, above.) If SIO is selected as a shift register, an XAS instruction must be performed once every 4 instruction cycles to effect a continous data stream.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower 8 bits of the ROM address register PC with the *contents* of ROM addressed by the 9-bit word, PC_8, A, M. PC_8 is not affected by this instruction.

Note that JID requires 2 instruction cycles to execute.

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 9-bit word PC_8, A, M. LQID can be used for table lookup or code conversion such as BCD to seven-segment. The LQID instruction "pushes" the stack (PC + 1 $\rightarrow$ SA $\rightarrow$ SB) and replaces the least significant 8 bits of PC as follows: A $\rightarrow$ $PC_{7:4}$, RAM(B) $\rightarrow$ $PC_{3:0}$, leaving PC_8 unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SB $\rightarrow$ SA $\rightarrow$ PC), restoring the saved value of PC to continue sequential program execution. Since LQID pushes SA $\rightarrow$ SB, the previous contents of SB are lost. Also, when LQID pops the stack, the previously pushed contents of SA are left in SB. The net result is that the contents of SA are placed in SB (SA $\rightarrow$ SB). Note that LQID takes two instruction cycle times to execute.

INSTRUCTION SET NOTES

a. The first word of a COP401L program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, one instruction cycle time is devoted to skipping each byte of the skipped instruction. Thus all program paths except JID and LQID take the same number of cycle times whether instructions are skipped or executed. JID and LQID instructions take 2 cycles if executed and 1 cycle if skipped.

c. The ROM is organized into 8 pages of 64 words each. The Program Counter is a 9-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID or LQID instruction is located in the last word of a page, the instruction operates as if it were in the next page. For example: a JP located in the last word of a page will jump to a location in the next page. Also, a LQID or JID located in the last word of page 3 or 7 will access data in the next group of 4 pages.

Typical Applications

PROM-BASED SYSTEM

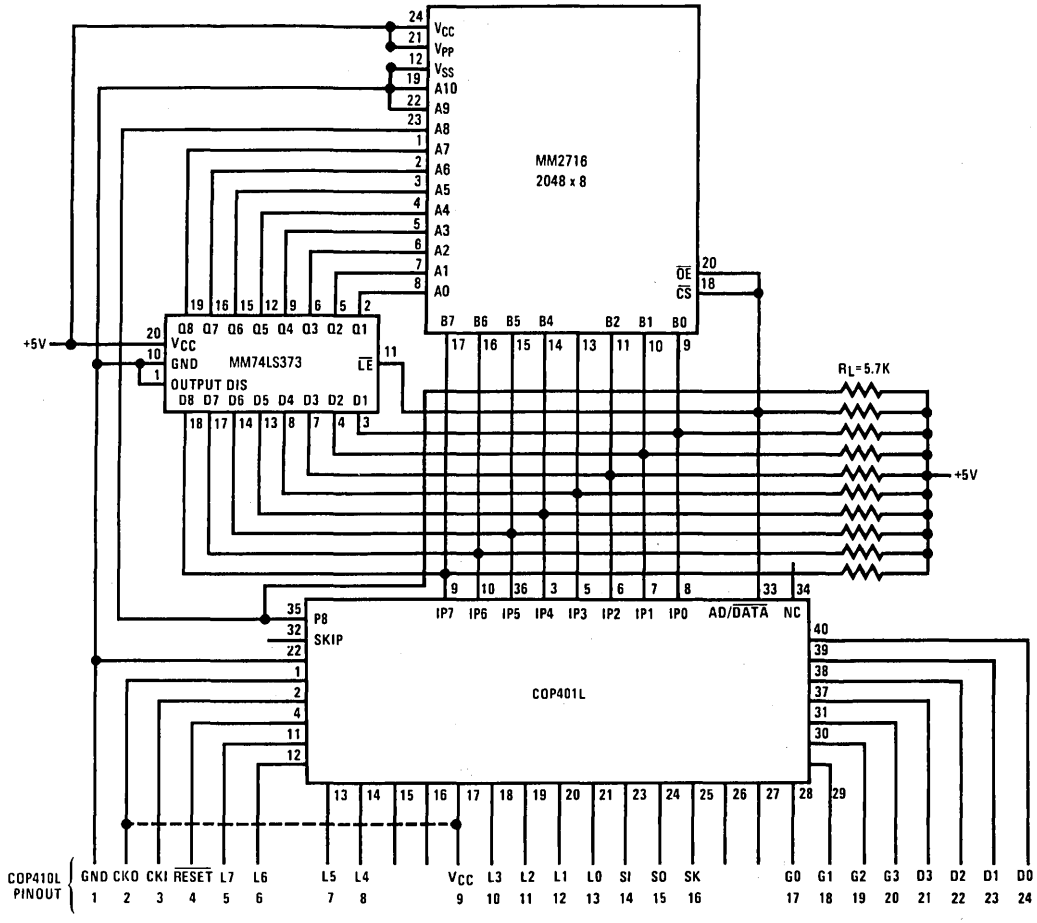
The COP401L may be used to emulate the COP410L. *Figure 8* shows the interconnect to implement a COP401L hardware emulation. This connection uses one MM5204 EPROM as external memory. Other memory can be used such as bipolar PROM or RAM.

Pins IP_7–IP_0 are bidirectional inputs and outputs. When the AD/$\overline{\text{DATA}}$ clocking output turns on, the EPROM drivers are disabled and IP_7–IP_0 output addresses. The 8-bit latch (MM74C373) latches the address to drive the memory.

When AD/$\overline{\text{DATA}}$ turns off, the EPROM is enabled and the IP_7–IP_0 pins will input the memory data. P8 outputs the most significant address bit to the memory. (SKIP output may be used for program debug if needed.)

24 of the COP401L pins may be configured exactly the same as a COP410L.

Typical Applications (Continued)



TL/DD/6913–12

FIGURE 8. COP401L Used to Emulate a COP410L

Option Table

COP401L MASK OPTIONS

The following COP410L options have been implemented in this basic version of the COP401L.

Option Value	Comment
Option 1 = 0	Ground—no option
Option 2 = 1	CKO is RAM power supply input
Option 3 = N/A	CKI is external clock divide-by-32 (not available on COP410L)
Option 4 = 0	Reset has load to V_{CC}
Option 5 = 2	
Option 6 = 2	L outputs are LED direct-drive
Option 7 = 2	
Option 8 = 2	
Option 9 = 1	V_{CC} pin 4.5V to 9.5V operation
Option 10 = 2	
Option 11 = 2	L outputs are LED direct-drive
Option 12 = 2	
Option 13 = 2	

Option Value	Comment
Option 14 = 0	SI has load to V_{CC}
Option 15 = 2	SO is push-pull output
Option 16 = 2	SK is push-pull output
Option 17 = 0	
Option 18 = 0	G outputs are standard
Option 19 = 0	
Option 20 = 0	
Option 21 = 0	
Option 22 = 0	D outputs are standard
Option 23 = 0	very high current
Option 24 = 0	
Option 25 = 0	L
Option 26 = 0	G Have standard TTL input levels
Option 27 = 0	SI
Option 28 = N/A	40-pin package

National Semiconductor Corporation

COP401L-X13/COP401L-R13 ROMless N-Channel Microcontroller

General Description

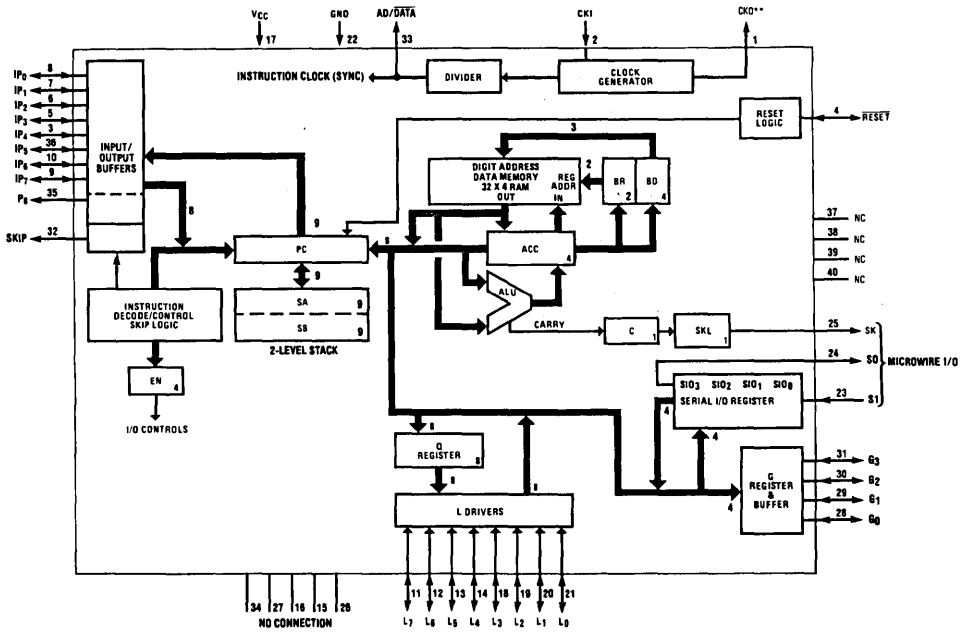
The COP401L-X13/COP401L-R13 ROMless Microcontrollers are members of the COPS™ family of microcontrollers, fabricated using N-channel, silicon gate MOS technology. The COP401L-X13/COP401L-R13 contain CPU, RAM, I/O and are identical to a COP413L device except the ROM has been removed and pins have been added to output the ROM address and to input the ROM data. In a system the COP401L-X13/COP401L-R13 will perform exactly as the COP413L. This important benefit facilitates development and debug of a COP program prior to masking the final part.

There are two clock oscillator configurations available. The crystal oscillator configuration is called COP401L-X13 and the RC oscillator configuration is called COP401L-R13.

Features

- Circuit equivalent of COP413L
- Low cost
- Powerful instruction set
- 512 × 8 ROM, 32 × 4 RAM
- Two-level subroutine stack
- 16 μs instruction time
- Single supply operation (4.5–5.5V)
- Low current drain (8 mA max)
- Internal binary counter register with serial I/O
- MICROWIRE™ compatible serial I/O
- General purpose outputs
- Software/hardware compatible with other members of COP400 family
- Pin-for-pin compatible with COP402 and COP404L
- High noise immunity inputs (V_{IL} = 1.2V, V_{IH} = 3.6V)

Block Diagram



TL/DD/8528-1

**COP401L-X13 only

FIGURE 1

COP401L-X13/COP401L-R13 Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin Relative to GND	−0.3 to +7V
Ambient Operating Temperature	0°C to +70°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temp. (Soldering, 10 seconds)	300°C
Power Dissipation COP413L	0.3 Watt at 70°C
Total Source Current	25 mA
Total Sink Current	40 mA

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $0°C \le T_A \le +70°$, $4.5V \le V_{CC} \le 5.5V$ unless otherwise noted.

Parameter	Conditions	Min	Max	Units
Standard Operating Voltage (V_{CC})	(Note 1)	4.5	5.5	V
Power Supply Ripple	Peak to Peak		0.4	V
Operating Supply Current	All Inputs and Outputs Open		8	mA
Input Voltage Levels				
CKI Input Levels				
Ceramic Resonator Input (÷8)				
Logic High (V_{IH})		3.0		V
Logic Low (V_{IL})			0.4	V
CKI (RC), Reset Input Levels	(Schmitt Trigger Input)			
Logic High		0.7 V_{CC}		V
Logic Low			0.6	V
SO Input Level (Test Mode)	(Note 2)	2.5		V
IP0–IP7, SI Input Level				
Logic High	(TTL Level)	2.0		V
Logic Low			0.8	V
L, G Inputs				
Logic High	(High Trip Levels)	3.6		V
Logic Low			1.2	V
Input Capacitance			7	pF
Reset Input Leakage		−1	+1	μA
Output Current Levels				
Output Sink Current (I_{OL})				
SO and SK Outputs	V_{OL} = 0.4V	0.9		mA
L0–L7 Outputs, G0–G3	V_{OL} = 0.4V	0.4		mA
CKO	V_{OL} = 0.4V	0.2		mA
IP0–IP7, P8, SKIP, AD/$\overline{DATA}$	V_{OL} = 0.4V	1.6		mA
Output Source Current (I_{OH})				
L0–L7 G0–G3, SO, SK	V_{OH} = 2.4V	−25		μA
IP0–IP7, P8, SKIP, AD/$\overline{DATA}$	V_{OH} = 2.4V	−25		μA
SO, SK	V_{OH} = 1.0V	−1.2		mA
IP0–IP7, P8, SKIP, AD/$\overline{DATA}$	V_{OH} = 1.0V	−1.2		mA
SI Input Load Source Current	V_{IL} = 0V	−10	−140	μA
Total Sink Current Allowed				
L7–L4, G Port			4	mA
L3–L0			4	mA
Any Other Pin			2.0	mA
Total Source Current Allowed				
Each Pin			1.5	mA

Note 1: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

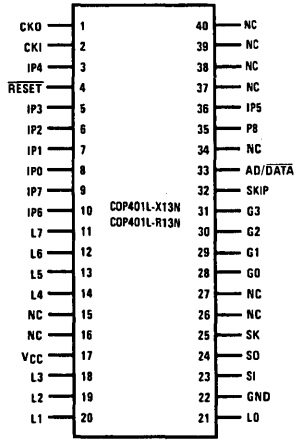
Note 2: SO output "0" level must be less than 0.8V for normal operation.

AC Electrical Characteristics $0°C \le T_A \le 70°C, 4.5V \le V_{CC} \le 5.5V$

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time - t_c		16	40	μs
CKI				
Input Frequency - fi	÷ 8 Mode	0.2	0.5	MHz
Duty Cycle		30	60	%
Rise Time	fi = 0.5 MHz		500	ns
Fall Time			200	ns
CKI Using RC (÷ 4)	R = 56 kΩ ±5% C = 100 pF ±10%			
Instruction Cycle Time (Note 1)		16	28	μs
Inputs:				
G3–G0, L7–L0				
t_{SETUP}			8.0	μs
t_{HOLD}			1.3	μs
SI, IP0–IP7				
t_{SETUP}			2.0	μs
t_{HOLD}			1.0	μs
Output Propagation Delay	Test Condition: C_L = 50 pF, V_{OUT} = 1.5V			
SO, SK Outputs	R_L = 20 kΩ			
tpd1, tpd0			4.0	μs
L, G Outputs	R_L = 20 kΩ			
tpd1, tpd0			5.6	μs
IP0–IP7, P8, SKIP	R_L = 5 kΩ			
tpd1, tpd0			7.2	μs

Note 1: Variation due to the device included.

Connection Diagram



TL/DD/8528-2

FIGURE 2

Order Number COP401L-X13N or COP401L-R13N
See NS Package Number N40A

Pin Descriptions

Pin	Description
L_7–L_0	8 bidirectional I/O ports
G_3–G_0	4 bidirectional I/O ports
SI	Serial input (or counter input)
SO	Serial output (or general purpose output)
SK	Logic-controlled clock (or general purpose output)
AD/$\overline{DATA}$	Address out/data in flag
CKI	System oscillator input
CKO	System oscillator output or NC
$\overline{RESET}$	System reset input
V_{CC}	Power supply
GND	Ground
IP7–IP0	8 bidirectional ROM address and data ports
P8	Most significant ROM address bit output
SKIP	Instruction skip output

Timing Waveform

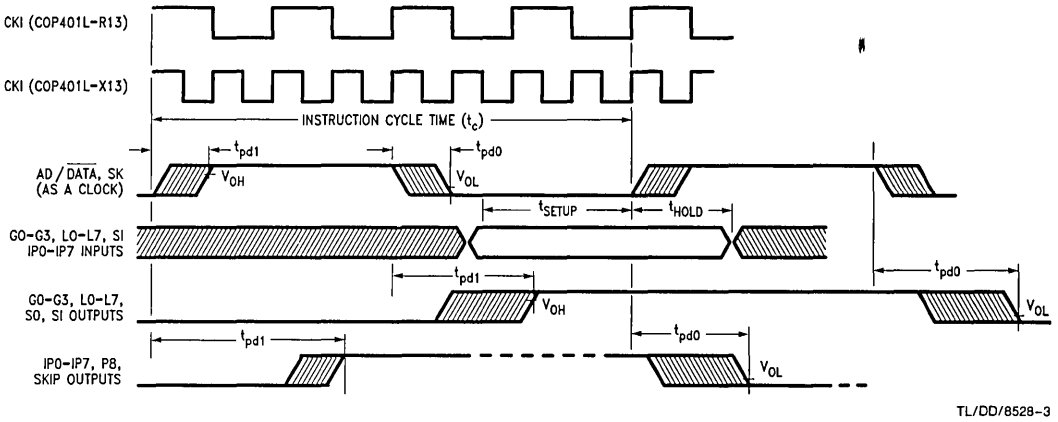


FIGURE 3. Input/Output Timing Diagram

Development Support

The MOLE (Microcontroller On Line Emulator) is a low cost development system and real time emulator for COP's products. They also include TMP, 8050, and the new 16-bit HPC Microcontroller Family. The MOLE provides effective support for the development of both software and hardware in the user's application.

The purpose of the MOLE is to provide a tool to write and assemble code, emulate code for the target microcontroller and assist in debugging of the system.

The MOLE can be connected to various hosts, IBM PC STARPLEX™, Kaypro, Apple, and Intel Systems, via RS-232 port. This link facilitate the up loading/down loading of code, supports host assembly and mass storage.

The MOLE consists of three parts; brain, personality and optional host software.

The brain board is the computing engine of the system. It is a self contained computer with its own firmware which provides for all system operation, emulation control, communication, from programming and diagnostic operation. It has three serial ports which can be connected to a terminal, host system, printer, modem or to other MOLE's in a multi-MOLE environment.

The personality board contains the necessary hardware and firmware needed to emulate the target microcontroller. The emulation cable which replaces the target controller attaches to this board. The software contains a cross assembler and communications program for up loading and down loading code from the MOLE.

MOLE Ordering Information

P/N	Description
MOLE-BRAIN	MOLE Computer Board
MOLE-COPS-PB1	COPS' Personality Board
MOLE-XXX-YYY	Optional Software

Where XXX = COPS, TMP, 8050, or HPC

YYY = Host System, IBM, APPLE, KAY (Kaypro), CP/M

Functional Description

A block diagram of the COP401L-X13/COP401L-R13 is given in *Figure 1*. Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1" (greater than 2 volts). When a bit is reset, it is a logic "0" (less than 0.8 volts).

PROGRAM MEMORY

Program Memory consists of a 512-byte external memory. As can be seen by an examination of the COP401L-X13/ COP401L-R13 instruction set, these words may be program instructions, program data or ROM addressing data. Because of the special characteristics associated with the JP, JSRP, JID and LQID instructions, ROM must often be thought of as being organized into 8 pages of 64 words each.

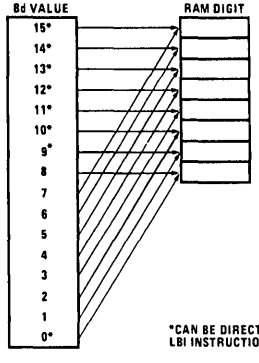
ROM addressing is accomplished by a 9-bit PC register. Its binary value selects one of the 512 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 9-bit binary count value. Two levels of subroutine nesting are implemented by the 9-bit subroutine save registers, SA and SB, providing a last-in, first-out (LIFO) hardware subroutine stack.

ROM instruction words are fetched, decoded and executed by the instruction Decode, Control and Skip Logic circuitry.

DATA MEMORY

Data memory consists of a 128-bit RAM, organized as 4 data registers of 8 4-bit digits. RAM addressing is implemented by a 6-bit B register whose upper 2 bits (Br) select 1 of 4 data registers and lower 3 bits of the 4-bit Bd select 1 of 8 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) is usually loaded into or from, or exchanged with, the A register (accumulator), it may also be loaded into the Q latches or loaded from the L ports. RAM addressing may also be performed directly by the XAD 3,15 instruction.

The most significant bit of Bd is not used to select a RAM digit. Hence each physical digit of RAM may be selected by two different values of Bd as shown in *Figure 4* below. The skip condition for XIS and XDS instructions will be true if Bd changes between 0 and 15, but NOT between 7 and 8 (see Table 3).



TL/DD/8528–4

FIGURE 4. RAM Digit Address to Physical RAM Digit Mapping

INTERNAL LOGIC

The 4-bit A register (accumulator) is the source and destination register for most I/O, arithmetic, logic and data memory access operations. It can also be used to load the Bd portion of the B register, to load 4 bits of the 8-bit Q latch data, to input 4 bits of the 8-bit L I/O port data and to perform data exchanges with the SIO register.

A 4-bit adder performs the arithmetic and logic functions of the COP401L-X13/COP401L-R13, storing its results in A. It also outputs a carry bit to the 1-bit C register, most often employed to indicate arithmetic overflow. The C register, in conjunction with the XAS instruction and the EN register, also serves to control the SK output. C can be outputted directly to SK or can enable SK to be a sync clock each instruction cycle time. (See XAS instruction and EN register description, below).

The G register contents are outputs to 4 general-purpose bidirectional I/O ports.

The Q register is an internal, latched, 8-bit register, used to hold data loaded from M and A, as well as 8-bit data from ROM. Its contents are output to the L I/O ports when the L drivers are enabled under program control. (See LEI instruction.)

The 8 L drivers, when enabled, output the contents of latched Q data to the L I/O ports. Also, the contents of L may be read directly into A and M.

The SIO register functions as a 4-bit serial-in-/serial-out shift register or as a binary counter depending on the contents of the EN Register. (See EN register description, below.) Its contents can be exchanged with A, allowing it to input or output a continuous serial data stream. SIO may also be used to provide additional parallel I/O by connecting SO to external serial-in/parallel-out shift registers.

The XAS instruction copies C into the SKL Latch. In the counter mode, SK is the output of SKL in the shift register mode, SK outputs SKL ANDed with internal instruction cycle clock.

The EN register is an internal 4-bit register loaded under program control by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN registers (EN_3–EN_0).

1. The least significant bit of the enable register, EN_0, selects the SIO Register as either a 4-bit shift register or a 4-bit binary counter. With EN_0 set, SIO is an asynchronous binary counter, decrementing its value by one upon each low-going pulse ("1" to "0") occurring on the SI Input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO Output is equal to the value of EN_3. With EN_0 reset, SIO is a serial shift register shifting left each instruction cycle time. The data pesent at SI goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. (See 4 below.) The SK output becomes a logic-controlled clock.
2. EN_1 is not used. It has no effect on COP401L-X13/ COP401L-R13 operation.
3. With EN_2 set, the L drivers are enabled to output the data in Q to the L I/O ports. Resetting EN_2 disables the L drivers, placing the L I/O ports in a high impedance input state.

Functional Description (Continued)

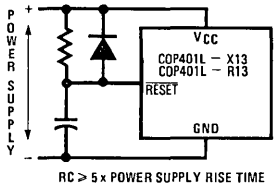
TABLE I. Enable Register Modes - Bits EN_3 and EN_0

EN_3	EN_0	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL=1, SK=Clock If SKL=0, SK=0
1	0	Shift Register	Input to Shift Register	Serial Out	If SKL=1, SK=Clock If SKL=0, SK=0
0	1	Binary Counter	Input to Binary Counter	0	If SKL=1, SK=1 If SKL=0, SK=0
1	1	Binary Counter	Input to Binary Counter	1	If SKL=1, SK=1 If SKL=0, SK=0

4. EN_3, in conjunction with EN_0, affects the SO output. With EN_0 set (binary counter option selected) SO will output the value loaded into EN_3. With EN_0 reset (serial shift register option selected), setting EN_3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN_3 with the serial shift register option selected disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains reset to "0". Table 1 provides a summary of the modes associated with EN_3 and EN_0.

INITIALIZATION

The Reset Logic will initialize (clear) the device upon power-up if the power supply rise time is less than 1 ms and greater than 1 μs. If the power supply rise time is greater than 1 ms, the user must provide an external RC network and diode to the $\overline{RESET}$ pin as shown below (*Figure 5*). The $\overline{RESET}$ pin is configured as a Schmitt trigger input. If not used it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the $\overline{RESET}$ input, provided it stays low for at least three instruction cycle times.



TL/DD/8528–5

Figure 5. Power-Up Clear Circuit

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, EN, and G registers are cleared. The SK output is enabled as a SYNC output, providing a pulse each instruction cycle time. *Data Memory (RAM) is not cleared upon initialization.* The first instruction at address 0 must be a CLRA.

EXTERNAL MEMORY INTERFACE

The COP401L-X13/COP401L-R13 is designed for use with an external Program Memory. This memory may be implemented using any devices having the following characteristics:

1. random addressing
2. TTL-compatible TRI-STATE® outputs
3. TTL-compatible inputs
4. access time = 5 μs max.

Typically these requirements are met using bipolar or MOS PROMs.

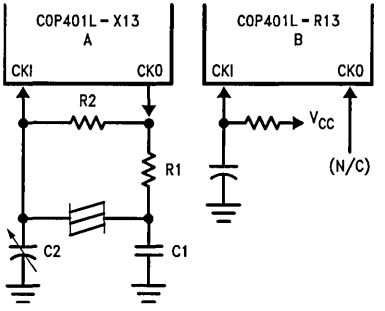
During operation, the address of the next instruction is sent out on P8 and IP7 through IP0 during the time that AD/$\overline{DATA}$ is high (logic "1" = address mode). Address data on the IP lines is stored into an external latch on the high-to-low transition of the AD/$\overline{DATA}$ line; P8 is a dedicated address output, and does not need to be latched. When AD/$\overline{DATA}$ is low (logic "0" = data mode), the output of the memory is gated onto IP7 through IP0, forming the input bus. Note that the AD/$\overline{DATA}$ output has a period of one instruction time, a duty cycle of approximately 50%, and specifies whether the IP lines are used for address output or instruction input.

OSCILLATOR

There are two basic clock oscillator configurations available as shown by *Figure 6*.

a. The COP401L-X13 is a Resonator Controlled Oscillator. CKI and CKO are connected to an external ceramic resonator. The instruction cycle frequency equals the resonator frequency divided by 8.

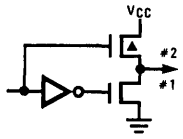
b. The COP401L-R13 is a RC Controlled Oscillator. CKI is configured as a single pin RC controlled Schmitt trigger oscillator. The instruction cycle equals the oscillation frequency divided by 4. CKO becomes no connection.



TL/DD/8528–6

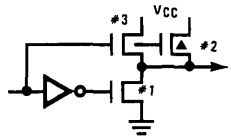
FIGURE 6. COP401L-X13/COP401L-R13 Oscillator

Functional Description (Continued)



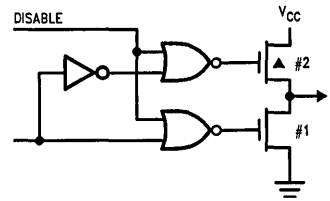
TL/DD/8528-7

a. Standard Output



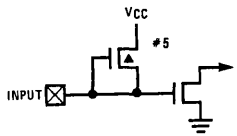
TL/DD/8528-8

b. Push-Pull Output



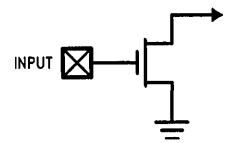
TL/DD/8528-9

c. Standard L Ouput



TL/DD/8528-10

d. Input With Load



TL/DD/8528-11

e. Hi-Z Input

FIGURE 7. Input and Output Configurations

Ceramic Resonator Oscillator

Resonator Value	Component Values			
	R1 (Ω)	R2 (Ω)	C1 (pF)	C2 (pF)
455 kHz	4.7k	1M	220	220

RC Controlled Oscillator

R (kΩ)	C (pF)	Instruction Cycle Time (in μs)
51	100	19 ± 15%
82	56	19 ± 13%

Note: 200 kΩ ≥ R ≥ 25 kΩ
220 pF ≥ C ≥ 50 pF

I/O CONFIGURATIONS

COP401L-X13/COP401L-R13 inputs and outputs have the following configurations, illustrated in *Figure 7*.

a. G0–G3—an enhancement mode device to ground in conjunction with depletion-mode device to V_{CC}.

b. SO, SK, IP0–IP7, P8, SKIP, AD/$\overline{\text{DATA}}$—an enhancement mode device to ground in conjunction with a depletion-mode device paralleled by an enhancement-mode device to V_{CC}. This configuration has been provided to allow for fast rise and fall times when driving capacitive loads.

c. L0–L7—same as a, but may be disabled.

d. SI has on-chip depletion load device to V_{CC}.

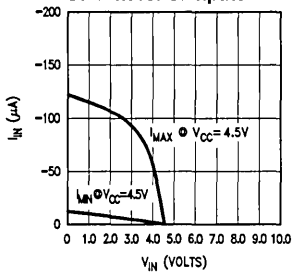
e. $\overline{\text{RESET}}$ has a Hi-Z input which must be driven to a "1" or "0" by external components.

Curves are given in *Figure 8* to allow the designer to effectively use the I/O configurations in designing a system.

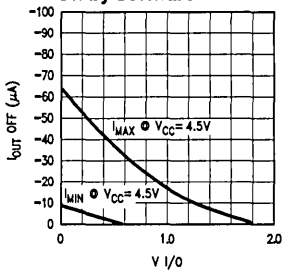
An important point to remember is that even when the L drivers are disabled, the depletion load device will source a small amount of current, however, when the L lines are used as inputs, the disabled depletion device can not be relied on to source sufficient current to pull an input to a logic "1".

Typical Performance Characteristics

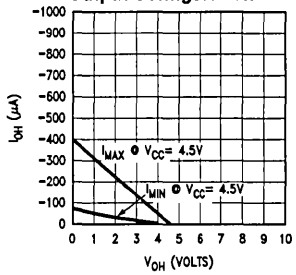
Current for SI Inputs



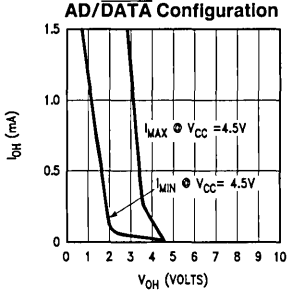
Input Current for L_0 through L_7 when Output Programmed Off by Software



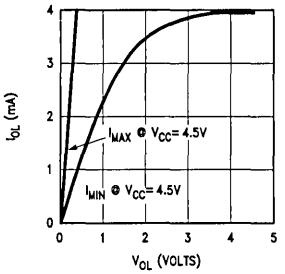
Source Current for L7–L0, G3–G0 Output Configuration



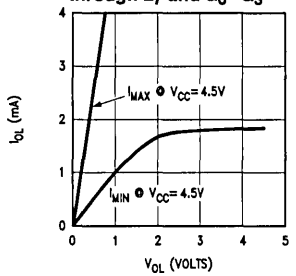
Source Current for SO, SK, IP0, IP7, P8, SKIP, AD/$\overline{DATA}$ Configuration



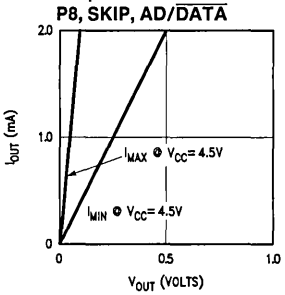
Output Sink Current for SO and SK



Output Sink Current for L_0 through L_7 and G_0–G_3



Output Sink Current IP0–IP7, P8, SKIP, AD/$\overline{DATA}$



TL/DD/8528–12

FIGURE 8. I/O Characteristics

1

COP401L-X13/COP401L-R13 Instruction Set

Table II is a symbol table providing internal architecture, Instruction operand and operational symbols used in the instruction set table.

Table III provides the mnemonic, operand, machine code, data flow, skip conditions, and description associated with each instruction in the COP401L-X13/COP401L-R13 instruction set.

TABLE II. COP401L-X13/COP401L-R13 Instruction Set Table Symbols

Symbol	Definition
Internal Architecture Symbols	
A	4-bit Accumulator
B	6-bit RAM Address Register
Br	Upper 2 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry Register
EN	4-bit Enable Register
G	4-bit Register to latch data for G I/O Port
L	8-bit TRI-STATE I/O Port
M	4-bit contents of RAM Memory pointed to by B Register
PC	9-bit ROM Address Register (program counter)
Q	8-bit Register to latch data for L I/O Port
SA	9-bit Subroutine Save Register A
SB	9-bit Subroutine Save Register B
SIO	4-bit Shift Register and Counter
SK	Logic Controlled Clock Output
Instruction Operand Symbols	
d	4-bit Operand Field, 0–15 binary (RAM Digit Select)
r	2-bit Operand Field, 0–3 binary (RAM Register Select)
a	9-bit Operand Field, 0–511 binary (ROM Address)
y	4-bit Operand Field, 0–15 binary (Immediate Data)
RAM(s)	Contents of RAM location addressed by s
ROM(t)	Contents of ROM location addressed by t
Operational Symbols	
+	Plus
−	Minus
→	Replaces
← →	Is exchanged with
=	Is equal to
$\overline{A}$	The one's complement of A
⊕	Exclusive-OR
:	Range of values

TABLE III. COP401L-X13/COP401L-R13 Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	0011 | 0000	A+C+RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	0011 | 0001	A+RAM(B) → A	None	Add RAM to A
AISC	y	5–	0101 | y	A+y → A	Carry	Add Immediate, Skip on Carry (y≠0)
CLRA		00	0000 | 0000	0 → A	None	Clear A
COMP		40	0100 | 0000	$\overline{A}$ → A	None	One's complement of A to A
NOP		44	0100 | 0100	None	None	No Operation
RC		32	0011 | 0010	"0" → C	None	Reset C
SC		22	0010 | 0010	"1" → C	None	Set C
XOR		02	0000 | 0010	A⊕RAM(B) → A	None	Exclusive-OR RAM with A
TRANSFER OF CONTROL INSTRUCTIONS						
JID		FF	1111 | 1111	ROM(PC_8,A,M) → $PC_{7:0}$	None	Jump Indirect (Note 2)
JMP	a	6–	0110 | 000 | a_8	a → PC	None	Jump
		–	$a_{7:0}$			
JP	a	–	1 | $a_{6:0}$ (pages 2, 3 only) or	a → $PC_{6:0}$	None	Jump within-Page (Note 3)
		–	11 | $a_{5:0}$ (all other pages)	a → $PC_{5:0}$		
JSRP	a	–	10 | $a_{5:0}$	PC+1 → SA → SB 010 → $PC_{8:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 4)
JSR	a	6–	0110 | 100 | a_8	PC+1 → SA → SB a → PC	None	Jump to Subroutine
		–	$a_{7:0}$			
RET		48	0100 | 1000	SB → SA → PC	None	Return from Subroutine
RETSK		49	0100 | 1001	SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
MEMORY REFERENCE INSTRUCTIONS						
CAMQ		33	0011 | 0011	A → $Q_{7:4}$	None	Copy A, RAM to Q
		3C	0011 | 1100	RAM(B) → $Q_{3:0}$		
LD	r	–5	00 | r | 0101	RAM(B) → A Br⊕r → Br	None	Load RAM into A, Exclusive-OR Br with r
LQID		BF	1011 | 1111	ROM(PC_8, A,M) → Q SA → SB	None	Load Q Indirect (Note 2)
RMB	0	4C	0100 | 1100	0 → $RAM(B)_0$	None	Reset RAM Bit
	1	45	0100 | 0101	0 → $RAM(B)_1$		
	2	42	0100 | 0010	0 → $RAM(B)_2$		
	3	43	0100 | 0011	0 → $RAM(B)_3$		
SMB	0	4D	0100 | 1101	1 → $RAM(B)_0$	None	Set RAM Bit
	1	47	0100 | 0111	1 → $RAM(B)_1$		
	2	46	0100 | 0110	1 → $RAM(B)_2$		
	3	4B	0100 | 1011	1 → $RAM(B)_3$		
STII	y	7–	0111 | y	y → RAM(B) Bd+1 → Bd	None	Store Memory Immediate and Increment Bd
X	r	–6	00 | r | 0110	RAM(B) ⟷ A Br⊕r → Br	None	Exchange RAM with A, Exclusive-OR Br with r

TABLE III. COP401L-X13/COP401L-R13 Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
MEMORY REFERENCE INSTRUCTIONS (Continued)						
XAD	3,15	23	0010 | 0011	RAM(3,15) ⟷ A	None	Exchange A with RAM (3,15)
		BF	1011 | 1111			
XDS	r	−7	00 | r | 0111	RAM(B) ⟷ A Bd − 1 → Bd Br ⊕ r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd, Exclusive-OR Br with r
XIS	r	−4	00 | r | 0100	RAM(B) ⟷ A Bd + 1 → Bd Br ⊕ r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd, Exclusive-OR Br with r
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	0101 | 0000	A → Bd	None	Copy A to Bd
CBA		4E	0100 | 1110	Bd → A	None	Copy Bd to A
LBI	r,d	–	00 | r | (d−1) (d = 0,9:15)	r,d → B	Skip until not a LBI	Load B immediate with r,d (Note 5)
LEI	y	33	0011 | 0011	y → EN	None	Load EN Immediate (Note 6)
		6−	0110 | y			
TEST INSTRUCTIONS						
SKC		20	0010 | 0000		C = "1"	Skip if C is True
SKE		21	0010 | 0001		A = RAM(B)	Skip if A Equals RAM
SKGZ		33	0011 | 0011		$G_{3:0} = 0$	Skip if G is Zero (all 4 bits)
		21	0010 | 0001			
SKGBZ		33	0011 | 0011	1st byte		Skip if G Bit is Zero
	0	01	0000 | 0001	2nd byte	$G_0 = 0$	
	1	11	0001 | 0001		$G_1 = 0$	
	2	03	0000 | 0011		$G_2 = 0$	
	3	13	0001 | 0011		$G_3 = 0$	
SKMBZ	0	01	0000 | 0001		$RAM(B)_0 = 0$	Skip if RAM Bit is Zero
	1	11	0001 | 0001		$RAM(B)_1 = 0$	
	2	03	0000 | 0011		$RAM(B)_2 = 0$	
	3	13	0001 | 0011		$RAM(B)_3 = 0$	
INPUT/OUTPUT INSTRUCTIONS						
ING		33	0011 | 0011	G → A	None	Input G Ports to A
		2A	0010 | 1010			
INL		33	0011 | 0011	$L_{7:4}$ → RAM(B) $L_{3:0}$ → A	None	Input L Ports to RAM, A
		2E	0010 | 1110			
OMG		33	0011 | 0011	RAM(B) → G	None	Output RAM to G Ports
		3A	0011 | 1010			
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO (Note 2)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicitly defined (e.g., Br and Bd are explicitly defined) Bits are numbered 0 to N where 0 signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit A register.

Note 2: For additional information on the operation of the XAS, JID, and LQID instructions, see below.

Note 3: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 4: A JSRP transfers program control to subroutine page 2 (010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 5: The machine code for the lower 4 bits of the LBI instruction equals the binary value of the "d" data *minus 1* e.g., to load the lower four bits of B (Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Note 6: Machine code for operand field y for LEI instruction should equal the binary value to be latched into EN, where a "1" or "0" in each bit of EN corresponds with the selection or deselection of a particular function associated with each bit. (See Functional Description, EN Register.)

Description of Selected Instructions

The following information is provided to assist the user in understanding the operation of several unique instructions and to provide notes useful to programmers in writing COP401L-X13/C0P401L-R13 programs.

XAS INSTRUCTION

XAS (Exchange A with SIO) exchanges the 4-bit contents of the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. An XAS instruction will also affect the SK output. (See Functional Description, EN Register, above.) If SIO is selected as a shift register, an XAS instruction must be performed once every 4 instruction cycles to effect a continuous data stream.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower 8 bits of the ROM address register PC with the *contents* of ROM addressed by the 9-bit word, PC_8, A, M. PC_8 is not affected by this instruction.

Note that JID requires 2 instruction cycles to execute.

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 9-bit word PC_8, A, M. LQID can be used for table lookup or code conversion such as BCD to seven-segment. The LQID instruction "pushes" the stack (PC + 1 $\rightarrow$ SA $\rightarrow$ SB) and replaces the least significant 8 bits of PC as follows: A $\rightarrow$ $PC_{7:4}$, RAM (B) $\rightarrow$ $PC_{3:0}$, leaving PC_8 unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SB $\rightarrow$ SA $\rightarrow$ PC), restoring the saved value of PC to continue sequential program execution. Since LQID pushes SA $\rightarrow$ SB, the previous contents of SB are lost. Also, when LQID pops the stack, the previously pushed contents of SA are left in SB. The net result is that the contents of SA are placed in SB (SA $\rightarrow$ SB). Note that LQID takes two instruction cycle times to execute.

INSTRUCTION SET NOTES

a. The first word of a COP401L-X13/COP401L-R13 program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, one instruction cycle time is devoted to skipping each byte of the skipped instruction. Thus all program paths except JID and LQID take the same number of cycle times whether instructions are skipped or executed. JID and LQID instructions take 2 cycles if executed and 1 cycle if skipped.

c. The ROM is organized into 8 pages of 64 words each. The Program Counter is a 9-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID or LQID instruction is located in the last word of a page, the instruction operates as if it were in the next page. For example: a JP located in the last word of a page will jump to a location in the next page. Also, a LQID or JID located in the last word of page 3 or will access data in the next group of 4 pages.

COPS Programming Manual

For detailed information on writing COPS programs, the COPS Programming Manual 424410284-001 provides an in-depth discussion of the COPS architecture, instruction set and general techniques of COPS programming. This manual is written with the programmer in mind.

Typical Applications

PROM-Based System

The COP401L-X13/COP401L-R13 may be used to emulate the COP413L. *Figure 9* shows the interconnect to implement a COP401L-X13/COP401L-R13 hardware emulation. This connection uses one MM2716 EPROM as external memory. Other memory can be used such as bipolar PROM or RAM.

Pins IP_7–IP_0 are bidirectional inputs and outputs. When the AD/$\overline{DATA}$ clocking output turns on, the EPROM drivers are disabled and IP_7–IP_0 output addresses. The 8-bit latch (MM74C373) latches the addresses to drive the memory.

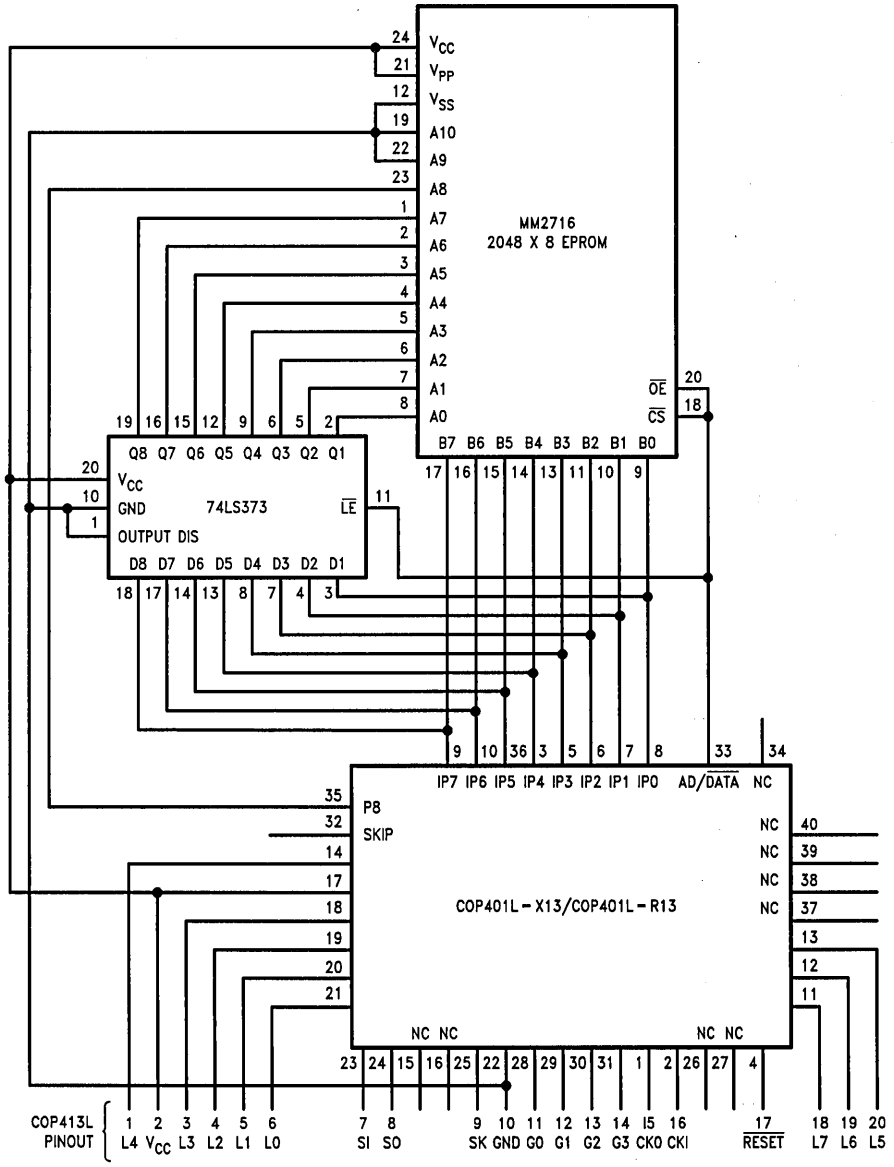
When AD/$\overline{DATA}$ turns off, the EPROM is enabled and the IP_7–IP_0 pins will input the memory data. P8 outputs the most significant address bit to the memory. (SKIP output may be used for program debug if needed.)

Twenty of the COP401L-X13/COP401L-R13 pins may be configured exactly the same as the COP413L. Selection of the COP401L-X13 or COP401L-R13 depends upon which oscillator is selected for the COP413L.

Oscillator Requirement		Order ROMless
COP413L Option 1 = 0	Ceramic Resonator or external input frequency divided by 8. CKO is oscillator out.	COP401L-X13
Option 1 = 1	Single Pin RC controlled oscillator divided by 4. CKO is no connection.	COP401L-R13

1

Typical Applications (Continued)



TL/DD/8528–13

FIGURE 9. COP401L-X13/COP401L-R13 Used to Emulate a COP413L

National Semiconductor Corporation

COP402/COP402M ROMless N-Channel Microcontrollers

General Description

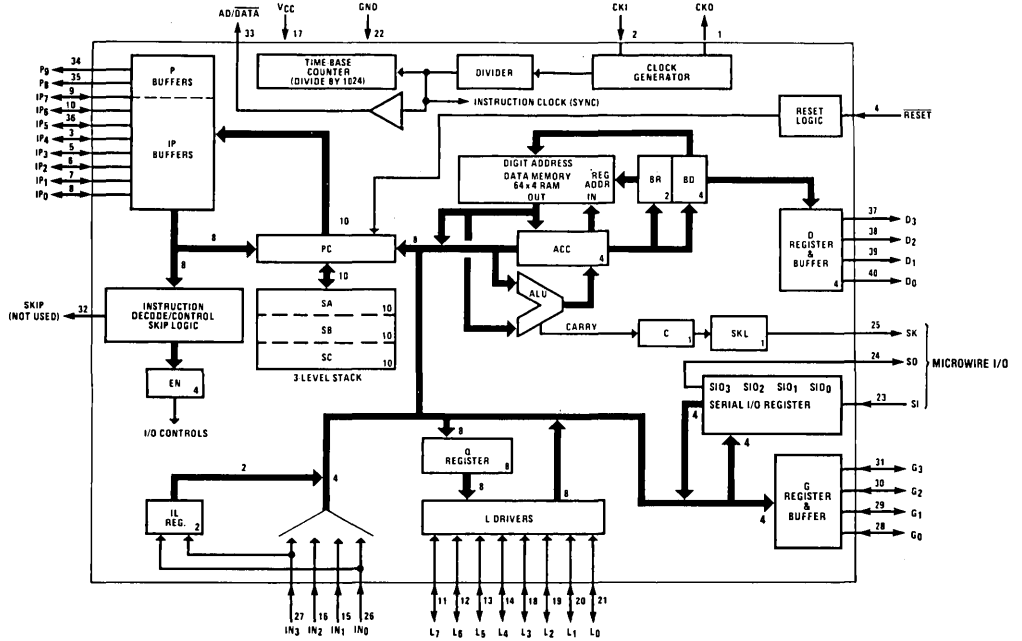
The COP402/COP402M ROMless Microcontrollers are members of the COPS™ family, fabricated using N-channel silicon gate MOS technology. Each part contains CPU, RAM, and I/O, and is identical to a COP420 device, except the ROM has been removed; pins have been added to output the ROM address and to input ROM data. In a system, the COP402 or 402M will perform exactly as the COP420; this important benefit facilitates development and debug of a COP420; this important benefit facilitates development and debug of a COP420 program prior to masking the final part. These devices are also appropriate in low volume applications, or when the program may require changing. The COP402M is identical to the COP402, except the MICROBUS™ interface option has been implemented.

The COP402 may also be used to emulate the COP410L, 411L, or 420L by appropriately reducing the clock frequency.

Features

- Extended temperature (−40°C to +85°C) COP302/COP302M, available as special order
- Low cost
- Exact circuit equivalent of COP420
- Standard 40-pin dual-in-line package
- Interfaces with standard PROM or ROM
- 64 x 4 RAM, addresses up to 1k x 8 ROM
- MICROBUS compatible (COP402M)
- Powerful instruction set
- True vectored interrupt, plus restart
- Three-level subroutine stack
- 4.0 μs instruction time
- Single supply operation (4.5V to 6.3V)
- Internal time-base counter for real-time processing
- Internal binary counter register with MICROWIRE™ serial I/O capability
- Software/hardware compatible with other members of COP400 family

Block Diagram



TL/DD/6915–1

FIGURE 1

COP402/COP402M and COP302/COP302M

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin	−0.3V to +7V
Operating Temperature Range	
COP402/COP402M	0°C to 70°C
Storage Temperature Range	−65°C to +150°C
Lead Temperature (soldering, 10 sec.)	300°C
Package Power Dissipation	750 mW at 25°C 400 mW at 70°C 250 mW at 85°C
Total Sink Current	50 mA
Total Source Current	70 mA

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

COP402/COP402M
DC Electrical Characteristics $0°C \le T_A \le 70°C$, $4.5V \le V_{CC} \le 6.3V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Operation Voltage		4.5	6.3	V
Power Supply Ripple	Peak to Peak (Note 3)		0.4	V
Supply Current	All Outputs Open V_{CC} = 5V		40	mA
Input Voltage Levels				
CKI Input Levels				
Crystal Input				
Logic High		2.4		V
Logic Low		−0.3	0.4	V
Schmitt Trigger Input				
RESET				
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.6	V
All Other Inputs				
Logic High	V_{CC} = Max	3.0		V
Logic High	V_{CC} = 5V ±5%	2.0		V
Logic Low		−0.3	0.8	V
Input Load Source Current	V_{CC} = 5V, V_{IN} = 0V	−100	−800	μA
Input Capacitance			7	pF
Hi-Z Input Leakage	V_{CC} = 5V	−1	+1	μA
Output Voltage Levels				
D, G, L, SK, SO Outputs				
TTL Operation	V_{CC} = 5V ±10%			
Logic High	I_{OH} = −100 μA	2.4		V
Logic Low	I_{OL} = 1.6 mA	−0.3	0.4	V
IP0–IP7, P8, P9, SKIP, CKO, AD/$\overline{DATA}$				
Logic High	I_{OH} = −75 μA	2.4		V
Logic Low	I_{OL} = 400 μA	−0.3	0.4	V
CMOS Operation (Note 1)				
Logic High	I_{OH} = −10 μA	V_{CC} − 1		V
Logic Low	I_{OL} = 10 μA	−0.3	0.2	V
Output Current Levels				
LED Direct Drive (COP402)	V_{CC} = 6V			
Logic High	V_{OH} = 2.0V	2.5	14	mA
TRI-STATE® (COP402M) Leakage Current	V_{CC} = 5V	−50	+50	μA
Allowable Sink Current				
Per Pin (L, D, G)			10	mA
Per Pin (All Others)			2	mA
Per Port (L)			16	mA
Per Port (D, G)			10	mA
Allowable Source Current				
Per Pin (L)			−15	mA
Per Pin (All Others)			−1.5	mA

Note 1: TRI-STATE and LED configurations are excluded.

COP402/COP402M
AC Electrical Characteristics $0°C \le T_A \le 70°C$, $4.5V \le V_{CC} \le 6.3V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time		4	10	μs
Operating CKI Frequency	÷16 Mode	1.6	4.0	MHz
CKI Duty Cycle (Note 1)		40	60	%
Rise Time	Frequency = 4 MHz		60	ns
Fall Time	Frequency = 4 MHz		40	ns
Inputs:				
SI				
t_{SETUP}		0.3		μs
t_{HOLD}		250		ns
All Other Inputs				
t_{SETUP}		1.7		μs
t_{HOLD}		300		ns
Output Propagation Delay	Test Conditions: $R_L = 5k, C_L = 50$ pF, $V_{OUT} = 1.5V$			
SO and SK				
t_{pd1}			1.0	μs
t_{pd0}			1.0	μs
CKO				
t_{pd1}			0.25	μs
t_{pd0}			0.25	μs
AD/$\overline{DATA}$, SKIP				
t_{pd1}			0.6	μs
t_{pd0}			0.6	μs
All Other Outputs				
t_{pd1}			1.4	μs
t_{pd0}			1.4	μs
MICROBUS Timing	$C_L = 100$ pF, $V_{CC} = 5V \pm 5\%$			
Read Operation *(Figure 4)*				
Chip Select Stable before $\overline{RD}$—t_{CSR}		65		ns
Chip Select Hold Time for $\overline{RD}$—t_{RCS}		20		ns
$\overline{RD}$ Pulse Width—t_{RR}		400		ns
Data Delay from $\overline{RD}$—t_{RD}			375	ns
$\overline{RD}$ to Data Floating—t_{DF}			250	ns
Write Operation *(Figure 5)*				
Chip Select Stable before $\overline{WR}$—t_{CSW}		65		ns
Chip Select Hold Time for $\overline{WR}$—t_{WCS}		20		ns
$\overline{WR}$ Pulse Width—t_{WW}		400		ns
Data Set-Up Time for $\overline{WR}$—t_{DW}		320		ns
Data Hold Time for $\overline{WR}$—t_{WD}		100		ns
INTR Transition Time from $\overline{WR}$—t_{WI}			700	ns

Note 1: Duty Cycle = $t_{WI}/(t_{W1} + t_{W0})$.

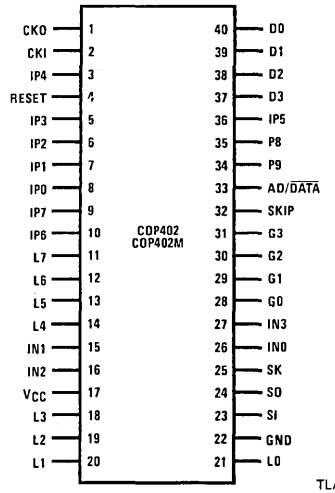
Note 2: See *Figure 9* for additional I/O characteristics.

Note 3: Voltage change must be less than 0.5V in a 1 ms period.

Note 4: Exercise great care not to exceed maximum device power dissipation limits when direct driving LEDs (or sourcing similar loads) at high temperature.

Connection Diagram

Dual-In-Line Package



TL/DD/6915-2

Top View

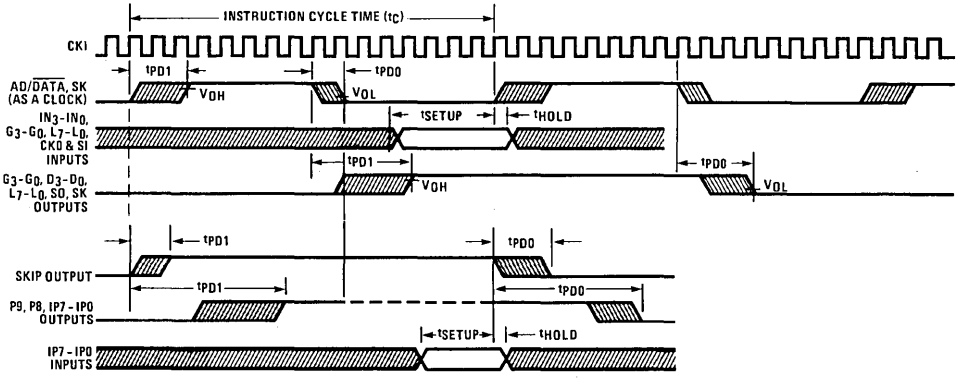
Order Number COP402N or COP402MN
See NS Package Number N40A

FIGURE 2.

Pin Descriptions

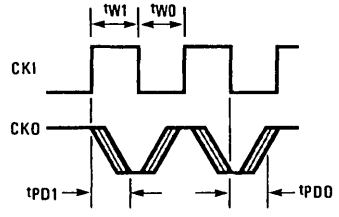
Pin	Description
L_7–L_0	8 bidirectional I/O ports with TRI-STATE
G_3–G_0	4 bidirectional I/O ports
D_3–D_0	4 general purpose outputs
IN_3–IN_0	4 general purpose inputs
SI	Serial input (or counter input)
SO	Serial output (or general purpose output)
SK	Logic-controlled clock (or general purpose output)
AD/$\overline{DATA}$	Address out/data in flag
SKIP	Instruction skip output
CKI	System oscillator input
CKO	System oscillator output
$\overline{RESET}$	System reset input
V_{CC}	Power supply
GND	Ground
IP7–IP0	8 bidirectional ROM address and data ports
P8, P9	2 most significant ROM address outputs

Timing Diagrams



TL/DD/6915-3

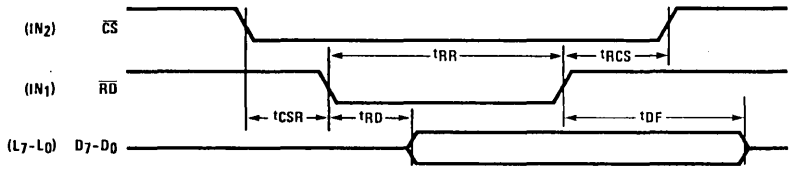
FIGURE 3a. Input/Output Timing Diagrams (Crystal ÷ 16 Mode)



TL/DD/6915-4

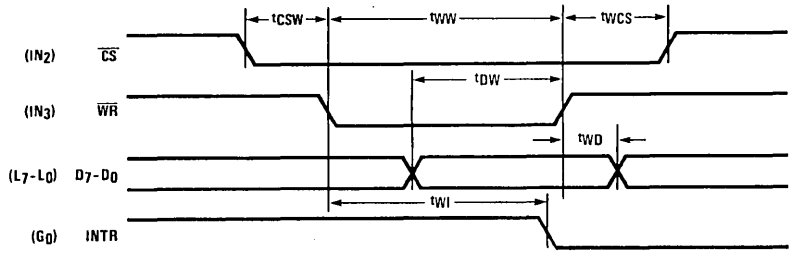
FIGURE 3b. CKO Output Timing

Timing Diagrams (Continued)



TL/DD/6915-5

FIGURE 4. MICROBUS Read Operation Timing



TL/DD/6915-6

FIGURE 5. MICROBUS Write Operation Timing

Functional Description

A block diagram of the COP402 is given in *Figure 1*. Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1" (greater than 2V). When a bit is reset, it is a logic "0" (less than 0.8V).

PROGRAM MEMORY

Program Memory consists of a 1,024-byte external memory (typically PROM). Words of this memory may be program instructions, program data or ROM addressing data. Because of the special characteristics associated with the JP, JSRP, JID and LQID instructions, ROM must often be thought of as being organized into 16 pages of 64 words each.

ROM addressing is accomplished by a 10-bit PC register. Its binay value selects one of the 1,024 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential **10-bit binary count** value. Three levels of subroutine nesting are implemented by the 10-bit subroutine save registers, SA, SB and SC, providing a last-in, first-out (LIFO) hardware subroutine stack.

ROM instruction words are fetched, decoded and executed by the Instruction Decode, Control and Skip Logic circuitry.

DATA MEMORY

Data memory consists of a 256-bit RAM, organized as 4 data registers of 16 4-bit digits. RAM addressing is implemented by a 6-bit B register whose upper 2 bits (Br) select 1 of 4 data registers and lower 4 bits (Bd) select 1 of 16 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) is usually loaded into or from, or exchanged with, the A register (accumulator), it may also be loaded into or from the Q latches or loaded from the L ports. RAM addressing may also be performed directly by the LDD and XAD instruction based upon the 6-bit contents of the operand field of these instructions. The Bd register also serves as a source register for 4-bit data sent directly to the D outputs.

INTERNAL LOGIC

The 4-bit **A register** (accumulator) is the source and destination register for most I/O, arithmetic, logic and data memory access operations. It can also be used to load the Br and Bd portions of the B register, to load and input 4 bits of the 8-bit Q latch data, to input 4 bits of the 8-bit L I/O port data and to perform data exchanges with the SIO register.

A **4-bit adder** performs the arithmetic and logic functions of the COP402/402M, storing its results in A. It also outputs a carry bit to the 1-bit **C register**, most often employed to indicate arithmetic overflow. The C register, in conjunction with the XAS instruction and the EN register, also serves to control the SK output. C can be outputted directly to SK or can enable SK to be a sync clock each instruction cycle time. (See XAS instruction and EN register description, below.)

Four **general-purpose inputs, IN_3–IN_0**, are provided; IN_1, IN_2, and IN_3 may be selected, by a mask-programmable option, as Read Strobe, Chip Select and Write Strobe inputs, respectively, for use in MICROBUS applications.

The **D register** provides 4 general-purpose outputs and is used as the destination register for the 4-bit contents of Bd.

The **G register** contents are outputs to 4 general-purpose bidirectional I/O ports. G_0 may be mask-programmed as a "ready" output for MICROBUS applications.

The **Q register** is an internal, latched, 8-bit register, used to hold data loaded to or from M and A, as well as 8-bit data from ROM. Its contents are output to the L I/O ports when the L drivers are enabled under program control. (See LEI instruction.) With the MICROBUS option selected, Q can also be loaded with the 8-bit contents of the L I/O ports upon the occurrence of a write strobe from the host CPU.

Functional Description (Continued)

The **8 L drivers**, when enabled, output the contents of latched Q data to the L I/O ports. Also, the contents of L may be read directly into A and M. As explained above, the MICROBUS option allows L I/O port data to be latched into the Q register. L I/O ports can be directly connected to the segments of a multiplexed LED display (using the LED Direct Drive output configuration option) with Q data being outputted to the Sa–Sg and decimal point segments of the display.

The **SIO register** functions as a 4-bit serial-in/serial-out shift register or as a binary counter depending on the contents of the EN register. (See EN register description below.) Its contents can be exchanged with A, allowing it to input or output a continuous serial data stream. SIO may also be used to provide additional parallel I/O by connecting SO to external serial-in/parallel-out shift registers.

The **XAS instruction** copies C into the SKL latch. In the counter mode, SK is the output of SKL. In the shift register mode, SK outputs SKL ANDed with internal instruction cycle clock.

The **EN register** is an internal 4-bit register loaded under program control by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register (EN_3–EN_0).

1. The least significant bit of the enable register, EN_0, selects the SIO register as either a 4-bit shift register or a 4-bit binary counter. With EN_0 set, SIO is an asynchronous binary counter, *decrementing* its value by one upon each low-going pulse ("1" to "0") occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output is equal to the value of EN_3. With EN_0 reset, SIO is a serial shift register shifting left each instruction cycle time. The data present at SI goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. (See 4 below.) The SK output becomes a logic-controlled clock.
2. With EN_1 set the IN_1 input is enabled as an interrupt input. Immediately following an interrupt, EN_1 is reset to disable further interrupts.
3. With EN_2 set, the L drivers are enabled to output the data in Q to the L I/O ports. Resetting EN_2 disables the L drivers, placing the L I/O ports in a high-impedance input state. If the MICROBUS option is being used, EN_2 does not affect the L drivers.
4. EN_3, in conjunction with EN_0, affects the SO output. With EN_0 set (binary counter option selected) SO will output the value loaded into EN_3. With EN_0 reset (serial shift register option selected), setting EN_3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN_3 with the serial shift register option selected disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains reset to "0." The table below provides a summary of the modes associated with EN_3 and EN_0.

INTERRUPT

The following features are associated with the IN_1 interrupt procedure and protocol and must be considered by the programmer when utilizing interrupts.

a. The interrupt, once acknowledged as explained below, pushes the next sequential program counter address (PC + 1) onto the stack, pushing in turn the contents of the other subroutine-save registers to the next lower level (PC + 1 → SA → SB → SC). Any previous contents of SC are lost. The program counter is set to hex address 0FF (the last word of page 3) and EN_1 is reset.

b. An interrupt will be acknowledged only after the following conditions are met:
   1. EN_1 has been set.
   2. A low-going pulse ("1" to "0") at least two instruction cycles wide occurs on the IN_1 input.
   3. A currently executing instruction has been completed.
   4. All successive transfer of control instructions and successive LBIs have been completed (e.g., if the main program is executing a JP instruction which transfers program control to another JP instruction, the interrupt will not be acknowledged until the second JP instruction has been executed.

c. Upon acknowledgement of an interrupt, the skip logic status is saved and later restored upon the popping of the stack. For example, if an interrupt occurs during the execution of ASC (Add with Carry, Skip on Carry) instruction which results in carry, the skip logic status is saved and program control is transferred to the interrupt servicing routine at hex address 0FF. At the *end* of the interrupt routine, a RET instruction is executed to "pop" the stack and return program control to the instruction following the original ASC. At *this time*, the skip logic is enabled and skips this instruction because of the previous ASC carry. Subroutines and the LQID instruction should not be nested within the interrupt servicing routine since their popping of the stack enables any previously saved main program skips, interfering with the orderly execution of the interrupt routine.

d. The first instruction of the interrupt routine at hex address 0FF must be a NOP.

e. An LEI instruction can be put immediately before the RET to re-enable interrupts.

TABLE I. Enable Register Modes—Bits EN_3 and EN_0

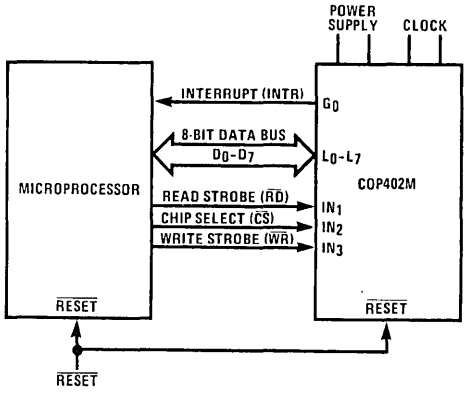
EN_3	EN_0	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL = 1, SK = SYNC If SKL = 0, SK = 0
1	0	Shift Register	Input to Shift Register	Serial Out	If SKL = 1, SK = SYNC If SKL = 0, SK = 0
0	1	Binary Counter	Input to Binary Counter	0	If SKL = 1, SK = 1 If SKL = 0, SK = 0
1	1	Binary Counter	Input to Binary Counter	1	If SKL = 1, SK = 1 If SKL = 0, SK = 0

Functional Description (Continued)

MICROBUS INTERFACE

The COP402M can be used as a peripheral microprocessor device, inputting and outputting data from and to a host microprocessor (μP). IN_1, IN_2, and IN_3 general purpose inputs become MICROBUS compatible read-strobe, chip-select, and write-strobe lines, respectively. IN_1 becomes $\overline{RD}$—a logic "0" on this input will cause Q latch data to be enabled to the L ports for input to the μP. IN_2 becomes $\overline{CS}$—a logic "0" on this line selects the COP402M as the μP peripheral device by enabling the operation of the $\overline{RD}$ and $\overline{WR}$ lines and allows for the selection of one of several peripheral components. IN_3 becomes $\overline{WR}$—a logic "0" on this line will write bus data from the L ports to the Q latches for input to the COP402M. G_0 becomes INTR, a "ready" output reset by a write pulse from the μP on the $\overline{WR}$ line, providing the "handshaking" capability necessary for asynchronous data transfer between the host CPU and the COP402M.

This option has been designed for compatibility with National's MICROBUS—a standard interconnect system for 8-bit parallel data transfer between MOS/LSI CPUs and interfacing devices. (See MICROBUS, National Publication.) The functioning and timing relationships between the COP402M signal lines affected by this option are as specified for the MICROBUS interface, and are given in the AC electrical characteristics and shown in the timing diagrams (*Figures 4* and *5*). Connection to the MICROBUS is shown in *Figure 6*.



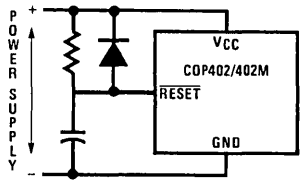
TL/DD/6915-7

FIGURE 6. MICROBUS Option Interconnect

INITIALIZATION

The Reset Logic will initialize (clear) the device upon power-up if the power supply rise time is less than 1 ms and greater than 1 μs. If the power supply rise time is greater than 1 ms, the user must provide an external RC network and diode to the $\overline{RESET}$ pin as shown below. The $\overline{RESET}$ pin is configured as a Schmitt trigger input. If not used it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the $\overline{RESET}$ input, provided it stays low for at least two instruction cycle times.

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN, G, and SO are cleared. The SK output is enabled as a SYNC output, providing a pulse each instruction cycle time. *Data Memory (RAM) is not cleared upon initialization.* The first instruction at address 0 must be a CLRA.



$RC \geq 5 \times$ Power Supply Rise Time TL/DD/6915-8

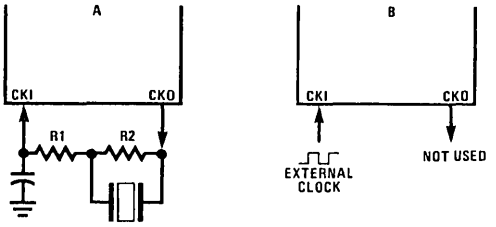
FIGURE 7. Power-Up Clear Circuit

OSCILLATOR

There are two basic clock oscillator configurations available as shown by *Figure 8*.

a. Crystal Controlled Oscillator. CKI and CKO are connected to an external crystal. The instruction cycle time equals the crystal frequency divided by 16.

b. External Oscillator. CKI is driven by an external clock signal. The instruction cycle time is the clock frequency divided by 16.



TL/DD/6915-9

Crystal Value	Component Values		
	R1	R2	C
4 MHz	1k	1M	27 pF
3.58 MHz	1k	1M	27 pF
2.09 MHz	1k	1M	56 pF

FIGURE 8. COP402/402M Oscillator

EXTERNAL MEMORY INTERFACE

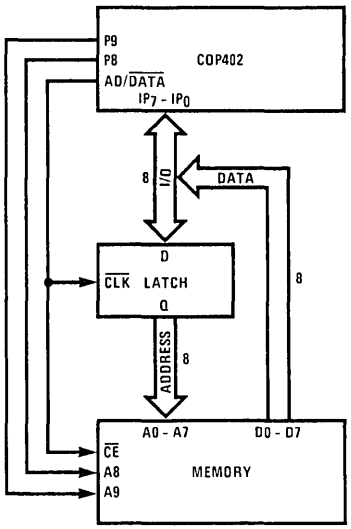
The COP402 and COP402M are designed for use with an external Program Memory. This memory may be implemented using any devices having the following characteristics:

1. random addressing
2. TTL-compatible TRI-STATE outputs
3. TTL = compatible inputs
4. access time = 1.0 μs, max.

Typically these requirements are met using bipolar or MOS PROMs.

Functional Description (Continued)

During operation, the address of the next instruction is sent out on P9, P8, and IP7 through IP0 during the time that AD/$\overline{\text{DATA}}$ is high (logic "1" = address mode). Address data on the IP lines is stored into an external latch on the high-to-low transition of the AD/$\overline{\text{DATA}}$ line; P9 and P8 are dedicated address outputs, and do not need to be latched. When AD/$\overline{\text{DATA}}$ is low (logic "0" = data mode), the output of the memory is gated onto IP7 through IP0, forming the input bus. Note that the AD/$\overline{\text{DATA}}$ output has a period of one instruction time, a duty cycle of approximately 50%, and specifies whether the IP lines are used for address output or instruction input. A simplified block diagram of the external memory interface is shown in *Figure 9*.



TL/DD/6915-10

FIGURE 9. External Memory Interface to COP402

INPUT/OUTPUT

COP402 outputs have the following configurations, illustrated in *Figure 10*.

a. Standard—an enhancement-mode device to ground in conjunction with a depletion-mode device to V_{CC}, compatible with TTL and CMOS input requirements.

b. High Drive—same as **a.** except greater current sourcing capability.

c. Push-Pull—an enhancement-mode device to ground in conjunction with a depletion-mode device paralleled by an enhancement-mode device to V_{CC}. This configuration has been provided to allow for fast rise and fall times when driving capacitive loads.

d. LED Direct Drive—an enhancement-mode device to ground and to V_{CC}, meeting the typical current sourcing requirements of the segments of an LED display. The sourcing device is clamped to limit current flow. These devices may be turned off under program control (see Functional Description, EN Register), placing the outputs in a high-impedance state to provide required LED segment blanking for a multiplexed display.

e. TRI-STATE Push-Pull—an enhancement-mode device to ground and V_{CC} intended to meet the requirements associated with the MICROBUS option. These outputs are TRI-STATE outputs, allowing for connection of these outputs to a data bus shared by other bus drivers.

f. Inputs have an on-chip depletion load device to V_{CC}, as shown in *Figure 10f*.

The above input and output configurations share common enhancement-mode and depletion-mode devices. Specifically, all configurations use one or more of six devices (numbered 1–6, respectively). Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figure 10* for each of these devices.

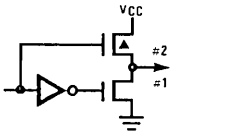
The SO, SK outputs are configured as shown in *Figure 10c*. The D and G outputs are configured as shown in *Figure 10a*.

Functional Description (Continued)

Note that when inputting data to the G ports, the G outputs should be set to "1". The L outputs are configured as in *Figure 10d* on the COP402. On the COP402M the L outputs are as in *Figure 10e*.

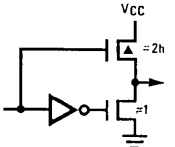
An important point to remember if using configuration d with the L drivers is that even when the L drivers are disabled, the depletion load device will source a small amount of current. (See *Figure 11*.)

IP7 through IP0 outputs are configured as shown in *Figure 10c*; P9, P8, SKIP, and AD/$\overline{\text{DATA}}$ are configured as shown in *Figure 10b*.



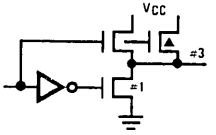
TL/DD/6915-11

a. Standard



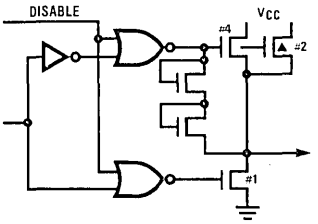
TL/DD/6915-12

b. High Drive



TL/DD/6915-13

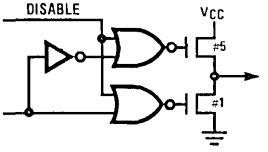
c. Push-Pull



TL/DD/6915-14

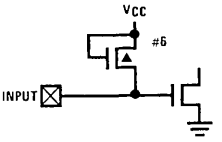
d. LED

(▲ is Depletion Device)



TL/DD/6915-15

e. TRI-STATE Push-Pull



TL/DD/6915-16

f. Input with Load

FIGURE 10. Input/Output Configurations

Typical Performance Characteristics

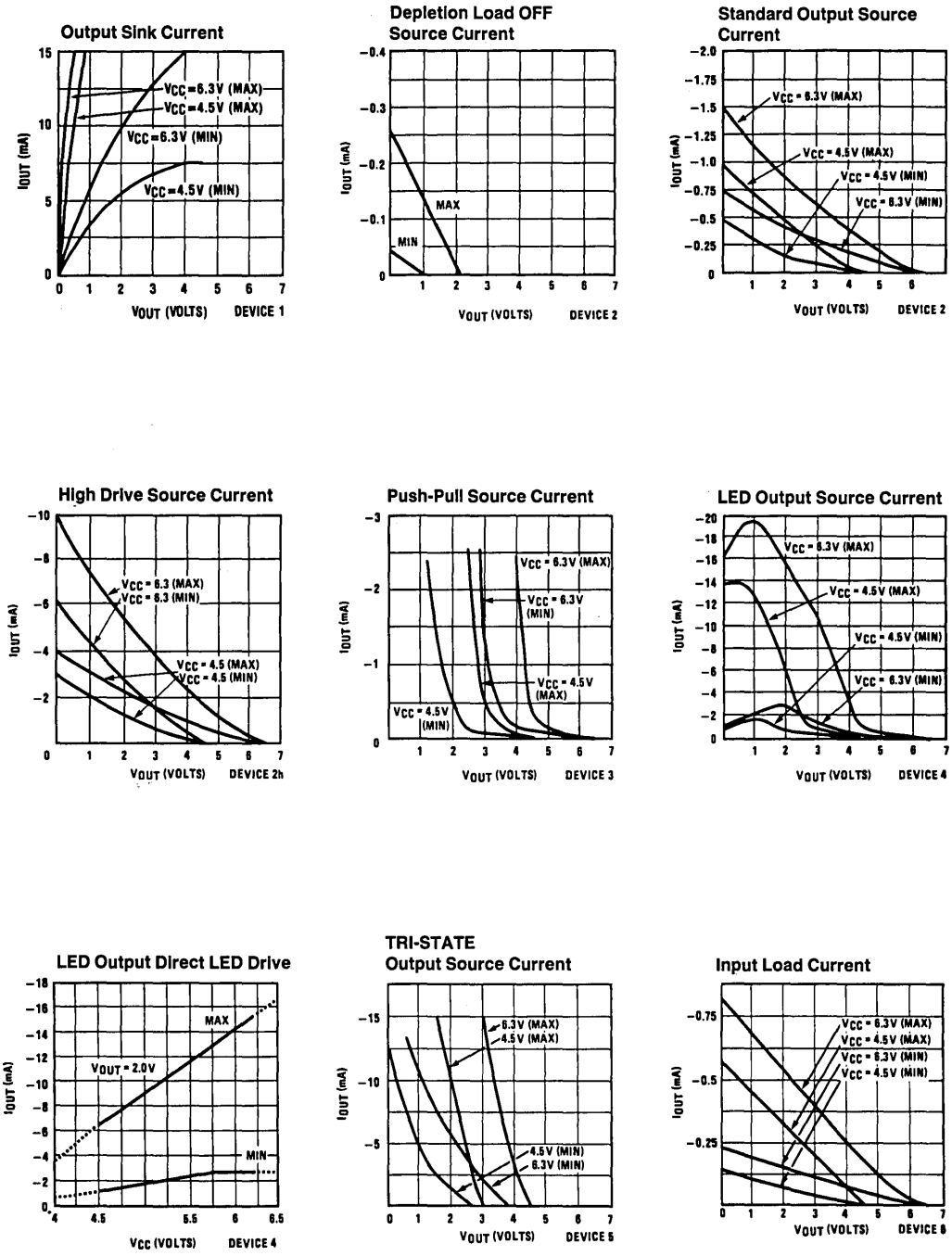
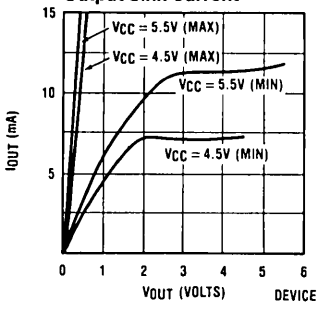


FIGURE 11. COP402/COP402M Input/Output Characteristics

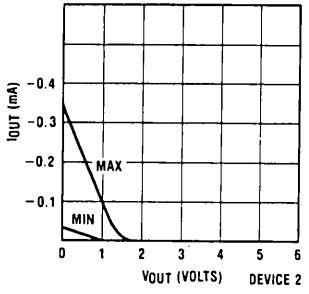
TL/DD/6915–17

Typical Performance Characteristics (Continued)

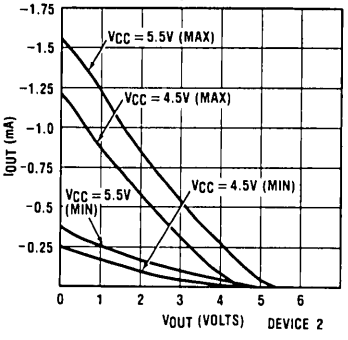
Output Sink Current



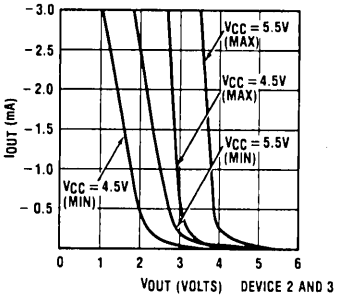
L Output Depletion Load Off Source Current



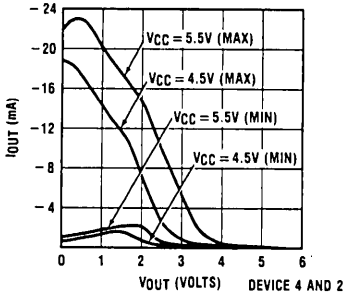
Standard Output Source Current



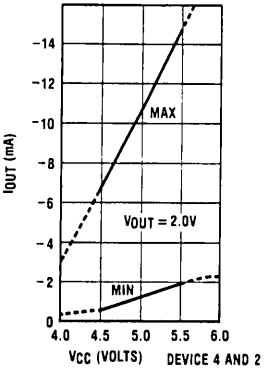
Push Pull Source Current



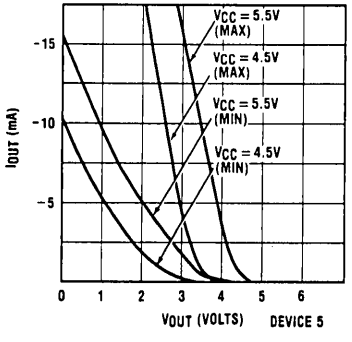
LED Output Source Current



LED Output Device LED Drive



TRI-STATE Output Source Current



Input Load Source Current

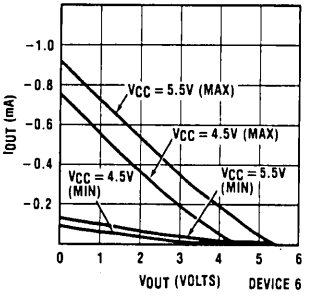


FIGURE 11a. COP302/COP302M Input/Output Characteristics

TL/DD/6915-18

Instruction Set

Table II is a symbol table providing internal architecture, instruction operand and operational symbols used in the instruction set table.

Table III provides the mnemonic, operand, machine code, data flow, skip conditions and description associated with each instruction in the COP402/402M instruction set.

TABLE II. COP402/COP402M Instruction Set Table Symbols

Symbol	Definition
INTERNAL ARCHITECTURE SYMBOLS	
A	4-bit Accumulator
B	6-bit RAM Address Register
Br	Upper 2 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry Register
D	4-bit Data Output Port
EN	4-bit Enable Register
G	4-bit Register to latch data for G I/O Port
IL	Two 1-bit Latches Associated with the IN_3 or IN_0 inputs
IN	4-bit Input port
L	8-bit TRI-STATE I/O Port
M	4-bit contents of RAM Memory pointed to by B Register
P	2-bit ROM Address Port
PC	10-bit ROM Address Register (program counter)
Q	8-bit Register to latch data for L I/O Port
SA	10-bit Subroutine Save Register A
SB	10-bit Subroutine Save Register B
SC	10-bit Subroutine Save Register C
SIO	4-bit Shift Register and Counter
SK	Logic-Controlled Clock Output

Symbol	Definition
INSTRUCTION OPERAND SYMBOLS	
d	4-bit Operand Field, 0–15 binary (RAM Digit Select)
r	2-bit Operand Field, 0–3 binary (RAM Register Select)
a	9-bit Operand Field, 0–511 binary (ROM Address)
y	4-bit Operand Field, 0–15 binary (Immediate Data)
RAM(s)	Contents of RAM location addressed by s
ROM(t)	Contents of ROM location addressed by t
OPERATIONAL SYMBOLS	
+	Plus
−	Minus
→	Replaces
↔	Is exchanged with
=	Is equal to
$\overline{A}$	The one's complement of A
⊕	Exclusive-OR
:	Range of values

TABLE III. COP402/COP402M Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	|0011|0000|	A + C + RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	|0011|0001|	A + RAM(B) → A	None	Add RAM to A
ADT		4A	|0100|1010|	A + 10_{10} → A	None	Add Ten to A
AISC	y	5–	|0101| y |	A + y → A	Carry	Add Immediate, Skip on Carry (y ≠ 0)
CASC		10	|0001|0000|	$\overline{A}$ + RAM(B) + C → A Carry → C	Carry	Complement and Add with Carry, Skip on Carry
CLRA		00	|0000|0000|	0 → A	None	Clear A
COMP		40	|0100|0000|	$\overline{A}$ → A	None	One's complement of A to A
NOP		44	|0100|0100|	None	None	No Operation
RC		32	|0011|0010|	"0" → C	None	Reset C
SC		22	|0010|0010|	"1" → C	None	Set C
XOR		02	|0000|0010|	A ⊕ RAM(B) → A	None	Exclusive-OR RAM with A

Instruction Set (Continued)

TABLE III. COP402/COP402M Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
TRANSFER OF CONTROL INSTRUCTIONS						
JID		FF	1111 | 1111	ROM ($PC_{9:8}$, A,M) → $PC_{7:0}$	None	Jump Indirect (Note 3)
JMP	a	6–	0110 | 00 | $a_{9:8}$	a → PC	None	Jump
		--	$a_{7:0}$			
JP	a	--	1 | $a_{6:0}$ (pages 2,3 only)	a → $PC_{6:0}$	None	Jump within Page (Note 4)
			or			
		--	11 | $a_{5:0}$ (all other pages)	a → $PC_{5:0}$		
JSRP	a	--	10 | $a_{5:0}$	PC + 1 → SA → SB → SC 0010 → $PC_{9:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 5)
JSR	a	6–	0110 | 10 | $a_{9:8}$	PC + 1 → SA → SB → SC a → PC	None	Jump to Subroutine
		--	$a_{7:0}$			
RET		48	0100 | 1000	SC → SB → SA → PC	None	Return from Subroutine
RETSK		49	0100 | 1001	SC → SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
MEMORY REFERENCE INSTRUCTIONS						
CAMQ		33	0011 | 0011	A → $Q_{7:4}$	None	Copy A, RAM to Q
		3C	0011 | 1100	RAM(B) → $Q_{3:0}$		
CQMA		33	0011 | 0011	$Q_{7:4}$ → RAM(B)	None	Copy Q to RAM, A
		2C	0010 | 1100	$Q_{3:0}$ → A		
LD	r	-5	00 | r | 0101	RAM(B) → A Br ⊕ r → Br	None	Load RAM into A, Exclusive-OR Br with r
LDD	r,d	23	0010 | 0011	RAM(r,d) → A	None	Load A with RAM pointed to directly by r,d
		--	00 | r | d			
LQID		BF	1011 | 1111	ROM($PC_{9:8}$,A,M) → Q SB → SC	None	Load Q Indirect (Note 3)
RMB	0	4C	0100 | 1100	0 → $RAM(B)_0$	None	Reset RAM Bit
	1	45	0100 | 0101	0 → $RAM(B)_1$		
	2	42	0100 | 0010	0 → $RAM(B)_2$		
	3	43	0100 | 0011	0 → $RAM(B)_3$		
SMB	0	4D	0100 | 1101	1 → $RAM(B)_0$	None	Set RAM Bit
	1	47	0100 | 0111	1 → $RAM(B)_1$		
	2	46	0100 | 0110	1 → $RAM(B)_2$		
	3	4B	0100 | 1011	1 → $RAM(B)_3$		
STII	y	7–	0111 | y	y → RAM(B) Bd + 1 → Bd	None	Store Memory Immediate and Increment Bd
X	r	-6	00 | r | 0110	RAM(B) ⟷ A Br ⊕ r → Br	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	r,d	23	0010 | 0011	RAM(r,d) ⟷ A	None	Exchange A with RAM pointed to directly by r,d
		--	10 | r | d			

Instruction Set (Continued)

TABLE III. COP402/COP402M Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
MEMORY REFERENCE INSTRUCTIONS (Continued)						
XDS	r	–7	|00|r|0111|	RAM(B) ⟷ A Bd – 1 → Bd Br ⊕ r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd, Exclusive-OR Br with r
XIS	r	–4	|00|r|0100|	RAM(B) ⟷ A Bd + 1 → Bd Br ⊕ r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd, Exclusive-OR Br with r
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	|0101|0000|	A → Bd	None	Copy A to Bd
CBA		4E	|0100|1110|	Bd → A	None	Copy Bd to A
LBI	r,d	– –	|00|r|(d – 1)| (d=0, 9:15) or 33 |0011|0011| – – |10|r|d| (any d)	r,d → B	Skip until not a LBI	Load B Immediate with r,d (Note 6)
LEI	y	33 6–	|0011|0011| |0110|y|	y → EN	None	Load EN Immediate (Note 7)
XABR		12	|0001|0010|	A ⟷ Br (0,0 → A_3,A_2)	None	Exchange A with Br
TEST INSTRUCTIONS						
SKC		20	|0010|0000|		C = "1"	Skip if C is True
SKE		21	|0010|0001|		A = RAM(B)	Skip if A Equals RAM
SKGZ		33 21	|0011|0011| |0010|0001|		$G_{3:0}$ = 0	Skip if G is Zero (all 4 bits)
SKGBZ		33	|0011|0011|	1st byte		Skip if G Bit is Zero
	0	01	|0000|0001|	2nd byte	G_0 = 0	
	1	11	|0001|0001|		G_1 = 0	
	2	03	|0000|0011|		G_2 = 0	
	3	13	|0001|0011|		G_3 = 0	
SKMBZ	0	01	|0000|0001|		$RAM(B)_0$ = 0	Skip if RAM Bit is Zero
	1	11	|0001|0001|		$RAM(B)_1$ = 0	
	2	03	|0000|0011|		$RAM(B)_2$ = 0	
	3	13	|0001|0011|		$RAM(B)_3$ = 0	
SKT		41	|0100|0001|		A time-base counter carry has occurred since last test	Skip on Timer (Note 3)

Instruction Set (Continued)

TABLE III. COP402/COP402M Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
INPUT/OUTPUT INSTRUCTIONS						
ING		33 2A	0011 | 0011 0010 | 1010	G → A	None	Input G Ports to A
ININ		33 28	0011 | 0011 0010 | 1000	IN → A	None	Input IN Inputs to A (Notes 2 and 8)
INIL		33 29	0011 | 0011 0010 | 1001	IL_3, "0", IL_0 → A	None	Input IL Latches to A (Note 3)
INL		33 2E	0011 | 0011 0010 | 1110	$L_{7:4}$ → RAM(B) $L_{3:0}$ → A	None	Input L Ports to RAM,A
OBD		33 3E	0011 | 0011 0011 | 1110	Bd → D	None	Output Bd to D Outputs
OGI	y	33 5–	0011 | 0011 0101 | y	y → G	None	Output to G Ports Immediate
OMG		33 3A	0011 | 0011 0011 | 1010	RAM(B) → G	None	Output RAM to G Ports
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO (Note 3)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicitly defined (e.g., Br and Bd are explicitly defined). Bits are numbered 0 to N where 0 signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit register.

Note 2: The ININ instruction is not available on the 24-pin COP421 since this device does not contain the IN inputs.

Note 3: For additional information on the operation of the XAS, JID, LQID, INIL, and SKT instructions, see below.

Note 4: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 5: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 6: LBI is a single-byte instruction if d = 0, 9, 10, 11, 12, 13, 14, or 15. The machine code for the lower 4 bits equals the binary value of the "d" data *minus 1*, e.g., to load the lower four bits of B (Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Note 7: Machine code for operand field y for LEI instruction should equal the binary value to be latched into EN, where a "1" or "0" in each bit of EN corresponds with the selection or deselection of a particular function associated with each bit. (See Functional Description, EN Register.)

Note 8: The COP402M will always read a "1" into A1 with the ININ instruction.

Description of Selected Instructions

The following information is provided to assist the user in understanding the operation of several unique instructions and to provide notes useful to programmers in writing programs.

XAS INSTRUCTION

XAS (Exchange A with SIO) exchanges the 4-bit contents of the accumulator with the 4-bit contents of the SIO register.

The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. An XAS instruction will also affect the SK output. (See Functional Description, EN Register, above.) If SIO is selected as a shift register, an XAS instruction must be performed once evey 4 instruction cycles to effect a continuous data stream.

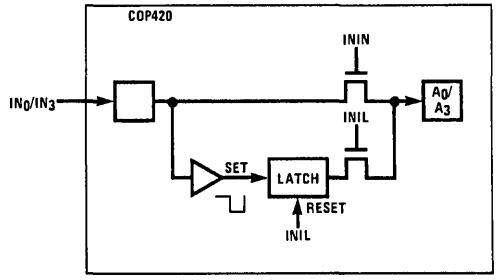
JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower 8 bits of the ROM address register PC with the *contents* of ROM addressed by the 10-bit word, $PC_{9:8}$, A, M. PC_9 and PC_8 are not affected by this instruction.

Note that JID requires 2 instruction cycles.

INIL INSTRUCTION

INIL (Input IL Latches to A) inputs 2 latches, IL_3 and IL_0 (see *Figure 12*) and CKO into A. The IL_3 and IL_0 latches are set if a low-going pulse ("1" to "0") has occurred on the IN_3 and IN_0 inputs since the last INIL instruction, provided the input pulse stays low for at least two instruction times. Execution of an INIL inputs IL_3 and IN_0 into A3 and A0 respectively, and resets these latches to allow them to respond to subsequent low-going pulses on the IN_3 and IN_0 lines. If CKO is mask programmed as a general purpose input, an INIL will input the state of CKO into A2. If CKO has not been so programmed, a "1" will be placed in A2. A "0" is always placed in A1 upon the execution of an INIL. The general purpose inputs IN_3–IN_0 are input to A upon the execution of an ININ instruction. (See Table III, ININ instruction.) INIL is useful in recognizing pulses of short duration or pulses which occur too often to be read conveniently by an ININ instruction.



TL/DD/6915–19

FIGURE 12. IN_0/IN_3 Latches

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 10-bit word PC_9, PC_8, A, M. LQID can be used for table lookup or code conversion such as BCD to seven-segment. The LQID instruction "pushes" the stack (PC + 1 → SA → SB → SC) and replaces the least significant 8 bits of PC as follows: A → $PC_{7:4}$, RAM(B) → $PC_{3:0}$, leaving PC_9 and PC_8 unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SC → SB → SA → PC), restoring the saved value of PC to continue sequential program execution. Since LQID pushes SB → SC, the previous contents of SC are lost. Also, when LQID pops the stack, the previously pushed contents of SB are left in SC. The net result is that the contents of SB are placed in SC (SB → SC). Note that LQID takes two instruction cycle times to execute.

SKT INSTRUCTION

The SKT (Skip on Timer) instruction tests the state of an internal 10-bit time-base counter. This counter divides the instruction cycle clock frequency by 1024 and provides a latched indication of counter overflow. The SKT instruction tests this latch, executing the next program instruction if the latch is not set. If the latch has been set since the previous test, the next program instruction is skipped and the latch is reset. The features associated with this instruction, therefore, allow the controller to generate its own time-base for real-time processing rather than relying on an external input signal.

For example, using a 2.097 MHz crystal as the time-base to the clock generator, the instruction cycle clock frequency will be 131 kHz (crystal frequency ÷ 16) and the binary counter output pulse frequency will be 128 Hz. For time-of-day or similar real-time processing, the SKT instruction can call a routine which increments a "seconds" counter every 128 ticks.

INSTRUCTION SET NOTES

a. The first word of a program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, one instruction cycle time is devoted to skipping each byte of the skipped instruction. Thus all program paths take the same number of cycle times whether instructions are skipped or executed, except JID and LQID. LQID and JID take two cycle times if executed and one if skipped.

c. The ROM is organized into 16 pages of 64 words each. The Program Counter is a 10-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID or LQID instruction is located in the last word of a page, the instruction operates as if it were in the next page. For example: a JP located in the last word of a page will jump to a location in the next page. Also, a LQID or JID located in the last word of page 3, 7, 11, or 15 will access data in the next group of 4 pages.

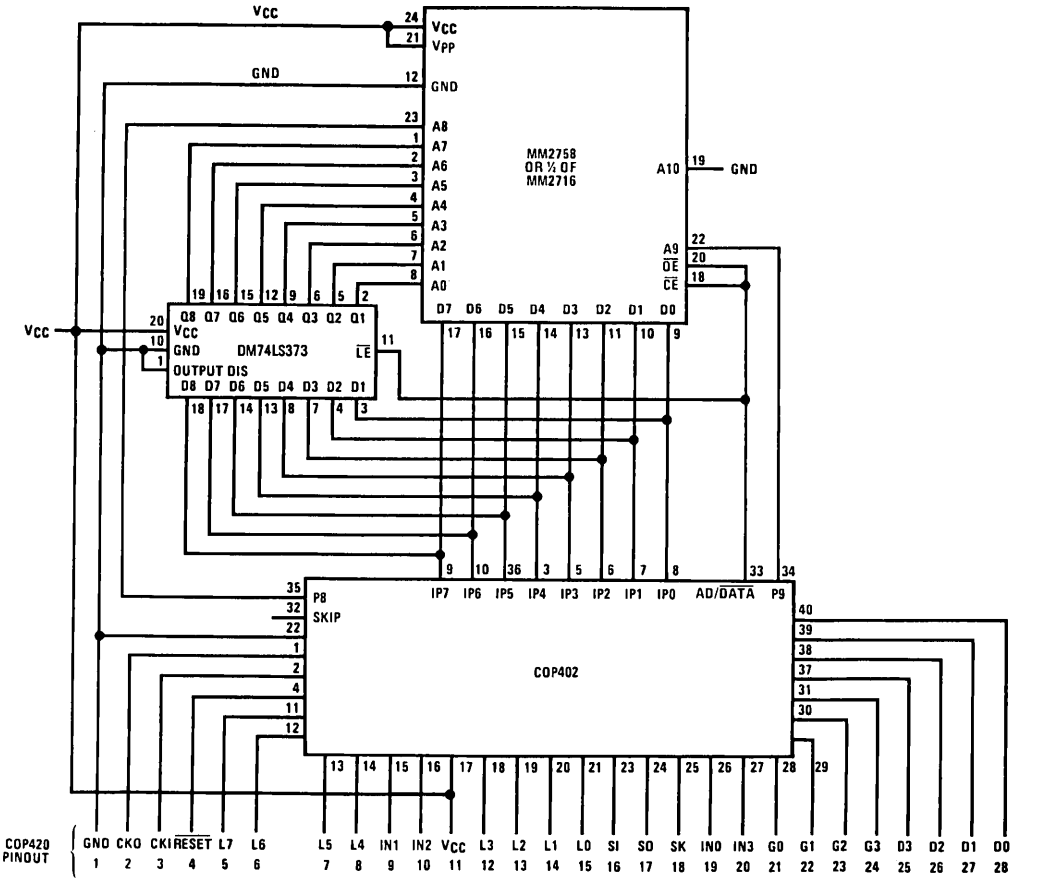
Typical Application: PROM-Based System

The COP402 may be used to exactly emulate the COP420, *Figure 13* shows the interconnect to implement a COP420 hardware emulation. This connection uses two MM5204 EPROMs as external memory. Other memory can be used such as bipolar PROM or RAM.

Pins IP7–IP0 are bidirectional inputs and outputs. When the AD/$\overline{\text{DATA}}$ clocking output turns on, the EPROM drivers are disabled and IP7–IP0 output addresses. The 8-bit latch (MM74C373) latches the addresses to drive the memory.

When AD/$\overline{\text{DATA}}$ turns off, the EPROMs are enabled and the IP7-IP0 pins will input the memory data. P8 and P9 output the most significant address bits to the memory. (SKIP output may be used for program debug if needed.)

The other 28 pins of the COP402 may be configured exactly the same as a COP420. The COP402M chip can be used if the MICROBUS feature of the COP420 is needed.



TL/DD/6915–20

FIGURE 13. COP402 Used to Emulate a COP420

Option List

COP402 MASK OPTIONS

The following COP420 options have been implemented in this basic version of the COP402. Subsequent versions of the COP402 will implement different combinations of available options; such versions will be identified as COP402-A, COP402-B, etc.

Option Value	Comment
Option 1 = 0	Ground Pin—no option available
Option 2 = 0	CKO is clock generator output to crystal
Option 3 = 0	CKI is crystal input ÷16 (may be overridden externally)
Option 4 = 0	RESET pin has load device to V_{CC}
Option 5 = 2 (402) = 3 (402M)	L7 has LED direct-drive output L7 has TRI-STATE push-pull output
Option 6 = 2, 3	L6 same as L7
Option 7 = 2, 3	L5 same as L7
Option 8 = 2, 3	L4 same as L7
Option 9 = 0 (402) = 1 (402M)	IN1 has load device to V_{CC} Hi Z
Option 10 = 0 (402) = 1 (402M)	IN2 has load device to V_{CC} Hi Z
Option 11 = 0	V_{CC} pin—no option available
Option 12 = 2, 3	L3 same as L7
Option 13 = 2, 3	L2 same as L7
Option 14 = 2, 3	L1 same as L7
Option 15 = 2, 3	L0 same as L7
Option 16 = 0	SI has load device to V_{CC}
Option 17 = 2	SO has push-pull output
Option 18 = 2	SK has push-pull output
Option 19 = 0	IN0 has load device to V_{CC}
Option 20 = 0 (402) = 1 (402M)	IN3 has load device to V_{CC} Hi Z
Option 21 = 0	G0 has standard output
Option 22 = 0	G1 same as G0
Option 23 = 0	G2 same as G0
Option 24 = 0	G3 same as G0
Option 25 = 0	D3 has standard output
Option 26 = 0	D2 same as D3
Option 27 = 0	D1 same as D3
Option 28 = 0	D0 same as D3
Option 29 = 0 (402) = 1 (402M)	normal operation MICROBUS operation
Option 30 = N/A	40-pin package

National Semiconductor Corporation

COP404 ROMless N-Channel Microcontroller

General Description

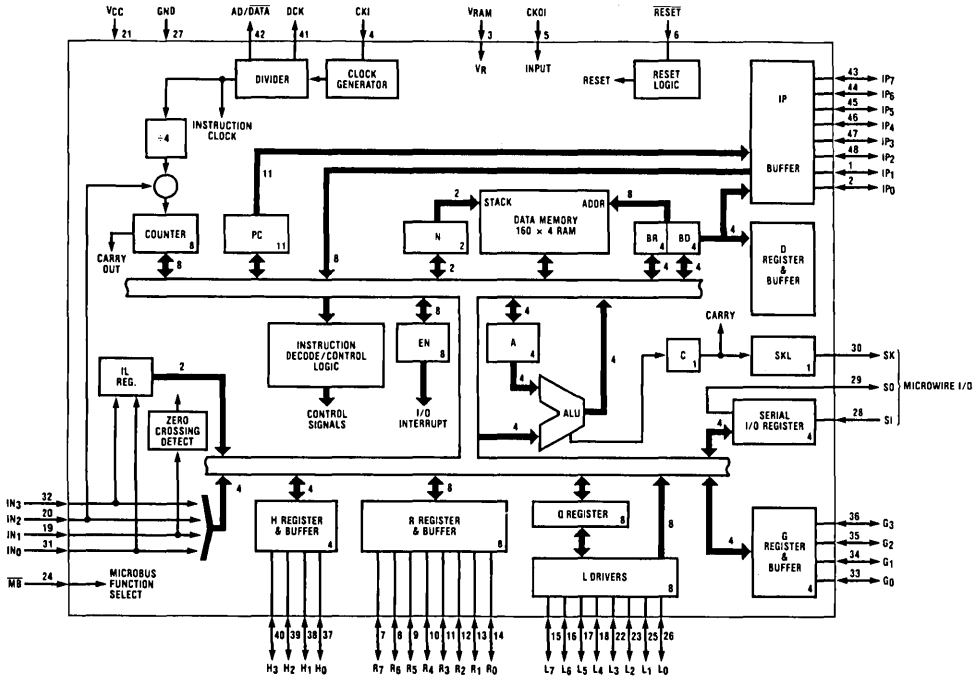
The COP404 ROMless N-Channel Microcontrollers are members of the COPS™ family, fabricated using N-channel, silicon gate MOS technology. Each microcontroller contains all system timing, internal logic, RAM and I/O necessary to implement dedicated control functions in a variety of applications, and is identical to the COP440/COP340 devices, except that the ROM has been removed; pins have been added to output the ROM address and to input ROM data. In a system, the COP404 will perform exactly as the COP440; this important benefit facilitates development and debug of a COP440 program prior to masking the final part. Features include single supply operation, various output configurations, and an instruction set, internal architecture, and I/O scheme designed to facilitate keyboard input, display output and data manipulation. Standard test procedures and reliable high-density fabrication techniques provide the medium to large volume customers with a controller-oriented processor at a low end-product cost.

For extended temperature range (−40°C to +85°C) COP304 available on special order.

Features

- Exact circuit equivalent of COP440
- Standard 48-pin dual-in-line package
- Interfaces with standard PROM or ROM
- Enhanced, more powerful instruction set
- 160 × 4 RAM, addresses up to 2k × 8 ROM
- MICROBUS™ compatible
- Zero-crossing detect circuitry with hysteresis
- True multi-vectored interrupt from four selectable sources (plus restart)
- Four-level subroutine stack (in RAM)
- 4 μs cycle time
- Single supply operation (4.5V–6.3V)
- Programmable time-base counter for real-time processing
- Internal binary counter/register with MICROWIRE™ compatible serial I/O
- General purpose and TRI-STATE® outputs
- TTL/CMOS compatible in and out
- Software/hardware compatible with other members of COP400 family
- Compatible dual CPU device available

Block Diagram



TL/DD/6916–1

FIGURE 1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Zero-Crossing Detect Pin Relative to GND	−1.2V to +15V
Voltage at Any Other Pin Relative to GND	−0.5V to +7V
Ambient Operating Temperature	0°C to +70°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C
Power Dissipation	0.75W at 25°C 0.4W at 70°C
Total Source Current	150 mA
Total Sink Current	90 mA

Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 6.3V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Operating Voltage (V_{CC})	(Note 3)	4.5	6.3	V
Power Supply Ripple	(Peak to Peak)		0.4	V
Operating Supply Current	(All Inputs and Outputs Open)			
	T_A = 0°C		44	mA
	T_A = 25°C		37	mA
	T_A = 70°C		30	mA
V_R RAM Power Supply Current	V_R = 3.3V		3	mA
Input Voltage Levels				
CKI Input Levels (÷16)				
Logic High (V_{IH})	V_{CC} = Max.,	2.5		V
Logic High (V_{IH})	V_{CC} = 5V±5%	2.0		V
Logic Low (V_{IL})		−0.3	0.4	V
$\overline{\text{RESET}}$ Input Levels	(Schmitt Trigger Input)			
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.6	V
Zero-Crossing Detect Input (IN_1)	Zero-Crossing Interrupt Input; INIL Instruction			
Trip Point		−0.15	0.15	V
Logic High (V_{IH}) Limit			12	V
Logic Low (V_{IL}) Limit		−0.8		V
IN_1				
Logic High	Interrupt Input; ININ Instruction;	3.0		V
Logic Low	MICROBUS Input	−0.3	0.8	V
All Other Inputs				
Logic High	V_{CC} = Max.	2.5		V
Logic High	V_{CC} = 5V ±5%	2.0		V
Logic Low		−0.3	0.8	V
IN_1 Input Resistance to Ground	V_{IH} = 1.0V	1.5	4.6	kΩ
Input Load Source Current	V_{IH} = 2.0V, V_{CC} = 4.5V	14	230	μA
Input Capacitance			7.0	pF
Hi-Z Input Leakage		−1.0	+1.0	μA
Output Voltage Levels				
Standard Output				
TTL Operation				
Logic High (V_{OH})	I_{OH} = −100 μA	2.4		V
Logic Low (V_{OL})	I_{OL} = 1.6 mA		0.4	V
CMOS Operation				
Logic High (V_{OH})	I_{OH} = −10 μA	V_{CC} − 0.4		V
Logic Low (V_{OL})	I_{OL} = 10 μA		0.2	V
TRI-STATE Output				
TTL Operation				
Logic High (V_{OH})	I_{OH} = −100 μA	2.4		V
Logic Low (V_{OL})	I_{OL} = 1.6 mA		0.4	V
CMOS Operation	33 kΩ ≥ R_L ≥ 4.7 kΩ			
Logic High (V_{OH})	I_{OH} = −10 μA	V_{CC} − 0.5		V
Logic Low (V_{OL})	I_{OL} = 1.6 mA		0.4	V
Output Current Levels				
Standard Output Source Current	V_{CC} = 4.5V, V_{OH} = 2.4V	−100	−650	μA
TRI-STATE Output Leakage Current		−2.5	+2.5	μA

DC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 6.3V$ unless otherwise noted (Continued)

Parameter	Conditions	Min	Max	Units
Total Sink Current Allowed				
All I/O Combined			90	mA
Each L, R Port			20	mA
Each D, G, H Port			10	mA
SO, SK			2.5	mA
IP			1.8	mA
Total Source Current Allowed	(Note 4)			
All I/O Combined			150	mA
L Port			120	mA
L_7–L_4			70	mA
L_3–L_0			70	mA
Each L Pin			23	mA
All Other Output Pins			1.6	mA

Note 1: TRI-STATE configuration is excluded.

AC Electrical Characteristics $0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 6.3V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Instruction Cycle Time—t_E		4.0	10	μs
CKI Frequency	÷16 Mode	1.6	4.0	MHz
Duty Cycle (Note 1)	f_I = 4 MHz	30	60	%
Rise Time	f_I = 4 MHz		60	ns
Fall Time	f_I = 4 MHz		40	ns
INPUTS: *(Figure 3)*				
SI				
t_{SETUP}		0.3		μs
t_{HOLD}		300		ns
IP				
t_{SETUP}		0.25		μs
t_{HOLD}		250		ns
t_{HOLD}	From AD/$\overline{DATA}$ Rising Edge	0		ns
All Other Inputs				
t_{SETUP}		1.7		μs
t_{HOLD}		300		ns
OUTPUT PROPAGATION DELAY	Test Condition:			
IP	C_L = 50 pF, V_{OUT} = 1.5V			
t_{pd1A}, t_{pd0A}			1.94	μs
t_{pd1B}, t_{pd0B}			0.94	μs
DCK				
t_{pd1}, t_{pd0}			375	ns
AD/$\overline{DATA}$				
t_{pd1}, t_{pd0}			300	ns
SO, SK				
t_{pd1}, t_{pd0}	R_L = 2.4 kΩ		1.0	μs
All Other Outputs	R_L = 5.0 kΩ		1.4	μs
MICROBUS TIMING	C_L = 100 pF, V_{CC} = 5V±5%			
Read Operation	TRI-STATE outputs			
Chip Select Stable Before $\overline{RD}$—t_{CSR}		65		ns
Chip Select Hold Time for $\overline{RD}$—t_{RCS}		20		ns
$\overline{RD}$ Pulse Width—t_{RR}		400		ns
Data Delay from $\overline{RD}$—t_{RD}			375	ns
$\overline{RD}$ to Data Floating—t_{DF}			250	ns
Write Operation				
Chip Select Stable Before $\overline{WR}$—t_{CSW}		65		ns
Chip Select Hold Time for $\overline{WR}$—t_{WCS}		20		ns
$\overline{WR}$ Pulse Width—t_{WW}		400		ns
Data Set-Up Time for $\overline{WR}$—t_{DW}		320		ns
Data Hold Time for $\overline{WR}$—t_{WD}		100		ns
INTR Transition Time from $\overline{WR}$—t_{WI}			700	ns

Note 1: Duty Cycle = $t_{WI}/(t_{WI} + t_{WO})$.

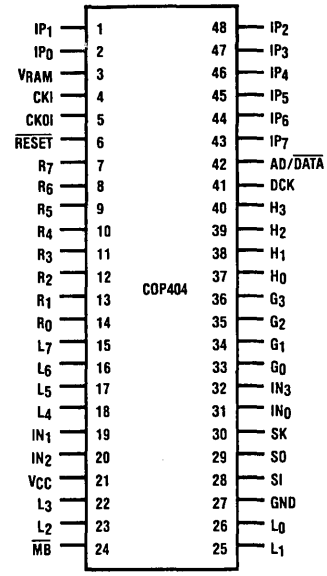
Note 2: See Figure for additional I/O Characteristics.

Note 3: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Note 4: Exercise great care not to exceed maximum device power dissipation limits when direct-driving LEDs (or sourcing similar loads) at high temperature.

Connection Diagram

Dual-In-Line Package



TL/DD/6916-2

Top View

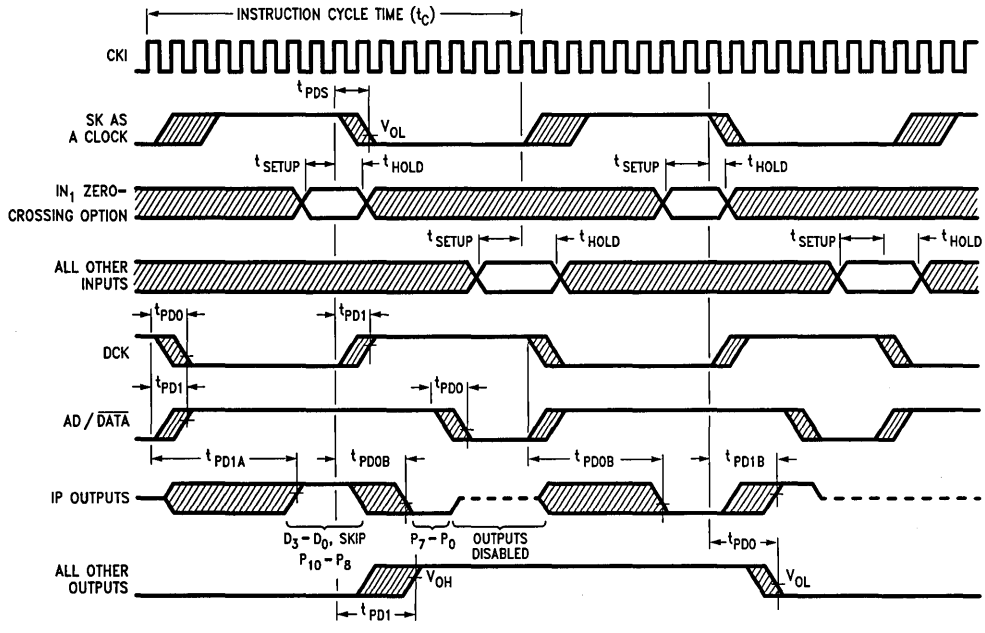
FIGURE 2

Order Number COP404N
See NS Package Number N48A

Pin Descriptions

Pin	Description
L_7–L_0	8-bit bidirectional TRI-STATE I/O port
G_3–G_0	4-bit bidirectional I/O port
IN_3–IN_0	4-bit general purpose input port
H_3–H_0	4-bit bidirectional I/O port.
R_7–R_0	8-bit bidirectional TRI-STATE I/O port
SI	Serial input
SO	Serial output (or general purpose output)
SK	Logic-controlled clock (or general purpose output)
CKI	System oscillator input
CKOI	General purpose input
V_{RAM}	Power supply to first 4 registers of RAM
$\overline{MB}$	MICROBUS function select
DCK	Clock output to latch D outputs and high order address bits
AD/$\overline{DATA}$	Address out/data in flag
IP_1–IP_0	8-bit bidirectional port for ROM address, ROM data and D outputs
$\overline{RESET}$	System reset input
V_{CC}	Power Supply
GND	Ground

Timing Diagram



TL/DD/6916-3

FIGURE 3. Input/Output Timing Diagrams (÷ 16 Mode)

Functional Description

The COP404 is a ROMless microcontroller for emulating the COP440 or for stand-alone applications. Please refer to the COP440 description for detail functional description. The following describes functions that are unique to the COP404 or are different from those in COP440. *Figures 1* and *2* show the COP404 block diagram and pin-out.

PROGRAM MEMORY

Program memory consists of 2048 bytes of external memory (on-chip in the COP440) that can be accessed through the IP port. See External Memory Interface below.

D PORT

The D3–D0 outputs are missing from this 48-pin package, but may be recovered through the IP port (see External Memory Interface below). Note that the recovered signals have the same timing but different output drive capability as those from the COP440 (see D Port Characteristics below).

MICROBUS AND ZERO-CROSSING DETECT INPUT OPTION

The MICROBUS compatible I/O, selected by a mask option on the COP440, is selected by tying the $\overline{MB}$ pin directly to ground. When the MICROBUS compatible I/O is not desired, the $\overline{MB}$ pin should be tied to V_{CC}. Note that none of the IN inputs are Hi-Z. Since zero-crossing detect input (used by INIL instruction and zero-crossing interrupt feature) is chosen for IN1, the IN1 input "1" level for ININ instruction, IN1 interrupt, and MICROBUS input is 3V. Even though the MICROBUS option and zero-crossing detector option appear on the COP404, they are mutually exclusive on the COP440.

OSCILLATOR

CKI is an external clock input signal. The clock frequency is divided by 16 to give the execution frequency.

CKO PIN OPTIONS

Two different CKO functions of the COP440 are available on the COP404. V_{RAM} supplies power to the lower four registers of RAM, and CKOI is an interrupt input or a general purpose input, reading into bit 2 of A (accumulator) through the INIL instruction.

EXTERNAL MEMORY INTERFACE

The COP404 is designed for use with an external program memory. This memory may be implemented using any devices having the following characteristics:

1. Random addressing
2. TTL-compatible TRI-STATE outputs
3. TTL-compatible inputs
4. Access time = 450 ns maximum

Typically these requirements are met using bipolar or MOS PROMs.

Figure 3 shows the timings for IP port and the external memory interface clocks—DCK and AD/$\overline{DATA}$. While DCK is low, the upper three address bits, P10–P8, of the next instruction to be executed appear at IP2–IP0 respectively; D3–D0 appear at IP7–IP4 and IP3 contains the SKIP output used by the COPS Program Development System (PDS). The rising edge of DCK clocks these data into D flip-flops, e.g., 74LS374. The timing of D port data is then the same for COP404 and COP440. After DCK has risen to a "1" level, the remaining address bits (P7–P0) appear at IP7–IP0. The falling edge of AD/$\overline{DATA}$ latches these data into flow-through latches, e.g., 74LS373. The latched addresses provide the inputs to the external memory. When AD/$\overline{DATA}$ goes low, the IP outputs are disabled and the IP lines become program memory inputs from the external memory. Note that DCK has a duty cycle of about 50% and AD/$\overline{DATA}$ has a duty cycle of about 75%. *Figure 4* shows how to emulate the COP440 using a COP404 and an EPROM as the external memory.

I/O OPTIONS

All inputs except IN1 and CKI have on-chip depletion load devices to V_{CC}. IN1 has a resistive load to GND due to the zero-crossing input. CKI is a Hi-Z input.

G and H ports have standard outputs. L and R ports have TRI-STATE outputs. IP port, DCK, AD/$\overline{DATA}$, SO and SK have push-pull outputs.

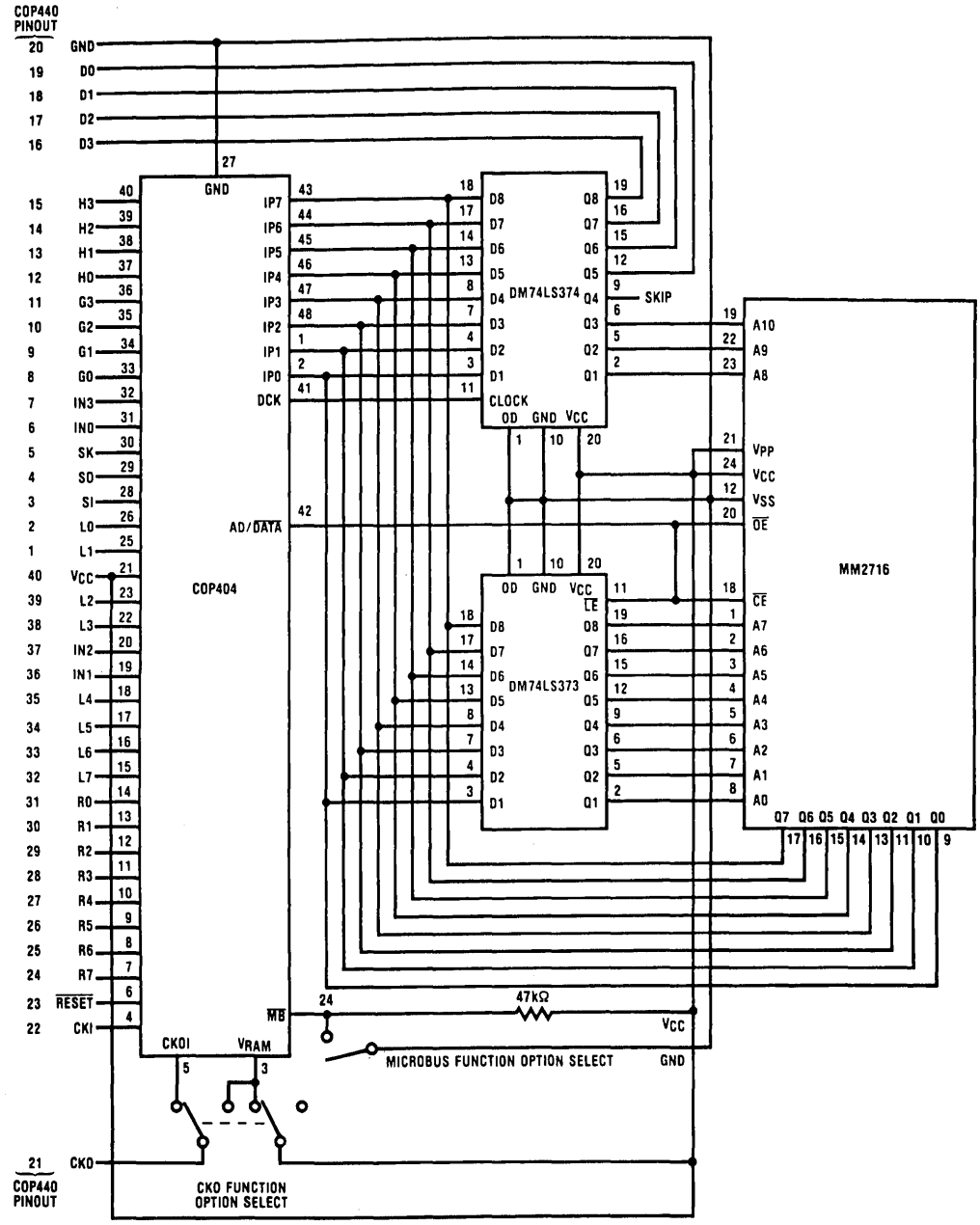
LED DRIVE

The TRI-STATE outputs of L port may be used to drive the segments of an LED display. External current limiting resistors of 100Ω must be connected between the L outputs and the LED segments.

D PORT CHARACTERISTICS

Since the D port is recovered through an external latch, the output drive is that of the latch and not that of COP440. Using the set-up as shown in *Figure 4*, at an output "0" level of 0.4V, the 74LS374 may sink 10 times as much current as the COP440. At an output "1" level of 2.4V, the 74LS374 may source 10 times as much current as the COP440. On the other hand, the output "1" level of 74LS374 latch does not go to V_{CC} without an external pull-up resistor. In order to better approximate the COP440 output characteristics, add a 74C906 buffer to the output of the 74LS374, thus emulating an open drain D output. A pull-up resistor of 10k should be added to the input of the buffer. To emulate the standard output, add a pull-up resistor between 2.7k and 15k to the output of the 74C906.

Functional Description (Continued)



TL/DD/6916-4

FIGURE 4. COP404 Used to Emulate a COP440

Option Table

COP404 MASK OPTIONS

The following COP440 options have been implemented in the COP404.

Option	Value	Comment
Option 1–2	= 3	L outputs are TRI-STATE
Option 3	= 0	SI has load to V_{CC}
Option 4	= 2	SO is push-pull output
Option 5	= 2	SK is push-pull output
Option 6	= 0	IN0 has load to V_{CC}
Option 7	= 0	IN3 has load to V_{CC}
Option 8–11	= 0	G outputs are standard
Option 12–15	= 0	H outputs are standard
Option 16–19	= N/A	D outputs are derived from external latch, see *Figure 4*
Option 20	= N/A	GND—No option
Option 21	= 1,2	CKO is replaced by V_{RAM} and CKOI
Option 22	= 0	CKI is input clock divided by 16
Option 23	= 0	$\overline{\text{RESET}}$ has load to V_{CC}
Option 24–31	= 3	R outputs are TRI-STATE
Option 32–35	= 3	L outputs are TRI-STATE
Option 36	= 2	IN1 is zero-crossing detect input
Option 37	= 0	IN2 has load to V_{CC}
Option 38–39	= 3	L outputs are TRI-STATE
Option 40	= N/A	V_{CC}—No option available
Option 41	= 0,1	MICROBUS option is pin selectable
Option 42–48	= 0	Inputs have standard TTL levels
Option 49	= N/A	No option available
Option 50	= N/A	48-pin package

National Semiconductor Corporation

PRELIMINARY

microCMOS

COP404C ROMless CMOS Microcontrollers

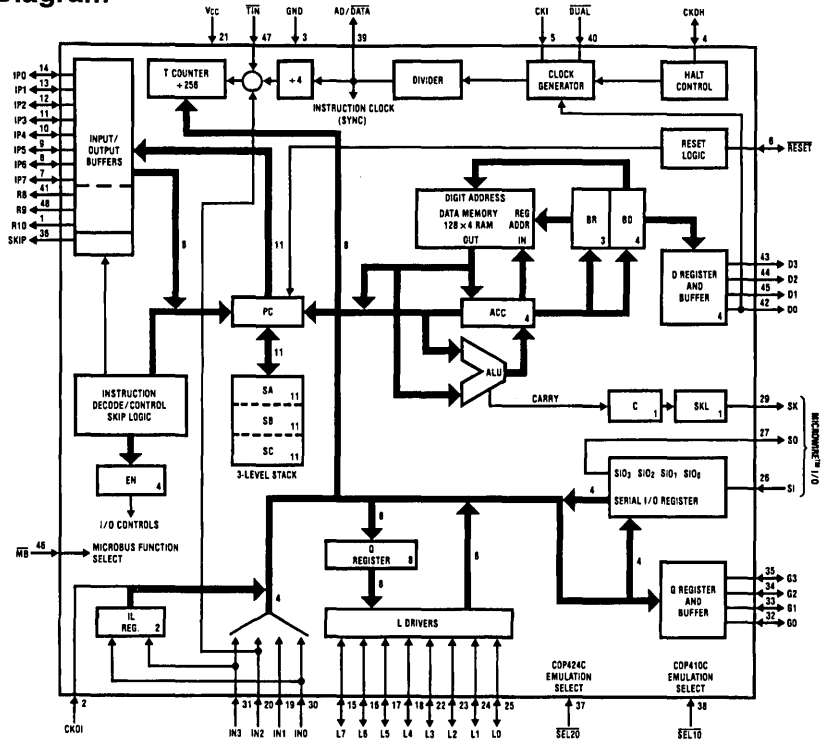
General Description

The COP404C ROMless Microcontroller is a member of the COPS™ family, fabricated using double-poly, silicon gate CMOS (microCMOS) technology. The COP404C contains CPU, RAM, I/O and is identical to a COP444C device except the ROM has been removed and pins have been added to output the ROM address and to input the ROM data. The COP404C can be configured, by means of external pins, to function as a COP444C, a COP424C, or a COP410C. Pins have been added to allow the user to select the various functional options that are available on the family of mask-programmed CMOS parts. The COP404C is primarily intended for use in the development and debug of a COP program for the COP444C/445C, COP424C/425C, and COP410C/411C devices prior to masking the final part. The COP404C is also appropriate in low volume applications or when the program might be changing.

Features

- Accurate emulation of the COP444C, COP424C and COP410C
- Lowest Power Dissipation (50 μW typical)
- Fully static (can turn off the clock)
- Power saving IDLE state and HALT mode
- 4 μs instruction time, plus software selectable clocks
- 128 × 4 RAM, addresses 2k × 8 ROM
- True vectored interrupt, plus restart
- Three-level subroutine stack
- Single supply operation (2.4V to 5.5V)
- Programmable read/write 8-bit timer/event counter
- Internal binary counter register with MICROWIRE™ serial I/O capability
- General purpose and TRI-STATE® outputs
- LSTTL/CMOS compatible
- MICROBUS™ compatible
- Software/hardware compatible with other members of the COP400 family

Block Diagram



TL/DD/5530–1

FIGURE 1. Block Diagram

Absolute Maximum Ratings

Supply Voltage	6V	Operating temperature range	0° to +70°C
Voltage at any pin	−0.3V to V_{CC} + 0.3V	Storage temperature range	−65°C to +150°C
Total Allowable Source Current	25 mA	Lead temperature (soldering, 10 sec.)	300°C
Total Allowable Sink Current	25 mA		

DC Electrical Characteristics $0°C \le T_a \le 70°C$ unless otherwise specified

Parameter	Conditions	Min	Max	Units
Operating Voltage		2.4	5.5	V
Power Supply Ripple (Note 5)	peak to peak		0.1 V_{CC}	V
Supply Current (Note 1)	V_{CC}=2.4V, t_c=64 μs		120	μA
	V_{CC}=5.0V, t_c=16 μs		700	μA
	V_{CC}=5.0V, t_c=4 μs (T_c is instruction cycle time)		3000	μA
HALT Mode Current (Note 2)	V_{CC}=5.0V, F_{IN}=0 kHz, T_A=25°C		20	μA
	V_{CC}=2.4V, F_{IN}=0 kHz, T_A=25°C		6	μA
Input Voltage Levels				
RESET, D0 (clock input)				
CKI				
Logic High		0.9 V_{CC}		V
Logic Low			0.1 V_{CC}	V
All other inputs (Note 7)				
Logic High		0.7 V_{CC}		V
Logic Low			0.2 V_{CC}	V
Input Pull-up current	V_{CC}=4.5V, V_{IN}=0	30	330	μA
Hi-Z input leakage		−1	+1	μA
Input capacitance (Note 4)			7	pF
Output Voltage Levels	Standard outputs			
LSTTL Operation	V_{CC}=5.0V ±10%			
Logic High	I_{OH}=−100 μA	2.7		V
Logic Low	I_{OL}=400 μA		0.4	V
CMOS Operation				
Logic High	I_{OH}=−10 μA	V_{CC}−0.2		V
Logic Low	I_{OL}=10 μA		0.2	V
Output current levels				
Sink (Note 6)	V_{CC}=4.5V, V_{OUT}=V_{CC}	1.2		mA
	V_{CC}=2.4V, V_{OUT}=V_{CC}	0.2		mA
Source (Standard option)	V_{CC}=4.5V, V_{OUT}=0V	0.5		mA
	V_{CC}=2.4V, V_{OUT}=0V	0.1		mA
Source (Low current option)	V_{CC}=4.5V, V_{OUT}=0V	30	330	μA
	V_{CC}=2.4V, V_{OUT}=0V	6	80	μA
Allowable Sink/Source current per pin (Note 6)			5	mA
Allowable Loading on CKOH			100	pF
Current needed to over-ride HALT (Note 3)				
To continue	V_{CC}=4.5V, V_{IN}=2V_{CC}		.7	mA
To halt	V_{CC}=4.5V, V_{IN}=7V_{CC}		1.6	mA
TRI-STATE leakage current		−2.5	+2.5	μA

Note: Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.

COP404C

AC Electrical Characteristics $0°C \le T_A \le 70°C$ unless otherwise specified

Parameter	Conditions	Min	Max	Units
Instruction Cycle	$V_{CC} \ge 4.5V$	4	DC	μs
Time (t_c)	$4.5V > V_{CC} \ge 2.4V$	16	DC	μs
Operating CKI	$V_{CC} \ge 4.5V$	DC	1.0	MHz
Frequency	$4.5V > V_{CC} \ge 2.4V$	DC	250	kHz
Duty Cycle (Note 4)	$f_1 = 4$ MHz	40	60	%
Rise Time (Note 4)	$f_1 = 4$ MHz external clock		60	ns
Fall Time (Note 4)	$f_1 = 4$ MHz external clock		40	ns
Instruction Cycle	$R = 30k$, $V_{CC} = 5V$			
Time using D0 as a	$C = 82$ pF	8	16	μs
RC Oscillator Dual-				
Clock Input (Note 4)				
INPUTS: (See *Fig. 3*)				
t_{SETUP}				
G Inputs	$V_{CC} \ge 4.5V$	$T_c/4 + .7$		μs
SI Input	$V_{CC} \ge 4.5V$	0.3		μs
IP Input	$V_{CC} \ge 4.5V$	1.0		μs
All Others	$V_{CC} \ge 4.5V$	1.7		μs
t_{HOLD}	$V_{CC} \ge 4.5V$	0.25		μs
	$4.5V > V_{CC} \ge 2.4V$	1.0		μs
OUTPUT PROPAGATION DELAY	$V_{OUT} = 1.5V$, $C_L = 100$ pF, $R_L = 5K$			
IP7–IP0, A10–A8, SKIP t_{PD1}, t_{PD0}	$V_{CC} \ge 4.5V$		1.94	μs
	$4.5V > V_{CC} \ge 2.4V$		7.75	μs
AD/$\overline{DATA}$ t_{PD1}, t_{PD0}	$V_{CC} \ge 4.5V$		375	ns
	$4.5V > V_{CC} \ge 2.4V$		1.5	μs
ALL OTHER OUTPUTS t_{PD1}, t_{PD0}	$V_{CC} > 4.5V$		1.0	μs
	$4.5V > V_{CC} \ge 2.4V$		4.0	μs
MICROBUS TIMING Read Operation (*Fig. 4*)	$C_L = 50$ pF, $V_{CC} = 5V \pm 5\%$			
Chip select stable before $\overline{RD}$ $-t_{CSR}$		65		ns
Chip select hold time for $\overline{RD}$ $-t_{RCS}$		20		ns
$\overline{RD}$ pulse width $-t_{RR}$		400		ns
Data delay from $\overline{RD}$ $-t_{RD}$			375	ns
$\overline{RD}$ to data floating $-t_{DF}$ (Note 4)			250	ns
Write Operation (*Fig. 5*)				
Chip select stable before $\overline{WR}$ $-t_{CSW}$		65		ns
Chip select hold time for $\overline{WR}$ $-t_{WCS}$		20		ns
$\overline{WR}$ pulse width $-t_{WW}$		400		ns
Data set-up time for $\overline{WR}$ $-t_{DW}$		320		ns
Data hold time for $\overline{WR}$ $-t_{WD}$		100		ns
INTR transition time from $\overline{WR}$ $-t_{WI}$			700	ns

Note 1: Supply current is measured after running for 2000 cycle times with a square-wave clock on CKI and all other pins pulled up to V_{CC} with 20k resistors. See current drain equation on page 16.

Note 2: Test conditions: All inputs tied to V_{CC}; L lines in TRI-STATE mode and tied to Ground; all outputs tied to Ground.

Note 3: When forcing HALT, current is only needed for a short time (approx. 200 ns) to flip the HALT flip-flop.

Note 4: This parameter is only sampled and not 100% tested. Variation due to the device included.

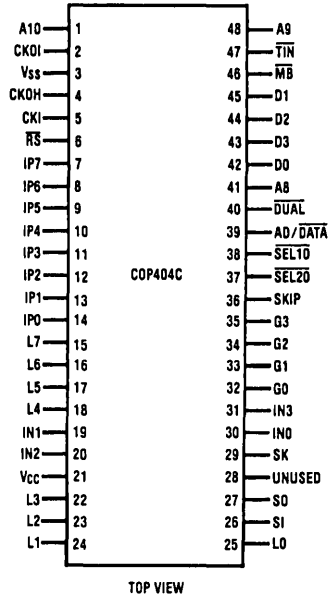
Note 5: Voltage change must be less than 0.5 volts in a 1 ms period.

Note 6: SO output sink current must be limited to keep V_{OL} less than 0.2 V_{CC} to prevent entering test mode.

Note 7: $\overline{MB}$, $\overline{TIN}$, $\overline{DUAL}$, $\overline{SEL1}$0, $\overline{SEL2}$0, input levels at V_{CC} or V_{SS}.

Connection Diagram

Dual-In-Line Package



TL/DD/5530–2

Order Number COP404CN
See NS Package Number N48A

Pin Descriptions

Pin	Description
V_{CC}	Most positive voltage
V_{SS}	Ground
CKI	Clock input
$\overline{RS}$	Reset input
CKOI	General purpose input
L0–L7	8 TRI-STATE I/O
G0–G3	4 general purpose I/O
D1–D3	3 general purpose outputs
D0	Either general purpose output or Dual-Clock RC input
IN0–IN3	4 general purpose inputs
SO	Serial data output
SI	Serial data input
SK	Serial data clock output
IP0–IP7	I/O for ROM address and data
A8, A9, A10	3 address outputs
SKIP	Skip status output
AD/$\overline{DATA}$	Clock output
$\overline{MB}$	MICROBUS select input
CKOH	Halt I/O pin
$\overline{DUAL}$	Dual-Clock select input
$\overline{TIN}$	Timer input select pin
$\overline{SEL10}$	COP410C emulation select input
$\overline{SEL20}$	COP424C emulation select input
UNUSED	Ground

FIGURE 2

The internal architecture is shown in *Figure 1*. Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1", when a bit is reset, it is a logic "0".

PROGRAM MEMORY

Program Memory consists of a 2048-byte external memory (typically PROM). Words of this memory may be program instructions, constants or ROM addressing data.

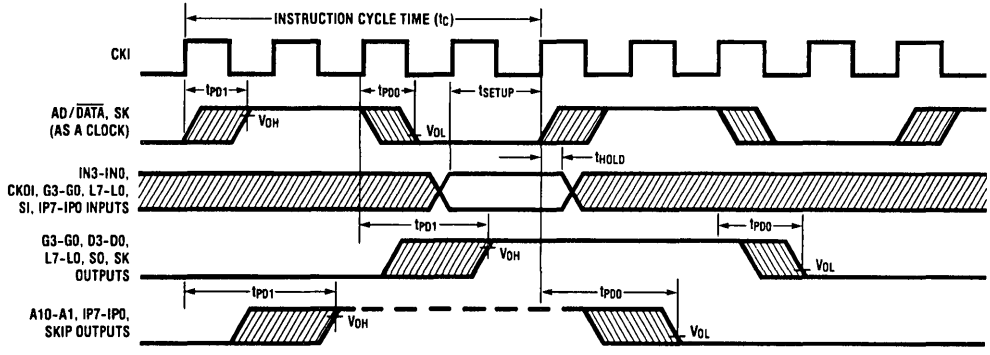
ROM addressing is accomplished by a 11-bit PC register which selects one of the 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 11-bit binary count value.

Three levels of subroutine nesting are implemented by a three level deep stack. Each subroutine call or interrupt pushes the next PC address into the stack. Each return pops the stack back into the PC register.

DATA MEMORY

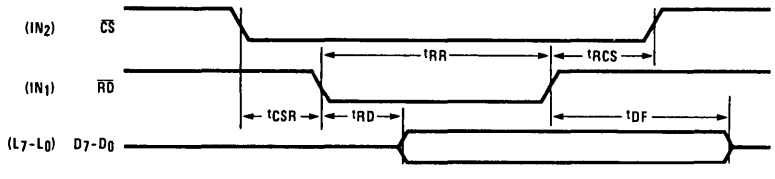
Data memory consists of a 512-bit RAM, organized as 8 data registers of 16 × 4-bit digits. RAM addressing is implemented by a 7-bit B register whose upper 3 bits (B_r) select 1 of 8 data registers and lower 4 bits (B_d) select 1 of 16 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) are usually loaded into or from, or exchanged with, the A register (accumulator), it may also be loaded into or from the Q latches or T counter or loaded from the L ports. RAM addressing may also be performed directly by the LDD and XAD instructions based upon the immediate operand field of these instructions. The B_d register also serves as a source register for 4-bit data sent directly to the D outputs.

Timing Diagrams



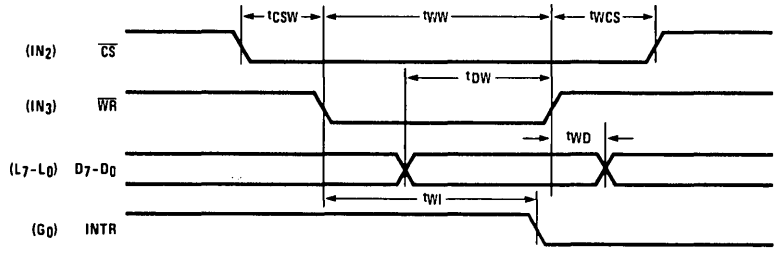
TL/DD/5530-3

FIGURE 3. Input/Output Timing



TL/DD/5530-4

FIGURE 4. MICROBUS Read Operation Timing



TL/DD/5530-5

FIGURE 5. MICROBUS Write Operation Timing

Functional Description

INTERNAL LOGIC

The processor contains its own 4-bit A register (accumulator) which is the source and destination register for most I/O, arithmetic, logic, and data memory access operations. It can also be used to load the B_r and B_d portions of the B register, to load and input 4 bits of the 8-bit Q latch or T counter, L I/O ports data, to input 4-bit G, or IN ports, and to perform data exchanges with the SIO register.

A 4-bit adder performs the arithmetic and logic functions, storing the results in A. It also outputs a carry bit to the 1-bit C register, most often employed to indicate arithmetic overflow. The C register in conjunction with the XAS instruction and the EN register, also serves to control the SK output.

The 8-bit T counter is a binary up counter which can be loaded to and from M and A using CAMT and CTMA instructions. This counter may be operated in two modes: as a timer if $\overline{TIN}$ pin is tied to Ground or as an external event counter if $\overline{TIN}$ pin is tied to V_{CC}. When the T counter overflows, an overflow flag will be set (see SKT and IT instructions below). The T counter is cleared on reset. A functional block diagram of the timer/counter is illustrated in *Figure 10a.*

Four general-purpose inputs, IN3–IN0, are provided. IN1, IN2 and IN3 may be selected (by pulling $\overline{MB}$ pin low) as Read Strobe, Chip Select, and Write Strobe inputs, respectively, for use in MICROBUS application.

The D register provides 4 general-purpose outputs and is used as the destination register for the 4-bit contents of B_d. In the dual clock mode, D0 latch controls the clock selection (see dual oscillator below).

The G register contents are outputs to a 4-bit general-purpose bidirectional I/O port. G0 may be selected as an output for MICROBUS applications.

The Q register is an internal, latched, 8-bit register, used to hold data loaded to or from M and A, as well as 8-bit data from ROM. Its contents are outputted to the L I/O ports when the L drivers are enabled under program control. With the MICROBUS option selected, Q can also be loaded with the 8-bit contents of the L I/O ports upon the occurrence of a write strobe from the host CPU.

The 8 L drivers, when enabled, output the contents of latched Q data to the L I/O port. Also, the contents of L may be read directly into A and M. As explained above, the MICROBUS option allows L I/O port data to be latched into the Q register.

The SIO register functions as a 4-bit serial-in/serial-out shift register for MICROWIRE™ I/O and COPS peripherals, or as a binary counter (depending on the contents of the EN register). Its contents can be exchanged with A.

The XAS instruction copies C into the SKL latch. In the counter mode, SK is the output SKL; in the shift register mode, SK outputs SKL ANDed with the clock.

EN is an internal 4-bit register loaded by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register:

0. The least significant bit of the enable register, EN0, selects the SIO register as either a 4-bit shift register or a 4-bit binary counter. With EN0 set, SIO is an asynchronous binary counter, decrementing its value by one upon each low-going pulse ("1" to "0") occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output equals the value of EN3. With EN0 reset, SIO is a serial shift register left shifting 1 bit each instruction cycle time. The data present at SI goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. The SK outputs SKL ANDed with the instruction cycle clock.
1. With EN1 set, interrupt is enabled. Immediately following an interrupt, EN1 is reset to disable further interrupts.
2. With EN2 set, the L drivers are enabled to output the data in Q to the L I/O port. Resetting EN2 disables the L drivers, placing the L I/O port in a high-impedance input state.
3. EN3, in conjunction with EN0, affects the SO output. With EN0 set (binary counter option selected) SO will output the value loaded into EN3. With EN0 reset (serial shift register option selected), setting EN3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN3 with the serial shift register option selected disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains set to "0".

INTERRUPT

The following features are associated with interrupt procedure and protocol and must be considered by the programmer when utilizing interrupts.

a. The interrupt, once recognized as explained below, pushes the next sequential program counter address (PC+1) onto the stack. Any previous contents at the bottom of the stack are lost. The program counter is set to hex address 0FF (the last word of page 3) and EN1 is reset.

b. An interrupt will be recognized only on the following conditions:
   1. EN1 has been set.
   2. A low-going pulse ("1" to "0") at least two instruction cycles wide has occurred on the IN1 input.
   3. A currently executing instruction has been completed.

TABLE I. ENABLE REGISTER MODES — BITS EN0 AND EN3

EN0	EN3	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL=1, SK=clock If SKL=0, SK=0
0	1	Shift Register	Input to Shift Register	Serial out	If SKL=1, SK=clock If SKL=0, SK=0
1	0	Binary Counter	Input to Counter	0	SK = SKL
1	1	Binary Counter	Input to Counter	1	SK = SKL

Functional Description (Continued)

4. All successive transfer of control instructions and successive LBIs have been completed (e.g. if the main program is executing a JP instruction which transfers program control to another JP instruction, the interrupt will not be acknowledged until the second JP instruction has been executed).

c. Upon acknowledgement of an interrupt, the skip logic status is saved and later restored upon popping of the stack. For example, if an interrupt occurs during the execution of an ASC (Add with Carry, Skip on Carry) instruction which results in carry, the skip logic status is saved and program control is transferred to the interrupt servicing routine at hex address 0FF. At the end of the interrupt routine, a RET instruction is executed to pop the stack and return program control to the instruction following the original ASC. At this time, the skip logic is enabled and skips this instruction because of the previous ASC carry. Subroutines should not be nested within the interrupt service routine, since their popping of the stack will enable any previously saved main program skips, interfering with the orderly execution of the interrupt routine.

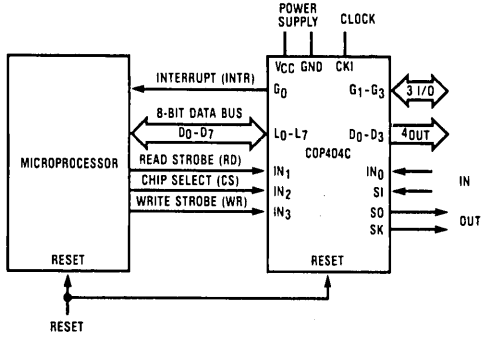
d. The instruction at hex address 0FF must be a NOP.

e. An LEI instruction may be put immediately before the RET instruction to re-enable interrupts.

MICROBUS INTERFACE

With $\overline{MB}$ pin tied to Ground, the COP404C can be used as a peripheral microprocessor device, inputting and outputting data from and to a host microprocessor (μP). IN1, IN2 and IN3 general purpose inputs become MICROBUS compatible read-strobe, chip-select, and write-strobe lines, respectively. IN1 becomes $\overline{RD}$ — a logic "0" on this input will cause Q latch data to be enabled to the L ports for input to the μP. IN2 becomes $\overline{CS}$ — a logic "0" on this line selects the COP404C and the μP peripheral device by enabling the operation of the $\overline{RD}$ and $\overline{WR}$ lines and allows for the selection of one of several peripheral components. IN3 becomes $\overline{WR}$ — a logic "0" on this line will write bus data from the L ports to the Q latches for input to the COP404C. G0 becomes INTR a "ready" output, reset by a write pulse from the μP on the $\overline{WR}$ line, providing the "handshaking" capability necessary for asynchronous data transfer between the host CPU and the COP404C.

This option has been designed for compatibility with National's MICROBUS - a standard interconnect system for 8-bit parallel data transfer between MOS/LSI CPUs and interfacing devices. (See MICROBUS National Publication). The functioning and timing relationships between the signal lines affected by this option are as specified for the MICROBUS interface, and are given in the AC electrical characteristics and shown in the timing diagrams (*Figures 4* and *5*). Connection of the COP404C to the MICROBUS is shown in *Figure 6*.



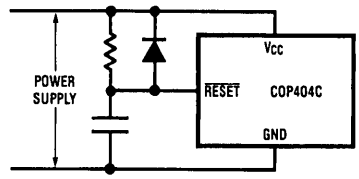
TL/DD/5530-7

FIGURE 6. MICROBUS Option Interconnect

INITIALIZATION

The external RC network shown in *Figure 7* must be connected to the $\overline{RESET}$ pin for the internal reset logic to initialize the device upon power-up. The $\overline{RESET}$ pin is configured as a Schmitt trigger input. If not used, it should be connected to V_{CC}. Initialization will occur whenever a logic "0" is applied to the $\overline{RESET}$ input, providing it stays low for at least three instruction cycle times.

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN, IL, T and G registers are cleared. The SKL latch is set, thus enabling SK as a clock output. Data Memory (RAM) is not cleared upon initialization. The first instruction at address 0 must be a CLRA (clear A register).



TL/DD/5530-8

FIGURE 7. Power-Up Circuit

TIMER

There are two modes selected by $\overline{TIN}$ pin:

a) Time-base counter ($\overline{TIN}$ pin low). In this mode, the instruction cycle frequency generated from CKI passes through a 2-bit divide-by-4 prescaler. The output of this prescaler increments the 8-bit T counter thus providing a 10-bit timer. The prescaler is cleared during execution of a CAMT instruction and on reset. For example, using a 1MHz crystal, the instruction cycle frequency of 250 kHz (divide by 4) increments the 10-bit timer every 4 μS. By presetting the counter and detecting overflow, accurate timeouts between 16 μS (4 counts) and 4.096 mS (1024 counts) are possible. Longer timeouts can be achieved by accumulating, under software control, multiple overflows.

b) External event counter ($\overline{TIN}$ pin high). In this mode, a low-going pulse ("1" to "0") at least 2 instruction cycles wide on the IN2 input will increment the 8-bit T counter.

Note: the IT instruction is not allowed in this mode.

HALT MODE

The COP404C is a FULLY STATIC circuit; therefore, the user may stop the system oscillator at any time to halt the chip. The chip may also be halted by two other ways (see *Figure 8*):

— Software HALT: by using the HALT instruction.

— Hardware HALT: by using the HALT I/O port CKOH. It is an I/O flip-flop which is an indicator of the HALT status. An external signal can over-ride this pin to start and stop the chip. By forcing CKOH high the

Functional Description (Continued)

chip will stop as soon as CKI is high and CKOH output will stay high to keep the chip stopped if the external driver returns to high impedance state.

Once in the HALT mode, the internal circuitry does not receive any clock signal and is therefore frozen in the exact state it was in when halted. All information is retained until continuing.

The chip may be awakened by one of two different methods:

— Continue function: by forcing CKOH low, the system clock will be re-enabled and the circuit will continue to operate from the point where it was stopped. CKOH will stay low.

— Restart: by forcing the $\overline{\text{RESET}}$ pin low (see Initialization)

The HALT mode is the minimum power dissipation state.

Note: if the user has selected dual-clock ($\overline{\text{DUAL}}$ pin tied to Ground) AND is forcing an external clock on D0 pin AND the COP404C is running from the D0 clock, the HALT mode - either hardware or software - will NOT be entered. Thus, the user should switch to the CKI clock to HALT. Alternatively, the user may stop the D0 clock to minimize power.

Oscillator Options

There are two basic clock oscillator configurations available as shown by *Figure 9.*

— CKI oscillator: CKI is configured as a LSTTL compatible input external clock signal. The external frequency is divided by 4 to give the instruction cycle time.

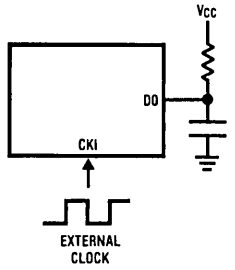
— Dual oscillator. By tying $\overline{\text{DUAL}}$ pin to Ground, pin D0 is now a single pin RC controlled Schmitt trigger oscillator input. The user may software select between the D0 oscillator (the instruction cycle time equals the D0 oscillation frequency divided by 4) by setting the D0 latch high or the CKI oscillator by resetting D0 latch low.

Note that even in dual clock mode, the counter, if used as a time-base counter, is always connected to the CKI oscillator.

For example, the user may connect up to a 1 MHz RC circuit to D0 for faster processing and a 32 kHz external clock to CKI for minimum current drain and time keeping.

Note: CTMA instruction is not allowed when the chip is running from D0 clock.

Figures 10a and *10b* show the timer and clock diagrams with and without Dual-Clock.



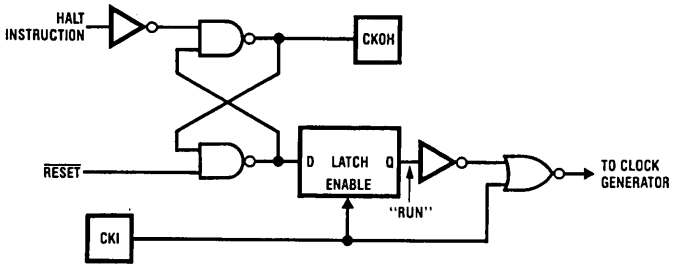
TL/DD/5530-9

R	C	Cycle Time	V_{CC}
15k	82 pF	4−9 μs	≥4.5V
30k	82 pF	8−16 μs	≥4.5V
60k	100 pF	16−32 μs	2.4−4.5V

Note: 15k≤R≤150k

50 pF≤C≤150 pF

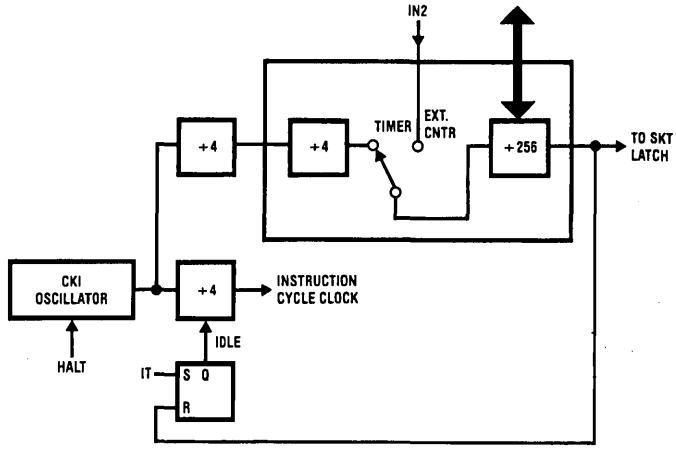
FIGURE 9. Dual-Oscillator Component Values



TL/DD/5530-10

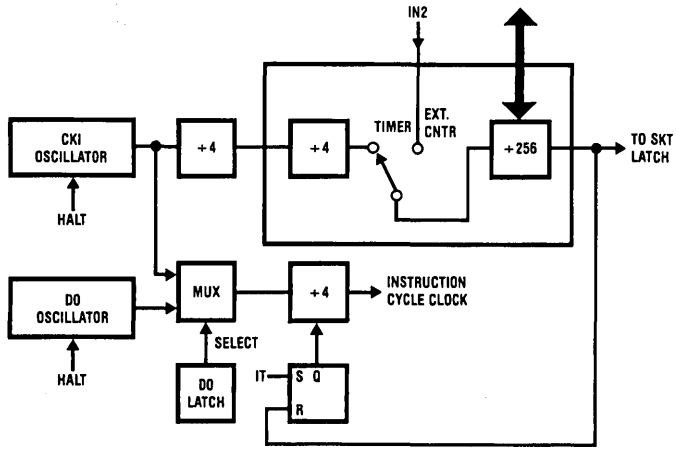
FIGURE 8. HALT Mode

Functional Description (Continued)



TL/DD/5530–11

FIGURE 10a. Clock and Timer Block Diagram without Dual-Clock



TL/DD/5530–12

Figure 10b. Clock and Timer Block Diagram with Dual-Clock

External Memory Interface

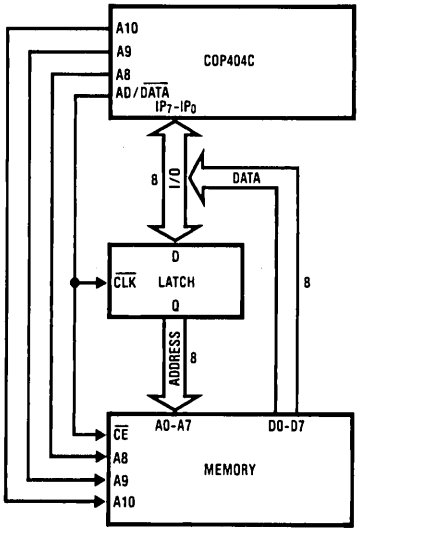
The COP404C is designed for use with an external Program Memory.

This memory may be implemented using any devices having the following characteristics:

1. random addressing
2. LSTTL or CMOS-compatible TRI-STATE outputs
3. LSTTL or CMOS-compatible inputs
4. access time = 1.0 μs max.

Typically, these requirements are met using bipolar PROMs or MOS/CMOS PROMs, EPROMs or E²PROMs.

During operation, the address of the next instruction is sent out on A10, A9, A8 and IP7 through IP0 during the time that AD/$\overline{\text{DATA}}$ is high (logic "1" = address mode). Address data on the IP lines is stored into an external latch on the high-to-low transition of the AD/DATA line; A10, A9 and A8 are dedicated address outputs, and do not need to be latched. When AD/$\overline{\text{DATA}}$ is low (logic "0" = data mode), the output of the memory is gated onto IP7 through IP0, forming the input bus. Note that AD/$\overline{\text{DATA}}$ output has a period of one instruction time, a duty cycle of approximately 50%, and specifies whether the IP lines are used for address output or data input. A simplified block diagram of the external memory interface is shown in *Figure 11.*



TL/DD/5530-13

FIGURE 11. External Memory Interface to COP404C

COP404C Instruction Set

Table II is a symbol table providing internal architecture, instruction operand and operation symbols used in the instruction set table.

Table III provides the mnemonic, operand, machine code data flow, skip conditions and description of each instruction.

Table II. Instruction Set Table Symbols

Symbol	Definition
Internal Architecture Symbols	
A	4-bit Accumulator
B	7-bit RAM address register
Br	Upper 3 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry register
D	4-bit Data output port
EN	4-bit Enable register
G	4-bit General purpose I/O port
IL	two 1-bit (IN0 and IN3) latches
IN	4-bit input port
L	8-bit TRI-STATE I/O port
M	4-bit contents of RAM addressed by B
PC	11-bit ROM address program counter
Q	8-bit latch for L port
SA	11-bit Subroutine Save Register A
SB	11-bit Subroutine Save Register B
SC	11-bit Subroutine Save Register C
SIO	4-bit Shift register and counter
SK	Logic-controlled clock output
SKL	1-bit latch for SK output
T	8-bit timer
Instruction operand symbols	
d	4-bit operand field, 0–15 binary (RAM digit select)
r	3-bit operand field, 0–7 binary (RAM register select)
a	11-bit operand field, 0–2047
y	4-bit operand field, 0–15 (immediate data)
RAM(x)	RAM addressed by variable x
ROM(x)	ROM addressed by variable x
Operational Symbols	
+	Plus
−	Minus
->	Replaces
<->	is exchanged with
=	Is equal to
−	
A	one's complement of A
⊕	exclusive-or
:	range of values

Instruction Set (Continued)

TABLE III. COP404C Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	|0011|0000|	A+C+RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	|0011|0001|	A+RAM(B) → A	None	Add RAM to A
ADT		4A	|0011|0001|	$A+10_{10}$ → A	None	Add Ten to A
AISC	y	5–	|0101| y |	A+y → A	Carry	Add Immediate. Skip on Carry (y≠ 0)
CASC		10	|0001|0000|	$\overline{A}$+RAM(B)+C → A Carry → C	Carry	Compliment and Add with Carry, Skip on Carry
CLRA		00	|0000|0000|	0 → A	None	Clear A
COMP		40	|0100|0000|	$\overline{A}$ → A	None	Ones complement of A to A
NOP		44	|0100|0100|	None	None	No Operation
RC		32	|0011|0010|	"0" → C	None	Reset C
SC		22	|0010|0010|	"1" → C	None	Set C
XOR		02	|0000|0010|	A ⊕ RAM(B) → A	None	Exclusive-OR RAM with A
TRANSFER OF CONTROL INSTRUCTIONS						
JID		FF	|1111|1111|	ROM ($PC_{10:8}$ A,M) → $PC_{7:0}$	None	Jump Indirect (note 2)
JMP	a	6– —	|0110|0|$a_{10:8}$| | $a_{7:0}$ |	a → PC	None	Jump
JP	a	—	|1| $a_{6:0}$ | (pages 2,3 only) or	a → $PC_{6:0}$	None	Jump within Page (Note 3)
		—	|11| $a_{5:0}$ | (all other pages)	a → $PC_{5:0}$		
JSRP	a	—	|10| $a_{5:0}$ |	PC+1 → SA → SB → SC 00010 → $PC_{10:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 4)
JSR	a	6– —	|0110|1|$a_{10:8}$| | $a_{7:0}$ |	PC+1 → SA → SB → SC a → PC	None	Jump to Subroutine
RET		48	|0100|1000|	SC → SB → SA → PC	None	Return from Subroutine
RETSK		49	|0100|1001|	SC → SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
HALT		33 38	|0011|0011| |0011|1000|		None	HALT processor
IT		33 39	|0011|0011| |0011|1001|		None	IDLE till timer overflows then continues
MEMORY REFERENCE INSTRUCTIONS						
CAMT		33 3F	|0011|0011| |0011|1111|	A → $T_{7:4}$ RAM(B) → $T_{3:0}$	None	Copy A, RAM to T
CTMA		33 2F	|0011|0011| |0010|1111|	$T_{7:4}$ → RAM(B) $T_{3:0}$ → A	None	Copy T to RAM, A
CAMQ		33 3C	|0011|0011| |0011|1100|	A → $Q_{7:4}$ RAM(B) → $Q_{3:0}$	None	Copy A, RAM to Q
CQMA		33 2C	|0011|0011| |0010|1100|	$Q_{7:4}$ → RAM(B) $Q_{3:0}$ → A	None	Copy Q to RAM, A
LD	r	–5	|00| r |0101| (r=0:3)	RAM(B) → A Br⊕ r → Br	None	Load RAM into A, Exclusive-OR Br with r
LDD	r,d	23 —	|0010|0011| |0| r | d |	RAM(r,d) → A	None	Load A with RAM pointed to direct by r,d
LQID		BF	|1011|1111|	ROM($PC_{10:8}$,A,M) → Q SB → SC	None	Load Q Indirect (Note 2)
RMB	0 1 2 3	4C 45 42 43	|0100|1100| |0100|0101| |0100|0010| |0100|0011|	0 → $RAM(B)_0$ 0 → $RAM(B)_1$ 0 → $RAM(B)_2$ 0 → $RAM(B)_3$	None	Reset RAM Bit

Instruction Set (Continued)

TABLE III. COP404C Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
SMB	0	4D	|0100|1101|	1 → RAM(B)$_0$	None	Set RAM Bit
	1	47	|0100|0111|	1 → RAM(B)$_1$		
	2	46	|0100|0110|	1 → RAM(B)$_2$		
	3	4B	|0100|1011|	1 → RAM(B)$_3$		
STII	y	7–	|0111| y |	y → RAM(B) Bd + 1 → Bd	None	Store Memory Immediate and Increment Bd
X	r	–6	|00| r |0110| (r=0:3)	RAM(B) ⟷ A Br ⊕ r → Br	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	r,d	23	|0010|0011|	RAM(r,d) ⟷ A	None	Exchange A with RAM pointed to directly by r,d
		—	|1| r | d |			
XDS	r	–7	|00| r |0111| (r=0:3)	RAM(B) ⟷ A Bd–1 → Bd Br ⊕ r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd. Exclusive-OR Br with r
XIS	r	–4	|00| r |0100| (r=0:3)	RAM(B) ⟷ A Bd+1 → Bd Br ⊕ r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd, Exclusive-OR Br with r
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	|0101|0000|	A → Bd	None	Copy A to Bd
CBA		4E	|0100|1110|	Bd → A	None	Copy Bd to A
LBI	r,d	—	|00| r |(d-1)| (r=0:3: d=0,9:15) or	r,d → B	Skip until not a LBI	Load B Immediate with r,d (Note 5)
		33	|0011|0011|			
		—	|1| r | d | (any r, any d)			
LEI	y	33	|0011|0011|	y → EN	None	Load EN Immediate (Note 6)
		6–	|0110| y |			
XABR		12	|0001|0010|	A ⟷ Br	None	Exchange A with Br (Note 7)
TEST INSTRUCTIONS						
SKC		20	|0010|0000|		C="1"	Skip if C is True
SKE		21	|0010|0001|		A=RAM(B)	Skip if A Equals RAM
SKGZ		33	|0011|0011|		$G_{3:0}=0$	Skip if G is Zero (all 4 bits)
		21	|0010|0001|			
SKGBZ		33	|0011|0011|	1st byte		Skip if G Bit is Zero
	0	01	|0000|0001|	2nd byte	$G_0=0$	
	1	11	|0001|0001|		$G_1=0$	
	2	03	|0000|0011|		$G_2=0$	
	3	13	|0001|0011|		$G_3=0$	
SKMBZ	0	01	|0000|0001|		RAM(B)$_0$=0	Skip if RAM Bit is Zero
	1	11	|0001|0001|		RAM(B)$_1$=0	
	2	03	|0000|0011|		RAM(B)$_2$=0	
	3	13	|0001|0011|		RAM(B)$_3$=0	
SKT		41	|0100|0001|		A time-base counter carry has occured since last test	Skip on Timer (Note 2)

Instruction Set (Continued)

TABLE III. COP404C Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
INPUT/OUTPUT INSTRUCTIONS						
ING		33	|0011|0011|	G → A	None	Input G Ports to A
		2A	|0010|1010|			
ININ		33	|0011|0011|	IN → A	None	Input IN Inputs to A
		28	|0010|1000|			
INIL		33	|0011|0011|	IL_3, CKO, "0", IL_0 → A	None	Input IL Latches to A (Note 2)
		29	|0010|1001|			
INL		33	|0011|0011|	$L_{7:4}$ → RAM(B)	None	Input L Ports to RAM,A
		2E	|0010|1110|	$L_{3:0}$ → A		
OBD		33	|0011|0011|	Bd → D	None	Output Bd to D Outputs
		3E	|0011|1110|			
OGI	y	33	|0011|0011|	y → G	None	Output to G Ports Immediate
		5–	|0101| y |			
OMG		33	|0011|0011|	RAM(B) → G	None	Output RAM to G Ports
		3A	|0011|1010|			
XAS		4F	|0100|1111|	A ⟷ SIO, C → SKL	None	Exchange A with SIO (Note 2)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicitly defined (e.g., Br and Bd are explicitly defined). Bits are numbered O to N where O signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit A register.

Note 2: For additional information on the operation of the XAS, JID, LQID, INIL, and SKT instructions, see below.

Note 3: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 4: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 5: LBI is a single-byte instruction if d = 0, 9, 10, 11, 12, 13, 14, or 15. The machine code for the lower 4 bits equals the binary value of the "d" data minus 1, e.g., to load the lower four bits of B(Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Note 6: Machine code for operand field y for LEI instruction should equal the binary value to be latched into EN, where a "1" or "0" in each bit of EN corresponds with the selection or deselection of a particular function associated with each bit. (See Functional Description, EN Register.)

Note 7: If $\overline{SEL2O}$ = 1, A ⟷ Br (0 → A3)

If $\overline{SEL2O}$ = 0, A ⟷ Br (0,0 → A3, A2).

Description of Selected Instructions

XAS INSTRUCTION

XAS (Exchange A with SIO) copies C to the SKL latch and exchanges the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. If SIO is selected as a shift register, an XAS instruction can be performed once every 4 instruction cycles to effect a continuous data stream.

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 11-bit word PC10: PC8, A, M. LQID can be used for table lookup or code conversion such as BCD to seven-segment. The LQID instruction "pushes" the stack (PC + 1 → SA → SB → SC) and replaces the least significant 8 bits of the PC as follows: A → PC (7:4), RAM(B) → PC(3:0), leaving PC(10), PC(9) and PC(8) unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SC → SB → SA → PC), restoring the saved value of PC to continue sequential program execution. Since LQID pushes SB → SC, the previous contents of SC are lost.

Note: LQID uses 2 instruction cycles if executed, one if skipped.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower 8 bits of the ROM address register PC with the contents of ROM addressed by the 11-bit word, PC10: 8, A, M. PC10, PC9 and PC8 are not affected by JID.

Note: JID uses 2 instruction cycles if executed, one if skipped.

Description of Selected Instructions (Continued)

SKT INSTRUCTION

The SKT (Skip On Timer) instruction tests the state of the T counter overflow latch (see internal logic, above), executing the next program instruction if the latch is not set. If the latch has been set since the previous test, the next program instruction is skipped and the latch is reset. The features associated with this instruction allow the processor to generate its own time-base for real-time processing, rather than relying on an external input signal

Note: If the most significant bit of the T counter is a 1 when a CAMT instruction loads the counter, the overflow flag will be set. The following sample of codes should be used when loading the counter:

```
CAMT    ; load T counter
SKT     ; skip if overflow flag is set and reset it
NOP
```

IT INSTRUCTION

The IT (idle till timer) instruction halts the processor and puts it in an idle state until the time-base counter overflows. This idle state reduces current drain since all logic (except the oscillator and time base counter) is stopped. IT instruction is not allowed if the T counter is used as an external event counter ($\overline{TIN}$ pin tied to V_{CC}).

INIL INSTRUCTION

INIL (Input IL Latches to A) inputs 2 latches, IL3 and IL0, CKOI and 0 into A. The IL3 and IL0 latches are set if a low-going pulse ("1" to "0") has occurred on the IN3 and IN0 inputs since the last INIL instruction, provided the input pulse stays low for at least two instruction cycles. Execution of an INIL inputs IL3 and IL0 into A3 and A0 respectively, and resets these latches to allow them to respond to subsequent low-going pulses on the IN3 and IN0 lines. The state of CKOI is input into A2. A 0 is input into A1. IL latches are cleared on reset.

Instruction Set Notes

a. The first word of a program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, they are still fetched from the program memory. Thus program paths take the same number of cycles whether instructions are skipped or executed except for JID, and LQID.

c. The ROM is organized into pages of 64 words each. The Program Counter is a 11-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID, or LQID is the last word of a page, it operates as if it were in the next page. For example: a JP located in the last word of a page will jump to a location in the next page. Also, a JID or LQID located in the last word of every fourth page (i.e. hex address 0FF, 1FF, 2FF, 3FF, 4FF, etc.) will access data in the next group of four pages.

Power Dissipation

The lowest power drain is when the clock is stopped. As the frequency increases so does current. Current is also lower at lower operating voltages. Therefore, for minimum power dissipation, the user should run at the lowest speed and voltage that his application will allow. The user should take care that all pins swing to full supply levels to insure that outputs are not loaded down and that inputs are not at some intermediate level which may draw current. Any input with a slow rise or fall time will draw additional current. For example, an RC oscillator on D0 will draw more current than a square wave clock input since it is a slow rising signal.

If using an external square wave oscillator, the following equation can be used to calculate the COP404C operating current drain:

$$I_{co} = I_q + V \times 40 \times F_i + V \times 1400 \times F_i / 4$$

where:

I_{co} = chip operating current drain in microamps

I_q = quiescent leakage current (from curve)

F_i = CKI frequency in MegaHertz

V = chip V_{CC} in volts

For example at 5 volts V_{CC} and 400 kHz:

$$I_{co} = 20 + 5 \times 40 \times .4 + 5 \times 1400 \times .4 / 4$$

$$I_{co} = 20 + 80 + 700 = 800\ \mu A$$

at 2.4 volts V_{CC} and 30 kHz:

$$I_{co} = 6 + 2.4 \times 40 \times .03 + 2.4 \times 1400 \times .03/4$$

$$I_{co} = 6 + 2.88 + 25.2 = 34.08\ \mu A$$

If an IT instruction is executed, the chip goes into the IDLE mode until the timer overflows. In IDLE mode, the current drain can be calculated from the following equation:

$$I_{ci} = I_q + V \times 40 \times F_i$$

For example, at 5 volts V_{CC} and 400 kHz

$$I_{ci} = 20 + 5 \times 40 \times .4 = 100\ \mu A$$

The total average current will then be the weighted average of the operating current and the idle current:

$$I_{ta} = I_{co} \times \frac{To}{To + Ti} + I_{ci} \times \frac{Ti}{To + Ti}$$

where:

I_{ta} = total average current

I_{co} = operating current

I_{ci} = idle current

T_o = operating time

T_i = idle time

I/O OPTIONS

COP404C outputs have the following configurations, illustrated in *Figure 12*.

a. Standard — A CMOS push-pull buffer with an N-channel device to ground in conjunction with a P-channel device to V_{CC}, compatible with CMOS and LSTTL. (Used on SO, SK, AD/$\overline{DATA}$, SKIP, A10:8 and D outputs.)

b. Low Current — This is the same configuration as a. above except that the sourcing current is much less. (Used on G outputs.)

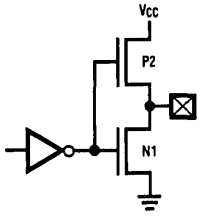
c. Standard TRI-STATE L Output — A CMOS output buffer similar to a. which may be disabled by program control. (Used on L outputs.)

All inputs have the following configuration:

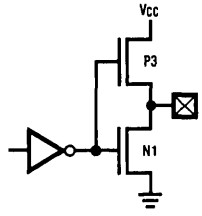
d. Input with on chip load device to V_{CC}. (Used on CKOI.)

e. HI-Z input which must be driven by the users logic. (Used on CKI, $\overline{RESET}$, IN, SI, $\overline{DUAL}$, $\overline{TIN}$, $\overline{MB}$, $\overline{SEL10}$ and $\overline{SEL20}$ inputs.)

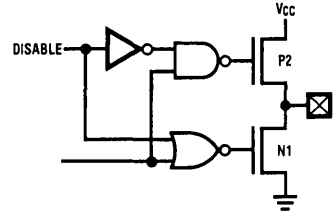
All output drivers use one or more of three common devices numbered 1 to 3. Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figure 13* for each of these devices to allow the designer to effectively use these I/O configurations.



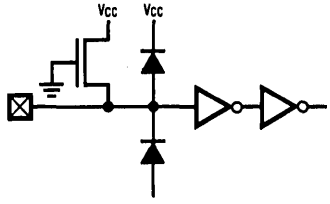
a. Standard Push-Pull Output



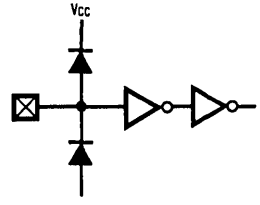
b. Low Current Push-Pull Output



Standard TRI-STATE "L" Output



d. Input with Load

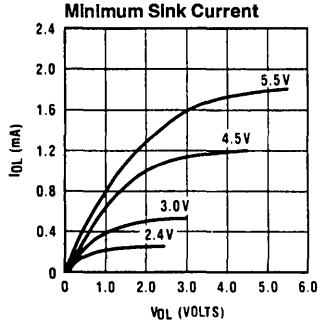


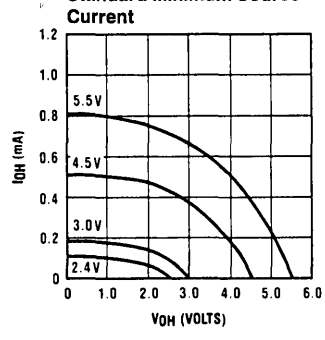
e. Hi-Z Input

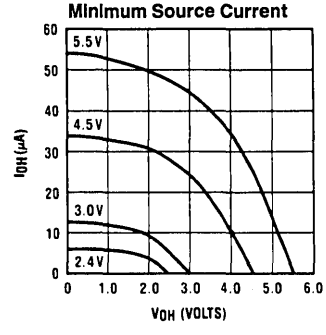
TL/DD/5530-15

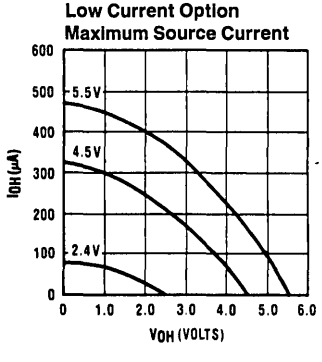
FIGURE 12. Input/Output Configurations

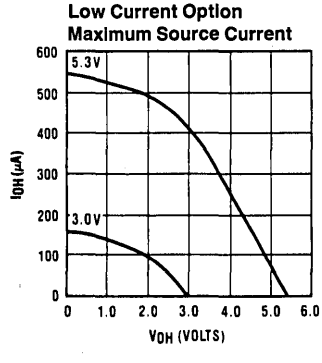
Typical Performance Characteristics

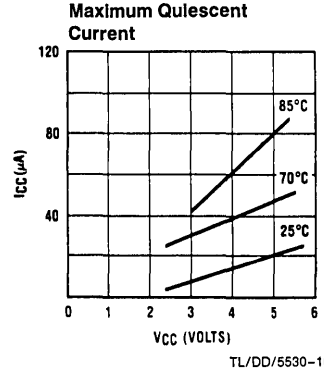












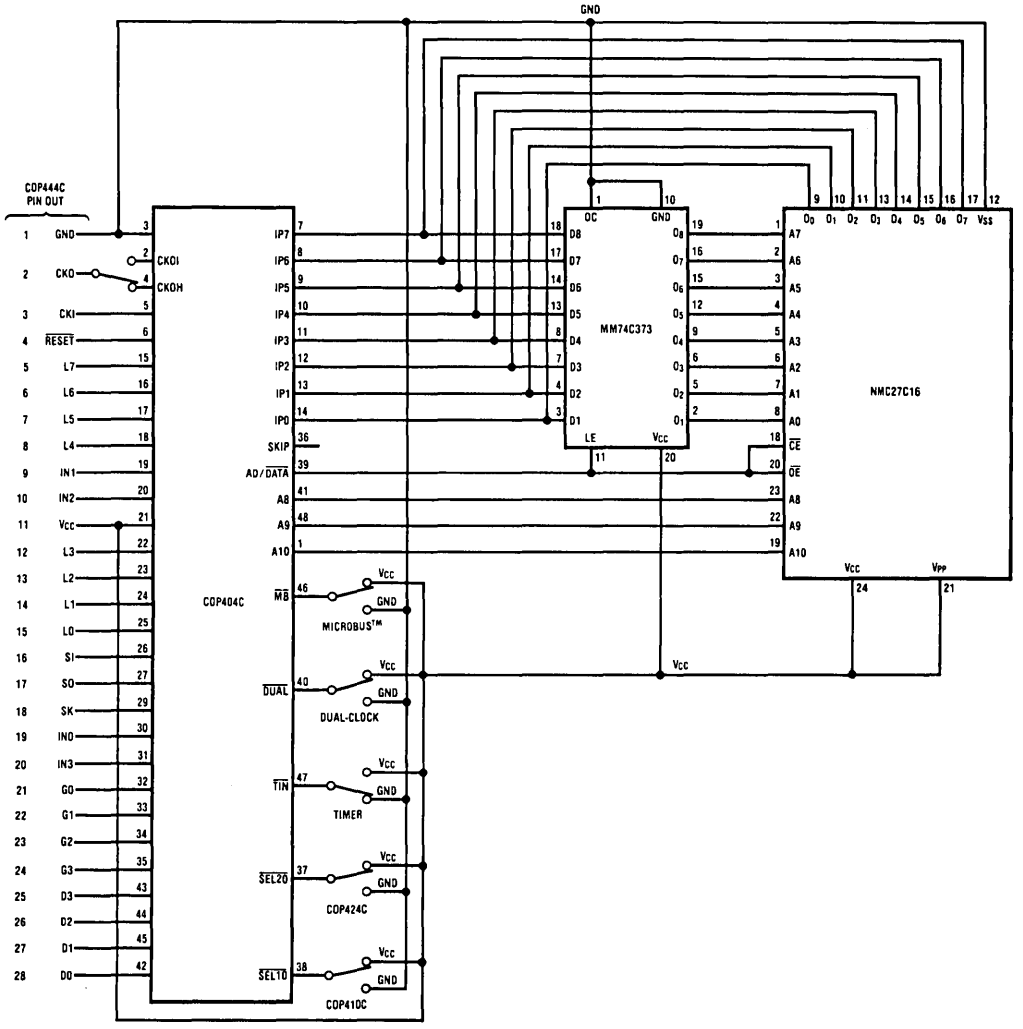
TL/DD/5530-16

FIGURE 13. Input/Output Characteristics

Emulation

The COP404C may be used to exactly emulate the COP444C/445C, COP424C/425C, and COP410C/411C. However, the Program Counter always addresses 2k of external ROM whatever chip is being emulated. *Figure 14* shows the interconnect to implement a hardware emulation. This connection uses a NMC27C16 EPROM as external memory. Other memory can be used such as bipolar PROM or RAM.

Pins IP7–IP0 are bidirectional inputs and outputs. When the AD/$\overline{\text{DATA}}$ clocking output turns on, the EPROM drivers are disabled and IP7–IP0 output addresses. The 8-bit latch (MM74C373) latches the addresses to drive the memory.



TL/DD/5530–14

FIGURE 14. COP404C Used To Emulate A COP444C

Emulation (Continued)

When AD/$\overline{DATA}$ turns off, the EPROM is enabled and the IP7–IP0 pins will input the memory data. A10, A9 and A8 output the most significant address bits to the memory. (SKIP output may be used for program debug if needed.)

— CKI is divided by 4. Other divide-by are emulated by external divider.

— CKO can be emulated as a general purpose input by using CKOI or as a Halt I/O port by using CKOH.

— $\overline{MB}$ pin can be pulled low if the MICROBUS feature of the COP444C and COP424C is needed. Othewise it should be high.

— $\overline{DUAL}$ pin can be pulled low if the Dual-Clock feature of the COP444C and COP424C is needed. Otherwise it should be high.

— $\overline{TIN}$ pin controls the input of the 8-bit timer of the COP444C and COP424C (internal timer if $\overline{TIN}$ is low, external event counter if $\overline{TIN}$ is high).

— The $\overline{SEL10}$ and $\overline{SEL20}$ inputs are used to emulate the COP444C/445C, COP424C/425C, or COP410C/411C.

- When emulating the COP444C/445C, the user must configure $\overline{SEL20}$ = 1 and $\overline{SEL10}$ = 1.
- When emulating the COP424C/425C, the user must configure $\overline{SEL20}$ = 0 and $\overline{SEL10}$ = 1. In this mode, the user RAM is physically halved. As in the COP424C/425C, the user has 64 digits (256 bits) of RAM available. Pin A10 should not be connected to the program memory (most significant address bit of the program memory should be grounded if using a 2k×8 memory).
- When emulating the COP410C/411C, the user must configure $\overline{SEL20}$ = 0 and $\overline{SEL10}$ = 0. In this mode, the user has 32 digits (128 bits) of RAM available organized in the same way as the COP410C/411C - 4 registers of 8 digits each. Pins A10 and A9 should not be connected to the program memory (the 2 most significant address bits of the program memory should be grounded).

Furthermore, the subroutine stack is decreased from 3 levels to 2 levels.

The pins $\overline{SEL10}$ and $\overline{SEL20}$ change the internal logic of the device to accurately emulate the devices as indicated above. However, the user must remember that the COP424C/425C is a subset of the COP444C/COP445C with respect to memory size. The COP410C/411C is a subset both in memory size and in function. The user must take care not to use features and instructions which are not available on the COP410C/411C (see table IV. below) when using the COP404C to emulate the COP410C/411C.

TABLE IV. FEATURES AND INSTRUCTIONS NOT AVAILABLE ON COP410C/411C.

Timer	ADT		
Dual-clock	CASC		
Interrupt	CAMT		
Microbus	CTMA		
	IT		
	LDD	r, d	
	XAD	r, d	(except 3, 15)
	XABR		
	SKT		
	ININ		
	INIL		
	OGI	y	

Option Table

COP404C MASK OPTIONS

The following COP444C options have been implemented in the COP404C:

Option value	Comment
Option 1 = 0	Ground Pin — no option available
Option 2 = 1, 2	CKO is replaced by CKOI and CKOH
Option 3 = 5	CKI is external clock input divided by 4
Option 4 = 1	$\overline{RESET}$ is Hi-Z input
Option 5–8 = 0	L outputs are standard TRI-STATE
Option 9 = 1	IN1 is a Hi-Z input
Option 10 = 1	IN2 is a Hi-Z input
Option 11 = 0	V_{CC} pin — no option available
Option 12–15 = 0	L outputs are standard TRI-STATE
Option 16 = 0	SI is a Hi-Z input
Option 17 = 0	SO is a standard output
Option 18 = 0	SK is a standard output
Option 19 = 1	IN0 is a Hi-Z input
Option 20 = 1	IN3 is a Hi-Z input
Option 21–24 = 1	G outputs are low-current
Option 25–28 = 0	D outputs are standard
Option 29 = 1	No internal initialization logic
Option 30 = 0, 1	DUAL-CLOCK is pin selectable
Option 31 = 0, 1	TIMER is pin selectable
Option 32 = 0, 1	MICROBUS is pin selectable
Option 33 = N/A	48-pin package

National Semiconductor Corporation

PRELIMINARY

COP404LSN-5 ROMless N-Channel Microcontrollers

General Description

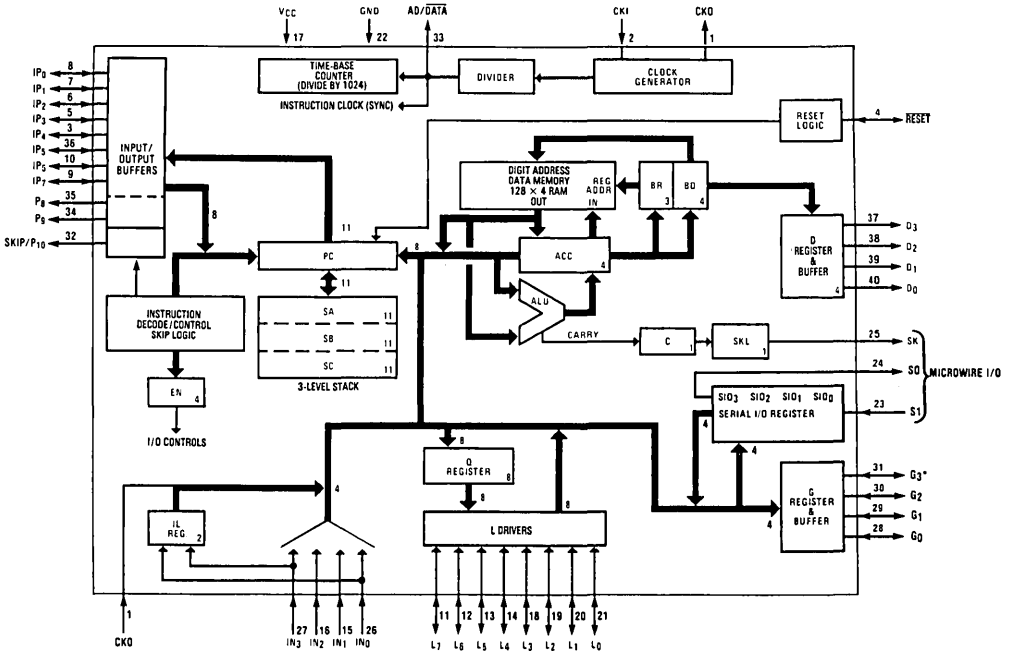
The COP404LSN-5 ROMless Microcontroller is a member of the COPS™ family, fabricated using N-channel, silicon gate MOS technology. The COP404LSN-5 contains CPU, RAM, I/O and is identical to a COP444L device except the ROM has been removed and pins have been added to output the ROM address and to input the ROM data. In a system the COP404LSNN-5 will perform exactly as the COP444L. This important benefit facilitates development and debug of a COP program prior to masking the final part. The COP404LSN-5 is also appropriate in low volume applications, or when the program might be changing. The COP404LSN-5 may be used to emulate the COP444L, COP445L, COP420L, and the COP421L.

Use COP404LSN-5 in volume applications. For extended temperature range (−40°C to +85°C), COP304L is available on a special order basis.

Features

- Exact circuit equivalent of COP444L
- Low cost
- Powerful instruction set
- 128 x 4 RAM, addresses 2048 x 8 ROM
- True vectored interrupt, plus restart
- Three-level subroutine stack
- 16 μs instruction time
- Single supply operation (4.5V–5.5V)
- Low current drain (16 mA max)
- Internal time-base counter for real-time processing
- Internal binary counter register with MICROWIRE™ compatible serial I/O
- General purpose outputs
- LSTTL/CMOS compatible in and out
- Direct drive of LED digit and segment lines
- Software/hardware compatible with other members of COP400 family

Block Diagram



TL/DD/8817–1

FIGURE 1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin Relative to GND	−0.5V to +10V
Ambient Operating Temperature	0°C to +70°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C
Power Dissipation	0.75W at 25°C 0.4W at 70°C
Total Source Current	120 mA
Total Sink Current	140 mA

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics

$4.5V \le V_{CC} \le 5.5V$; $0°C \le T_A \le 70°C$

Parameter	Conditions	Min	Max	Units
Operating Voltage (V_{CC})	(Note 2)	4.5	5.5	V
Power Supply Ripple	Peak to Peak		0.5	V
Operating Supply Current	All Inputs and Outputs Open		16	mA
Input Voltage Levels				
CKI Input Levels				
Crystal Input				
Logic High (V_{IH})		2.0		V
Logic Low (V_{IL})		−0.3	0.4	V
$\overline{\text{RESET}}$ Input Levels	Schmitt Trigger Input			
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.6	V
IP0–IP7, SI Input Levels				
Logic High	V_{CC} = 5.5V	2.4		V
Logic High	V_{CC} = 5V ±5%	2.0		V
Logic Low		−0.3	0.8	V
All Other Inputs				
Logic High	High Trip Level Options	3.6		V
Logic Low	Selected	−0.3	1.2	V
Input Capacitance			7	pF
Output Voltage Levels				
LSTTL Operation	V_{CC} = 5V ±10%			
Logic High (V_{OH})	I_{OH} = −25 μA	2.7		V
Logic Low (V_{OL})	I_{OL} = 0.36 mA		0.4	V
IP0–IP7, P8, P9, SKIP/P10	(Note 1)			
Logic High	I_{OH} = −80 μA	2.4		V
Logic Low	I_{OL} = 720 μA		0.4	V
Output Current Levels				
Output Sink Current				
SO and SK Outputs (I_{OL})	V_{CC} = 4.5V, V_{OL} = 0.4V	0.9		mA
L_0–L_7 Outputs	V_{CC} = 4.5V, V_{OL} = 0.4V	0.4		mA
G_0–G_3 and D_0–D_3 Outputs	V_{CC} = 4.5V, V_{OL} = 1.0V	15		mA
CKO	V_{CC} = 4.5V, V_{OL} = 0.4V	0.2		mA
Output Source Current				
D_0–D_3, G_0–G_3 Outputs (I_{OH})	V_{CC} = 4.5V, V_{OH} = 2.0V	−30	−250	μA
SO and SK Outputs (I_{OH})	V_{CC} = 4.5V, V_{OH} = 1.0V	−1.2		mA
L_0–L_7 Outputs	V_{CC} = 5.5V, V_{OH} = 2.0V	−1.4	−25	mA

DC Electrical Characteristics (Continued)

$0°C \le T_A \le +70°C$, $4.5V \le V_{CC} \le 5.5V$ unless otherwise noted

Parameter	Conditions	Min	Max	Units
Input Load Source Current (I_{IL})	$V_{CC} = 5.0V$, $V_{IL} = 0V$	−10	−140	μA
Total Sink Current Allowed				
All Outputs Combined			140	mA
D, G Ports			120	mA
L_7–L_4			4	mA
L_3–L_0			4	mA
All Other Pins			1.8	mA
Total Source Current Allowed				
All I/O Combined			120	mA
L_7–L_4			60	mA
L_3–L_0			60	mA
Each L Pin			30	mA
All Other Pins			1.5	mA

AC Electrical Characteristics $0°C \le T_A \le 70°C$, $4.5V \le V_{CC} \le 5.5V$ unless otherwise specified

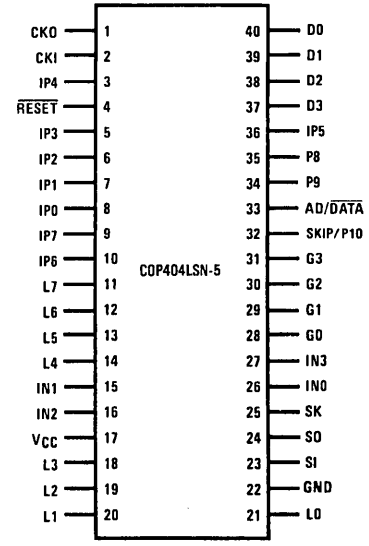
Parameter	Conditions	Min	Max	Units
Instruction Cycle Time		16	40	μs
CKI				
Input Frequency, f	(÷32 Mode)	0.8	2	MHz
Duty Cycle		30	60	%
Rise Time	$f_I = 2.0$ MHz		120	ns
Fall Time			80	ns
INPUTS:				
SI, IP7–IP0				
t_{SETUP}		2.0		μs
t_{HOLD}		1.0		μs
IN_3–IN_0, G_3–G_0, L_7–L_0				
t_{SETUP}		8.0		μs
t_{HOLD}		1.3		μs
OUTPUT PROPAGATION DELAY	Test Condition: $C_L = 50$ pF, $V_{OUT} = 1.5V$			
SO, SK Outputs	$R_L = 20$ kΩ			
t_{pd1}, t_{pd0}			4.0	μs
D_3–D_0, G_3–G_0, L_7–L_0	$R_L = 20$ kΩ			
t_{pd1}, t_{pd0}			5.6	μs
IP7–IP0, P8, P9, SKIP	$R_L = 5$ kΩ			
t_{pd1}, t_{pd0}			7.2	μs
P10	$R_L = 5$ kΩ			
t_{pd1}, t_{pd0}			6.0	μs

Note 1: COP404LSN-5 has Push-Pull drivers on these outputs.

Note 2: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Connection Diagram

Dual-In-Line Package



TL/DD/8817-2

Top View

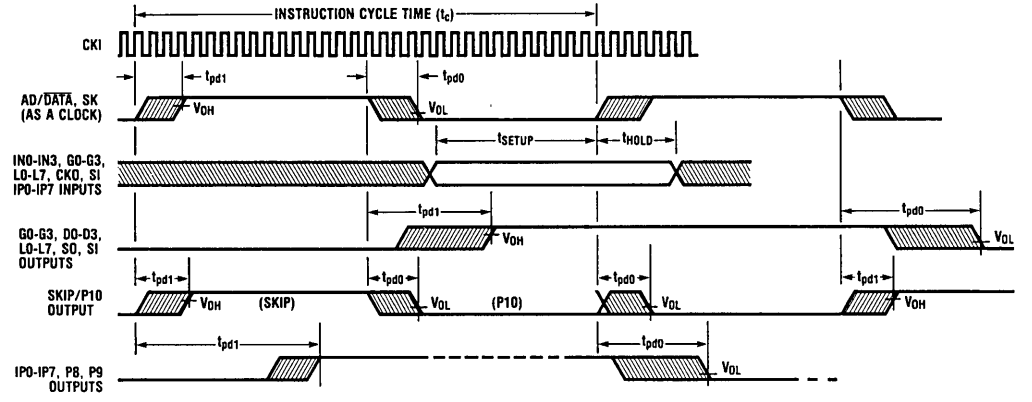
FIGURE 2

Order Number COP404LSN-5
See NS Package Number N40A

Pin Descriptions

Pin	Description
L_7–L_0	8 bidirecitonal I/O ports with TRI-STATE®
G_3–G_0	4 bidirectional I/O ports
D_3–D_0	4 general purpose outputs
IN_3–IN_0	4 general purpose outputs
SI	Serial input (or counter input
SO	Serial output (or general purpose output)
SK	Logic-controlled clock (or general purpose output)
AD/$\overline{\text{DATA}}$	Address out/data in flag
CKI	System oscillator input
CKO	System oscillator output (COP404LSN-5)
$\overline{\text{RESET}}$	System reset input
V_{CC}	Power supply
GND	Ground
IP7–IP0	8 bidirectional ROM address and data ports
P8, P9	2 ROM address outputs
SKIP/P10	Instruction skip output and most significant ROM address bit output

Timing Diagram



TL/DD/8817-3

FIGURE 3. Input/Output

Functional Description

A block diagram of the COP404LSN-5 is given in *Figure 1*. Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device. Positive logic is used. When a bit is set, it is a logic "1" (greater than 2V). When a bit is reset, it is a logic "0" (less than 0.8V).

PROGRAM MEMORY

Program Memory consists of a 2048 byte external memory. As can be seen by an examination of the COP404LSN-5 instruction set, these words may be program instructions, program data or ROM addressing data. Because of the special characteristics associated with the JP, JSRP, JID and LQID instructions, ROM must often be thought of as being organized into 32 pages of 64 words each.

ROM addressing is accomplished by an 11-bit PC register. Its binary value selects one of the 2048 8-bit words contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential 11-bit binary count value. Three levels of subroutine nesting are implemented by the 11-bit subroutine saves registers, SA, SB, and SC, providing a last-in, first-out (LIFO) hardware subroutine stack.

ROM instruction words are fetched, decoded and executed by the Instruction Decode, Control and Skip Logic circuitry.

DATA MEMORY

Data memory consists of a 512-bit RAM, organized as 8 data registers of 16 4-bit digits. RAM addressing is implemented by a 7-bit B register whose upper 3 bits (Br) select 1 of 8 data registers and lower 4 bits (Bd) select 1 of 16 4-bit digits in the selected data register. While the 4-bit contents of the selected RAM digit (M) is usually loaded into or from, or exchanged with, the A register (accumulator), it may also be loaded into or from the Q latches or loaded from the L ports. RAM addressing may also be performed directly by the LDD and XAD instructions based upon the 7-bit contents of the operand field of these instructions. The Bd register also serves as a source register for 4-bit data sent directly to the D outputs.

INTERNAL LOGIC

The 4-bit A register (accumulator) is the source and destination register for most I/O, arithmetic, logic and data memory access operations. It can also be used to load the Br and Bd portions of the B register, to load and input 4 bits of the 8-bit Q latch data, to input 4 bits of the 8-bit L I/O port data and to perform data exchanges with the SIO register.

A 4-bit adder performs the arithmetic and logic functions, storing its results in A. It also outputs a carry bit to the 1-bit C register, most often employed to indicate arithmetic overflow. The C register, in conjunction with the XAS instruction and the EN register, also serves to control the SK output. C can be outputted directly to SK or can enable SK to be a sync clock each instruction cycle time. (See XAS instruction and EN register description, below).

Four general-purpose inputs, IN_3–IN_0, are provided.

The D register provides 4 general-purpose outputs and is used as the destination register for the 4-bit contents of Bd. The D outputs can be directly connected to the digits of a multiplexed LED display.

The G register contents are outputs to 4 general-purpose bidirectional I/O ports. G I/O ports can be directly connected to the digits of a multiplexed LED display.

The Q register is an internal, latched, 8-bit register, used to hold data loaded to or from M and A, as well as 8-bit data from ROM. Its contents are output to the L I/O ports when the L drivers are enabled under program control. (See LEI instruction.)

The 8 L drivers, when enabled, output the contents of latched Q data to the L I/O ports. Also, the contents of L may be read directly into A and M. L I/O ports can be directly connected to the segments of a multiplexed LED display (using the LED Direct Drive output configuration option) with Q data being outputted to the Sa–Sg and decimal point segments of the display.

The SIO register functions as a 4-bit serial-in/serial-out shift register or as a binary counter depending on the contents of the EN register. (See EN register description, below.) Its contents can be exchanged with A, allowing it to input or output a continuous serial data stream. SIO may also be used to provide additional parallel I/O by connecting SO to external serial-in/parallel-out shift registers.

The XAS instruction copies C into the SKL latch. In the counter mode, SK is the output of SKL; in the shift register mode, SK outputs SKL ANDed with the clock.

The EN register is an internal 4-bit register loaded under program control by the LEI instruction. The state of each bit of this register selects or deselects the particular feature associated with each bit of the EN register (EN_3–EN_0).

1. The least significant bit of the enable register, EN_0, selects the SIO register as either a 4-bit shift register or a 4-bit binary counter. With EN_0 set, SIO is an asynchronous binary counter, *decrementing* its value by one upon each low-going pulse ("1" to "0") occurring on the SI input. Each pulse must be at least two instruction cycles wide. SK outputs the value of SKL. The SO output is equal to the value of EN_3. With EN_0 reset, SIO is a serial shift register shifting left each instruction cycle time. The data present at SI goes into the least significant bit of SIO. SO can be enabled to output the most significant bit of SIO each cycle time. (See 4 below.) The SK output becomes a logic-controlled clock.
2. With EN_1 set the IN_1 input is enabled as an interrupt input. Immediately following an interrupt, EN_1 is reset to disable further interrupts.
3. With EN_2 set, the L drivers are enabled to output the data in Q to the L I/O ports. Resetting EN_2 disables the L drivers, placing the L I/O ports in a high-impedance input state.

Functional Description (Continued)

4. EN_3, in conjunction with EN_0, affects the SO output. With EN_0 set (binary counter option selected) SO will output the value loaded into EN_3. With EN_0 reset (serial shift register option selected), setting EN_3 enables SO as the output of the SIO shift register, outputting serial shifted data each instruction time. Resetting EN_3 with the serial shift register option selected disables SO as the shift register output; data continues to be shifted through SIO and can be exchanged with A via an XAS instruction but SO remains reset to "0." The table below provides a summary of the modes associated with EN_3 and EN_0.

INTERRUPT

The following features are associated with the IN_1 interrupt procedure and protocol and must be considered by the programmer when utilizing interrupts.

a. The interrupt, once acknowledged as explained below, pushes the next sequential program counter address (PC+1) onto the stack, pushing in turn the contents of the other subroutine-save registers to the next lower level (PC+1 → SA → SB → SC). Any previous contents of SC are lost. The program counter is set to hex address 0FF (the last word of page 3) and EN_1 is reset.

b. An interrupt will be acknowledged only after the following conditions are met:
   1. EN_1 has been set.
   2. A low-going pulse ("1" to "0") at least two instruction cycles wide occurs on the IN_1 input.
   3. A currently executing instruction has been completed.
   4. All successive transfer of control instructions and successive LBIs have been completed (e.g., if the main program is executing a JP instruction which transfers program control to another JP instruction, the interrupt will not be acknowledged until the second JP instruction has been executed.

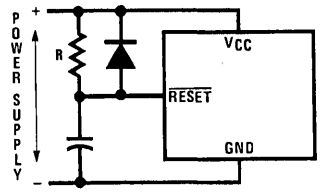
c. Upon acknowledgement of an interrupt, the skip logic status is saved and later restored upon popping of the stack. For example, if an interrupt occurs during the execution of ASC (Add with Carry, Skip on Carry) instruction which results in carry, the skip logic status is saved and program control is transferred to the interrupt servicing routine at hex address 0FF. At the *end* of the interrupt routine, a RET instruction is executed to "pop" the stack and return program control to the instruction following the original ASC. *At this time*, the skip logic is enabled and skips this instruction because of the previous ASC carry. Subroutines and LQID instructions should not be nested within the interrupt service routine, since their popping the stack will enable any previously saved main program skips, interfering with the orderly execution of the interrupt routine.

d. The first instruction of the interrupt routine at hex address 0FF must be a NOP.

e. A LEI instruction can be put immediately before the RET to re-enable interrupts.

INITIALIZATION

The Reset Logic will initialize (clear) the device upon power-up if the power supply rise time is less than 1 ms and greater than 1 μs. If the power supply rise time is greater than 1 ms, the user must provide an external RC network and diode to the $\overline{RESET}$ pin as shown below. The $\overline{RESET}$ pin is configured as a Schmitt trigger input. If the RC network is not used, the $\overline{RESET}$ pin should be left open. Initialization will occur whenever a logic "0" is applied to the $\overline{RESET}$ input, provided it stays low for at least three instruction cycle times.



TL/DD/8817–4

RC ≥ 5 × Power Supply Rise Time (R > 40k)

Upon initialization, the PC register is cleared to 0 (ROM address 0) and the A, B, C, D, EN, and G registers are cleared. The SK output is enabled as a SYNC output, providing a pulse each instruction cycle time. *Data Memory (RAM) is not cleared upon initialization.* The first instruction at address 0 must be a CLRA.

EXTERNAL MEMORY INTERFACE

The COP404LSN-5 is designed for use with an external Program Memory. This memory may be implemented using any devices having the following characteristics:

1. random addressing
2. TTL-compatible TRI-STATE outputs
3. TTL-compatible inputs
4. access time = 5 μs max.

Typically these requirements are met using bipolar or MOS PROMs.

During operation, the address of the next instruction is sent out on P10, P9, P8, and IP7 through IP0 during the time that AD/$\overline{DATA}$ is high (logic "1" = address mode). Address data on the IP lines is stored into an external latch on the high-to-low transition of the AD/$\overline{DATA}$ line; P9 and P8 are

Enable Register Modes — Bits EN_3 and EN_0

EN_3	EN_0	SIO	SI	SO	SK
0	0	Shift Register	Input to Shift Register	0	If SKL = 1, SK = CLOCK If SKL = 0, SK = 0
1	0	Shift Register	Input to Shift Register	Serial Out	If SKL = 1, SK = CLOCK If SKL = 0, SK = 0
0	1	Binary Counter	Input to Binary Counter	0	If SKL = 1, SK = 1 If SKL = 0, SK = 0
1	1	Binary Counter	Input to Binary Counter	1	If SKL = 1, SK = 1 If SKL = 0, SK = 0

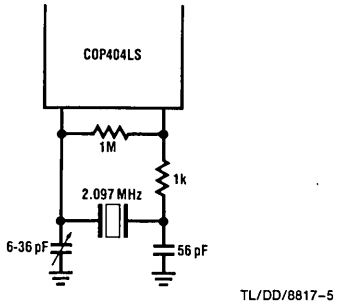
Functional Description (Continued)

dedicated address outputs, and do not need to be latched. SKIP/P10 outputs address data when AD/$\overline{\text{DATA}}$ is low. When AD/$\overline{\text{DATA}}$ is low (logic "0" = data mode), the output of the memory is gated onto IP7 through IP0, forming the input bus. Note that the AD/$\overline{\text{DATA}}$ output has a period of one instruction time, a duty cycle of approximately 50%, and specifies whether the IP lines are used for address output or instruction input.

OSCILLATOR

The basic clock oscillator configurations is shown in *Figure 4*.

Crystal Controlled Oscillator—CKI and CKO are connected to an external crystal. The instruction cycle time equals the crystal frequency divided by 32.



TL/DD/8817-5

FIGURE 4. Oscillator

INPUT/OUTPUT CONFIGURATIONS

COP404LSN-5 outputs have the following configurations, illustrated in *Figure 5*:

a. Standard—an enhancement mode device to ground in conjunction with a depletion-mode device to V_{CC}, compatible with LSTTL and CMOS input requirements. (Used on D and G outputs.)

b. Open-Drain—an enhancement-mode device to ground only, allowing external pull-up as required by the user's application.

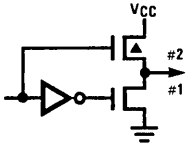
c. Push-Pull—an enhancement-mode device to ground in conjunction with a depletion-mode device paralleled by an enhancement-mode device to V_{CC}. This configuration has been provided to allow for fast rise and fall times when driving capacitive loads.

d. LED Direct Drive—an enhancement-mode device to ground and to V_{CC}, meeting the typical current sourcing requirements of the segments of an LED display. The sourcing device is clamped to limit current flow. These devices may be turned off under program control (see Functional Description, EN Register), placing the outputs in a high-impedance state to provide required LED segment blanking for a multiplexed display. (Used on L outputs.)

COP404LSN-5 inputs have an on-chip depletion load device to V_{CC}.

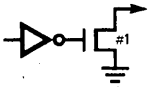
The above input and output configurations share common enhancement-mode and depletion-mode devices. Specifically, all configurations use one or more of six devices (numbered 1–6, respectively). Minimum and maximum current (I_{OUT} and V_{OUT}) curves are given in *Figure 6* for each of these devices to allow the designer to effectively use these I/O configurations in designing a system.

An important point to remember is that even when the L drivers are disabled, the depletion load device will source a small amount of current (see *Figure 6*, device 2); however, when the L-lines are used as inputs, the disabled depletion device can *not* be relied on to source sufficient current to pull an input to a logic "1".



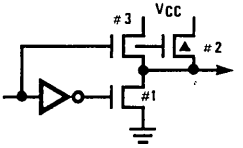
TL/DD/8817-6

a. Standard Output



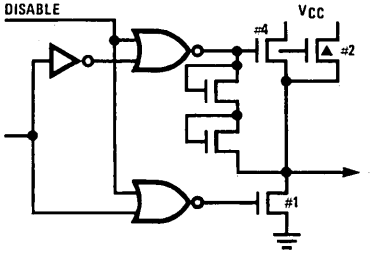
TL/DD/8817-7

b. Open-Drain Output



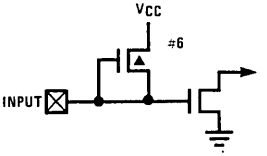
TL/DD/8817-8

c. Push-Pull Output



TL/DD/8817-9

d. L Output (LED)



TL/DD/8817-10

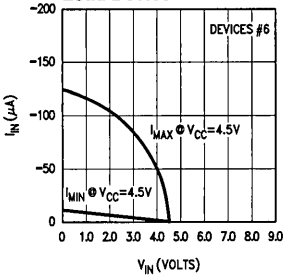
e. Input with Load

(▲ is Depletion Device)

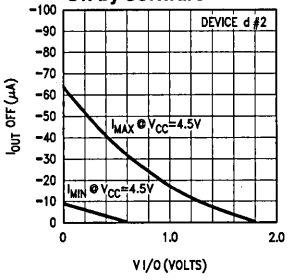
FIGURE 5. Output Configurations

Typical Performance Characteristics

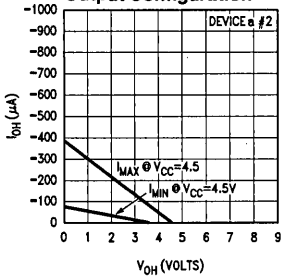
Current for Inputs with Load Device



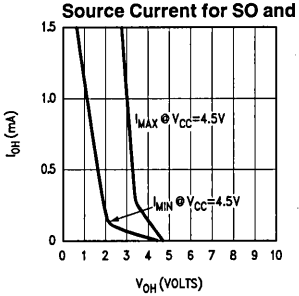
Input Current for L_0 through L_7 when Output Programmed Off by Software



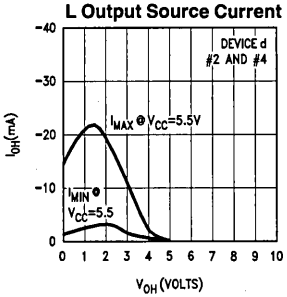
Source Current for Standard Output Configuration



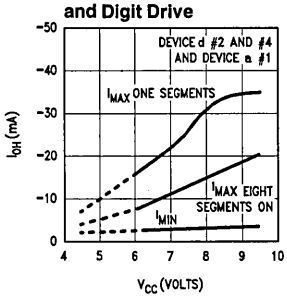
Source Current for SO and SK



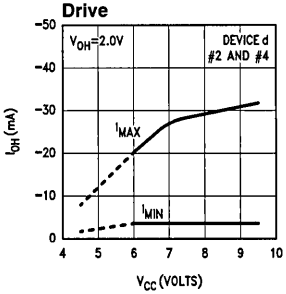
L Output Source Current



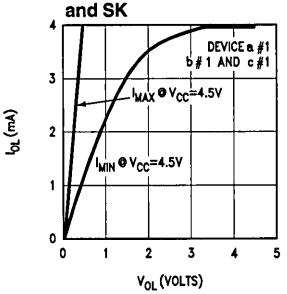
LED Output Direct Segment and Digit Drive



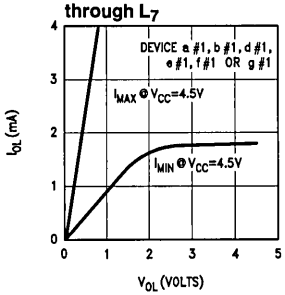
LED Output Drive Segment Drive



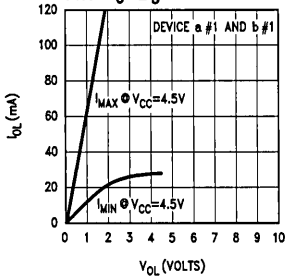
Output Sink Current for SO and SK



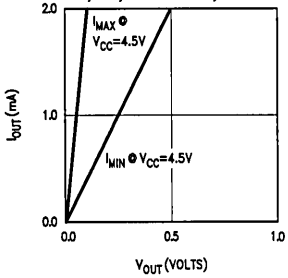
Output Sink Current for L_0 through L_7



Output Sink Current G_0–G_3 and D_0–D_3



Output Sink Current IP0–IP7, P8, P9, SKIP/P10, AD/$\overline{\text{DATA}}$



TL/DD/8817–11

FIGURE 6. COP404LSN-5 I/O Characteristics

COP404LSN-5 Instruction Set

Table I is a symbol table providing internal architecture, instruction operand and operational symbols used in the instruction set table.

Table II provides the mnemonic, operand, machine code, data flow, skip conditions, and description associated with each instruction in the COP404LSN-5 instruction set.

TABLE I. COP404LSN-5 Instruction Set Table Symbols

Symbol	Definition
INTERNAL ARCHITECTURE SYMBOLS	
A	4-bit Accumulator
B	10-bit RAM Address Register
Br	Upper 3 bits of B (register address)
Bd	Lower 4 bits of B (digit address)
C	1-bit Carry Register
D	4-bit Data Output Port
EN	4-bit Enable Register
G	4-bit Register to latch data for G I/O Port
IL	Two 1-bit latches associated with the IN_3 or IN_0 inputs
IN	4-bit Input Port
IP	8-bit bidirectional ROM address and Data Port
L	8-bit TRI-STATE I/O Port
M	4-bit contents of RAM Memory pointed to by B Register
P	3-bit ROM Address Register Port
PC	11-bit ROM Address Register (program counter)
Q	8-bit Register to latch data for L I/O Port
SA	11-bit Subroutine Save Register A
SB	11-bit Subroutine Save Register B
SC	11-bit Subroutine Save Register C
SIO	4-bit Shift Register and Counter
SK	Logic-Controlled Clock Output

Symbol	Definition
INSTRUCTION OPERAND SYMBOLS	
d	4-bit Operand Field, 0–15 binary (RAM Digit Select)
r	3-bit Operand Field, 0–7 binary (RAM Register Select)
a	11-bit Operand Field, 0–2047 binary (ROM Address)
y	4-bit Operand Field, 0–15 binary (Immediate Data)
RAM(s)	Contents of RAM location addressed by s
ROM(t)	Contents of ROM location addressed by t
OPERATIONAL SYMBOLS	
+	Plus
−	Minus
→	Replaces
↔	Is exchanged with
=	Is equal to
$\overline{A}$	The one's complement of A
⊕	Exclusive-OR
:	Range of values

TABLE II. COP404LSN-5 Instruction Set

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
ARITHMETIC INSTRUCTIONS						
ASC		30	|0011|0000|	A + C + RAM(B) → A Carry → C	Carry	Add with Carry, Skip on Carry
ADD		31	|0011|0001|	A + RAM(B) → A	None	Add RAM to A
ADT		4A	|0100|1010|	A + 10_{10} → A	None	Add Ten to A
AISC	y	5–	|0101| y |	A + y → A	Carry	Add Immediate, Skip on Carry (y ≠ 0)
CASC		10	|0001|0000|	$\overline{A}$ + RAM(B) + C → A Carry → C	Carry	Complement and Add with Carry, Skip on Carry
CLRA		00	|0000|0000|	0 → A	None	Clear A
COMP		40	|0100|0000|	$\overline{A}$ → A	None	One's complement of A to A
NOP		44	|0100|0100|	None	None	No Operation
RC		32	|0011|0010|	"0" → C	None	Reset C
SC		22	|0010|0010|	"1" → C	None	Set C
XOR		02	|0000|0010|	A ⊕ RAM(B) → A	None	Exclusive-OR RAM with A

TABLE II. COP404LSN-5 Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
TRANSFER OF CONTROL INSTRUCTIONS						
JID		FF	1111 1111	ROM ($PC_{10:8}$, A,M) → $PC_{7:0}$	None	Jump Indirect (Note 2)
JMP	a	6–	0110 0 $a_{10:8}$	a → PC	None	Jump
		--	$a_{7:0}$			
JP	a	--	1 $a_{6:0}$ (pages 2,3 only)	a → $PC_{6:0}$	None	Jump within Page (Note 4)
			or			
		--	11 $a_{5:0}$ (all other pages)	a → $PC_{5:0}$		
JSRP	a	--	10 $a_{5:0}$	PC + 1 → SA → SB → SC 00010 → $PC_{10:6}$ a → $PC_{5:0}$	None	Jump to Subroutine Page (Note 5)
JSR	a	6–	0110 1 $a_{10:8}$	PC + 1 → SA → SB → SC a → PC	None	Jump to Subroutine
		--	$a_{7:0}$			
RET		48	0100 1000	SC → SB → SA → PC	None	Return from Subroutine
RETSK		49	0100 1001	SC → SB → SA → PC	Always Skip on Return	Return from Subroutine then Skip
MEMORY REFERENCE INSTRUCTIONS						
CAMQ		33	0011 0011	A → $Q_{7:4}$ RAM(B) → $Q_{3:0}$	None	Copy A, RAM to Q
		3C	0011 1100			
CQMA		33	0011 0011	$Q_{7:4}$ → RAM(B) $Q_{3:0}$ → A	None	Copy Q to RAM, A
		2C	0010 1100			
LD	r	–5	00 r 0101 (r = 0:3)	RAM(B) → A Br ⊕ r → Br	None	Load RAM into A, Exclusive-OR Br with r
LDD	r,d	23	0010 0011	RAM(r,d) → A	None	Load A with RAM pointed to directly by r,d
		--	0 r d			
LQID		BF	1011 1111	ROM($PC_{10:8}$,A,M) → Q SB → SC	None	Load Q Indirect (Note 3)
RMB	0	4C	0100 1100	0 → $RAM(B)_0$	None	Reset RAM Bit
	1	45	0100 0101	0 → $RAM(B)_1$		
	2	42	0100 0010	0 → $RAM(B)_2$		
	3	43	0100 0011	0 → $RAM(B)_3$		
SMB	0	4D	0100 1101	1 → $RAM(B)_0$	None	Set RAM Bit
	1	47	0100 0111	1 → $RAM(B)_1$		
	2	46	0100 0110	1 → $RAM(B)_2$		
	3	4B	0100 1011	1 → $RAM(B)_3$		
STII	y	7–	0111 y	y → RAM(B) Bd + 1 → Bd	None	Store Memory Immediate and Increment Bd
X	r	–6	00 r 0110 (r = 0:3)	RAM(B) ⟷ A Br ⊕ r → Br	None	Exchange RAM with A, Exclusive-OR Br with r
XAD	r,d	23	0010 0011	RAM(r,d) ⟷ A	None	Exchange A with RAM pointed to directly by (r,d)
		--	1 r d			
XDS	r	–7	00 r 0111 (r = 0:3)	RAM(B) ⟷ A Bd – 1 → Bd Br ⊕ r → Br	Bd decrements past 0	Exchange RAM with A and Decrement Bd, Exclusive-OR Br with r

TABLE II. COP404LSN-5 Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
MEMORY REFERENCE INSTRUCTIONS (Continued)						
XIS	r	–4	|00|r|0100| (r = 0:3)	RAM(B) ⟷ A Bd + 1 → Bd Br ⊕ r → Br	Bd increments past 15	Exchange RAM with A and Increment Bd, Exclusive-OR Br with r
REGISTER REFERENCE INSTRUCTIONS						
CAB		50	|0101|0000|	A → Bd	None	Copy A to Bd
CBA		4E	|0100|1110|	Bd → A	None	Copy Bd to A
LBI	r,d	– –	|00|r|(d−1)| (r = 0:3; d = 0, 9:15)	r,d → B	Skip until not a LBI	Load B Immediate with r,d (Note 6)
			or			
		33	|0011|0011|			
		– –	|1| r | d | (any r, any d)			
LEI	y	33	|0011|0011|	y → EN	None	Load EN Immediate (Note 7)
		6–	|0110| y |			
XABR		12	|0001|0010|	A ⟷ Br (0 → A_3)	None	Exchange A with Br
TEST INSTRUCTIONS						
SKC		20	|0010|0000|		C = "1"	Skip if C is True
SKE		21	|0010|0001|		A = RAM(B)	Skip if A Equals RAM
SKGZ		33	|0011|0011|		$G_{3:0}$ = 0	Skip if G is Zero (all 4 bits)
		21	|0010|0001|			
SKGBZ		33	|0011|0011|	1st byte		Skip if G Bit is Zero
	0	01	|0000|0001|	2nd byte	G_0 = 0	
	1	11	|0001|0001|		G_1 = 0	
	2	03	|0000|0011|		G_2 = 0	
	3	13	|0001|0011|		G_3 = 0	
SKMBZ	0	01	|0000|0001|		$RAM(B)_0$ = 0	Skip if RAM Bit is Zero
	1	11	|0001|0001|		$RAM(B)_1$ = 0	
	2	03	|0000|0011|		$RAM(B)_2$ = 0	
	3	13	|0001|0011|		$RAM(B)_3$ = 0	
SKT		41	|0100|0001|		A time-base counter carry has occurred since last test	Skip on Timer (Note 2)
INPUT/OUTPUT INSTRUCTIONS						
ING		33	|0011|0011|	G → A	None	Input G Ports to A
		2A	|0010|1010|			
ININ		33	|0011|0011|	IN → A	None	Input IN Inputs to A
		28	|0010|1000|			
INIL		33	|0011|0011|	IL_3, CKO, "0", IL_0 → A	None	Input IL Latches to A (Note 2)
		29	|0010|1001|			
INL		33	|0011|0011|	$L_{7:4}$ → RAM(B) $L_{3:0}$ → A	None	Input L Ports to RAM, A
		2E	|0010|1110|			
OBD		33	|0011|0011|	Bd → D	None	Output Bd to D Outputs
		3E	|0011|1110|			

1

TABLE II. COP404LSN-5 Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Machine Language Code (Binary)	Data Flow	Skip Conditions	Description
INPUT/OUTPUT INSTRUCTIONS (Continued)						
OGI	y	33 5–	0011 | 0011 0101 | y	y → G	None	Output to G Ports Immediate
OMG		33 3A	0011 | 0011 0011 | 1010	RAM(B) → G	None	Output RAM to G Ports
XAS		4F	0100 | 1111	A ⟷ SIO, C → SKL	None	Exchange A with SIO (Note 2)

Note 1: All subscripts for alphabetical symbols indicate bit numbers unless explicitly defined (e.g., Br and Bd are explicitly defined). Bits are numbered 0 to N where 0 signifies the least significant bit (low-order, right-most bit). For example, A_3 indicates the most significant (left-most) bit of the 4-bit A register.

Note 2: For additional information on the operation of the XAS, JID, LQID, INIL, and SKT instructions, see below.

Note 3: The JP instruction allows a jump, while in subroutine pages 2 or 3, to any ROM location within the two-page boundary of pages 2 or 3. The JP instruction, otherwise, permits a jump to a ROM location within the current 64-word page. JP may not jump to the last word of a page.

Note 4: A JSRP transfers program control to subroutine page 2 (0010 is loaded into the upper 4 bits of P). A JSRP may not be used when in pages 2 or 3. JSRP may not jump to the last word in page 2.

Note 5: LBI is a single-byte instruction if d = 0, 9, 10, 11, 12, 13, 14, or 15. The machine code for the lower 4 bits equals the binary value of the "d" data *minus 1*, e.g., to load the lower four bits of B (Bd) with the value 9 (1001_2), the lower 4 bits of the LBI instruction equal 8 (1000_2). To load 0, the lower 4 bits of the LBI instruction should equal 15 (1111_2).

Note 6: Machine code for operand field y for LEI instruction should equal the binary value to be latched into EN, where a "1" or "0" in each bit of EN corresponds to the selection or deselection of a particular function associated with each bit. (See Functional Description, EN Register.)

Description of Selection Instructions

The following information is provided to assist the user in understanding the operation of several unique instructions and to provide notes useful to programmers in writing COP404LSN-5 programs.

XAS INSTRUCTIONS

XAS (Exchange A with SIO) exchanges the 4-bit contents of the accumulator with the 4-bit contents of the SIO register. The contents of SIO will contain serial-in/serial-out shift register or binary counter data, depending on the value of the EN register. An XAS instruction will also affect the SK output. (See Functional Description, EN Register.) If SIO is selected as a shift register, an XAS instruction must be performed once every 4 instruction cycles to effect a continuous data stream.

JID INSTRUCTION

JID (Jump Indirect) is an indirect addressing instruction, transferring program control to a new ROM location pointed to indirectly by A and M. It loads the lower 8 bits of the ROM address register PC with the *contents* of ROM addressed by the 11-bit word, $PC_{10:8}$, A, M. PC_{10}, PC_9 and PC_8 are not affected by this instruction.

Note: JID requires 2 instruction cycles to execute.

INIL INSTRUCTION

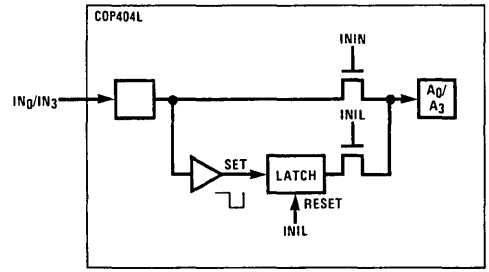
INIL (Input IL Latches to A) inputs 2 latches, IL_3 and IL_0 (see *Figure 7*) and CKO into A. The IL_3 and IL_0 latches are set if a low-going pulse ("1" to "0") has occurred on the IN_3 and IN_0 inputs since the last INIL instruction, provided the input pulse stays low for at least two instruction times. Execution of an INIL inputs IL_3 and IL_0 into A3 and A0 respectively, and resets these latches to allow them to respond to subsequent low-going pulses on the IN_3 and IN_0 lines. INIL will input "1" into A2 on the COP404LSN-5. A "0" is always placed in A1 upon the execution of an INIL. The general purpose inputs IN_3–IN_0 are input to A upon execution of an ININ instruction. (See Table II, ININ instruction.) INIL is useful in recognizing pulses of short duration or pulses which occur too often to be read conveniently by an ININ instruction.

Note: IL latches are not cleared on reset.

LQID INSTRUCTION

LQID (Load Q Indirect) loads the 8-bit Q register with the contents of ROM pointed to by the 11-bit word PC_{10}, PC_9, PC_8, A, M. LQID can be used for table lookup or code conversion such as BCD to seven-segment. The LQID instruction "pushes" the stack (PC + 1 → SA → SB → SC) and replaces the least significant 8 bits of PC as follows: A → $PC_{7:4}$, RAM(B) → $PC_{3:0}$, leaving PC_{10}, PC_9 and PC_8 unchanged. The ROM data pointed to by the new address is fetched and loaded into the Q latches. Next, the stack is "popped" (SC → SB → SA → PC), restoring the saved value of PC to continue sequential program execution. Since LQID pushes SB → SC, the previous contents of SC are lost. Also, when LQID pops the stack, the previously pushed contents of SB are left in SC. The net result is that the contents of SB are placed in SC (SB → SC).

Note: LQID takes two instruction cycle times to execute.



TL/DD/8817–12

FIGURE 7. INIL Hardware Implementation

Description of Selected Instructions (Continued)

SKT INSTRUCTION

The SKT (Skip On Timer) instruction tests the state of an internal 10-bit time-base counter. This counter divides the instruction cycle clock frequency by 1024 and provides a latched indication of counter overflow. The SKT instruction tests this latch, executing the next program instruction if the latch is not set. If the latch has been set since the previous test, the next program instruction is skipped and the latch is reset. The features associated with this instruction, therefore, allow the COP404LSN-5 to generate its own time-base for real-time processing rather than relying on an external input signal.

For example, using a 2.097 MHz oscillator as the time-base to the clock generator, the instruction cycle clock frequency will be 65 kHz (crystal frequency ÷ 32) and the binary counter output pulse frequency will be 64 Hz. For time-of-day or similar real-time processing, the SKT instruction can call a routine which increments a "seconds" counter every 64 ticks.

INSTRUCTION SET NOTES

a. The first word of a COP404LSN-5 program (ROM address 0) must be a CLRA (Clear A) instruction.

b. Although skipped instructions are not executed, one instruction cycle time is devoted to skipping each byte of the skipped instruction. Thus all program paths except JID and LQID take the same number of cycle times whether instructions are skipped or executed. JID and LQID instructions take 2 cycles if executed and 1 cycle if skipped.

c. The ROM is organized into 32 pages of 64 words each. The Program Counter is an 11-bit binary counter, and will count through page boundaries. If a JP, JSRP, JID or LQID instruction is located in the last word of a page, the instruction operates as if it were in the next page. For example: a JP located in the last word of a page will jump to a location in the next page. Also, a LQID or JID located in the last word of page 3, 7, 11, 15, 19, 23 or 27 will access data in the next group of four pages.

Typical Applications

PROM-BASED SYSTEM

The COP404LSN-5 may be used to exactly emulate the COP444L. *Figure 8* shows the interconnect to implement a COP444L hardware emulation. This connection uses a MM2716 EPROM as external memory. Other memory can be used such as bipolar PROM or RAM.

Pins IP7–IP0 are bidirectional inputs and outputs. When the AD/$\overline{\text{DATA}}$ clocking output turns on, the EPROM drivers are disabled and IP7–IP0 output addresses. The 8-bit latch (MM74LS373) latches the addresses to drive the memory.

When AD/$\overline{\text{DATA}}$ turns off, the EPROM is enabled and the IP7–IP0 pins will input the memory data. P8, P9 and SKIP/P10 output the most significant address bits to the memory. (SKIP output may be used for program debug if needed.)

The other 28 pins of the COP404LSN-5 may be configured exactly the same as a COP444L. The COP404LSN-5 V_{CC} can vary from 4.5V to 5.5V. However, 5V is used for the memory.

For In-Circuit emulation, see also COP444LP.

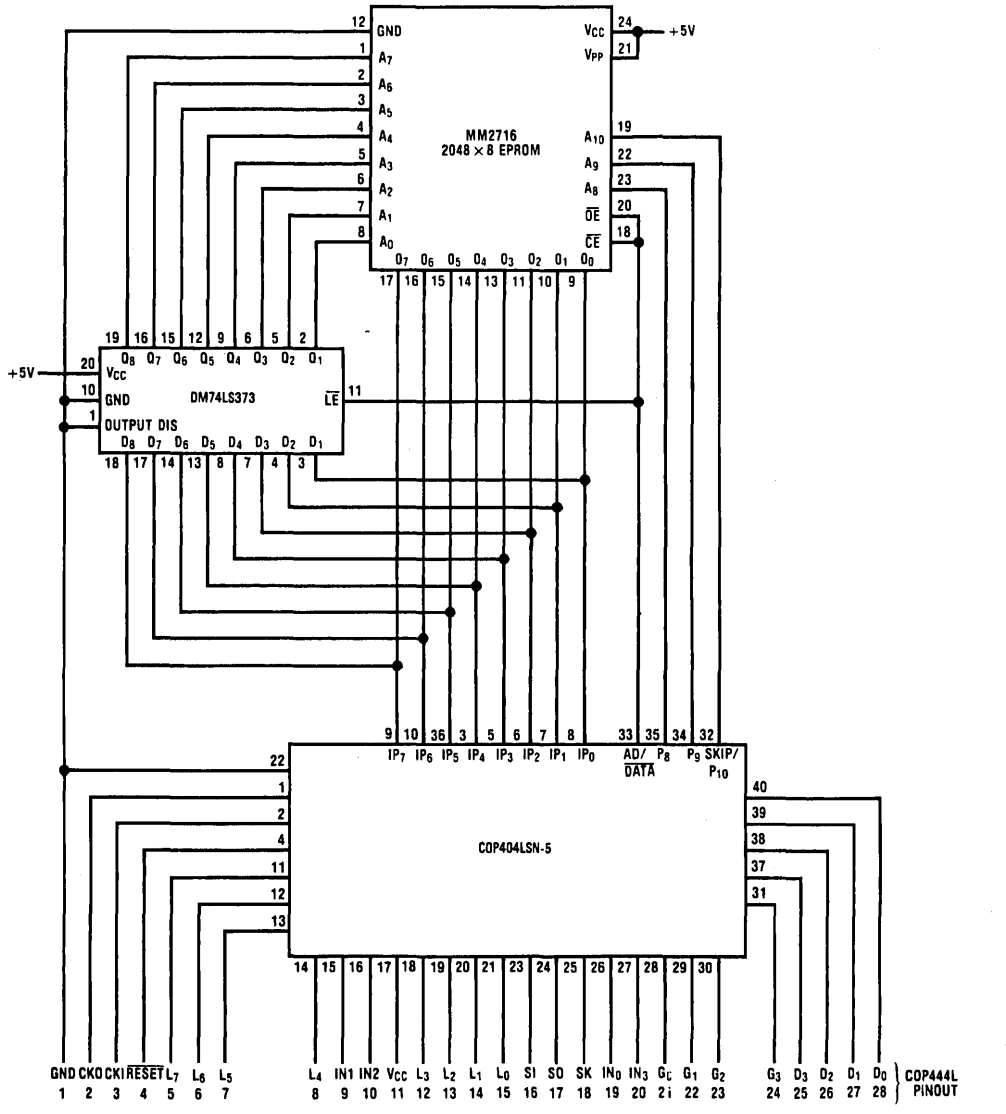
COP404LSN-5 Mask Options

The following COP444L options have been implemented on the COP404LSN-5.

Option Value	Comment
Option 1 = 0	Ground, no option available
Option 2 = 0	CKO is clock generator output to crystal/resonator
Option 3 = 0	CKI is oscillator input (divide by 32)
Option 4 = 0	$\overline{\text{RESET}}$ pin has load device to V_{CC}
Option 5 = 2	L_7 }
Option 6 = 2	L_6 } have LED direct-drive
Option 7 = 2	L_5 } output
Option 8 = 2	L_4 }
Option 9 = 0	IN1 has load device to V_{CC}
Option 10 = 0	IN2 has load device to V_{CC}
Option 11 = 1	V_{CC} 4.5V to 5.5V operation
Option 12 = 2	L_3 }
Option 13 = 2	L_2 } have LED direct-drive
Option 14 = 2	L_1 } output
Option 15 = 2	L_0 }
Option 16 = 0	SI has load to V_{CC}
Option 17 = 2	SO has push-pull output

Option Value	Comment
Option 18 = 2	SK has push-pull output
Option 19 = 0	IN0 has load device to V_{CC}
Option 20 = 0	IN3 has load device to V_{CC}
Option 21 = 0	G_0 }
Option 22 = 0	G_1 } have high current
Option 23 = 0	G_2 } standard output
Option 24 = 0	G_3 }
Option 25 = 0	D_3 }
Option 26 = 0	D_2 } have high current
Option 27 = 0	D_1 } standard output
Option 28 = 0	D_0 }
Option 29 = 1	L }
Option 30 = 1	IN } have higher voltage
Option 31 = 1	G } input levels
Option 32 = 0	SI has standard input level
Option 33 = 0	$\overline{\text{RESET}}$ has Schmitt trigger input
Option 34 = 0	CKO has standard input levels
Option 35 = N/A	40-pin package

Typical Applications (Continued)



TL/DD/8617–13

FIGURE 8. COP404LSN-5 System Diagram

Section 2

COP800 Family

Section 2 Contents

COP800C

The COP800 Family of Products combines the powerful COPS Instruction Set philosophy with a memory mapped core architecture to offer the lowest cost M²CMOS microcontroller available today.

These chips are complete microcomputers on a single chip. All system timing, internal logic, ROM, RAM, and I/O are provided on the chip to produce a cost effective system solution.

The M²CMOS process results in very low current drain and enables the user to select the optimum speed/power product for his system.

These devices are available in various DIP and SO package configurations.

Features

- Fully static M²CMOS
- 2.5V to 6.0V operating voltage
- 1 μs instruction cycle time
- Low current drain
- Ultra low current static halt mode
- Single supply operation
- MICROWIRE PLUS™ serial I/O
- 20 through 44 pin packages
- Software selectable I/O options
- Fully supported by National's PC biased Mole™ Development System
- E²PROM program space
- E²PROM data space
- Hi-I/O and LCD drive

COP800 Roadmap

Part #	ROM	RAM	EEPROM (ROM)	EEPROM (RAM)	Supply Voltage	INTERRUPT	Time Count	I/O	Pins	Features
COP820C	1k	64 x 8			2.5V–6.0V	YES	YES	24	28	STANDARD PART
COP821C	1k	64 x 8			2.5V–6.0V	YES	YES	20	24	STANDARD PART
COP822C	1k	64 x 8			2.5V–6.0V	YES	YES	16	20	STANDARD PART
COP8620C	1k	64 x 8		64 x 8	2.7V–6.0V	YES	YES	24	28	STANDARD PART
COP8621C	1k	64 x 8		64 x 8	2.7V–6.0V	YES	YES	20	24	STANDARD PART
COP8622C	1k	64 x 8		64 x 8	2.7V–6.0V	YES	YES	16	20	STANDARD PART
COP8720C		64 x 8	1k	64 x 8	2.7V–6.0V	YES	YES	24	28	STANDARD PART
COP8721C		64 x 8	1k	64 x 8	2.7V–6.0V	YES	YES	20	24	STANDARD PART
COP8722C		64 x 8	1k	64 x 8	2.7V–6.0V	YES	YES	16	20	STANDARD PART
COP840C	2k	128 x 8			2.5V–6.0V	YES	YES	24	28	STANDARD PART
COP8640C	2k	128 x 8		128 x 8	2.7V–6.0V	YES	YES	24	28	STANDARD PART
COP8740C		128 x 8	2k	128 x 8	2.7V–6.0V	YES	YES	24	28	STANDARD PART
COP848C	2k	128 x 8			2.5V–6.0V	YES	YES	36/40	40/44	24 x 4 LCD or I/O
COP888C	4k	128 x 8			2.5V–6.0V	YES	YES	36/40	40/44	UART 24 x 4 LCD or I/O
COP8788		128 x 8	4k	128 x 8	2.7V–6.0V	YES	YES	36/40	40/44	UART 24 x 4 LCD or I/O

The COP800C is the low cost CMOS system solution.

National Semiconductor Corporation

PRELIMINARY

COP820C/COP821C/COP822C Single Chip microCMOS Microcontrollers

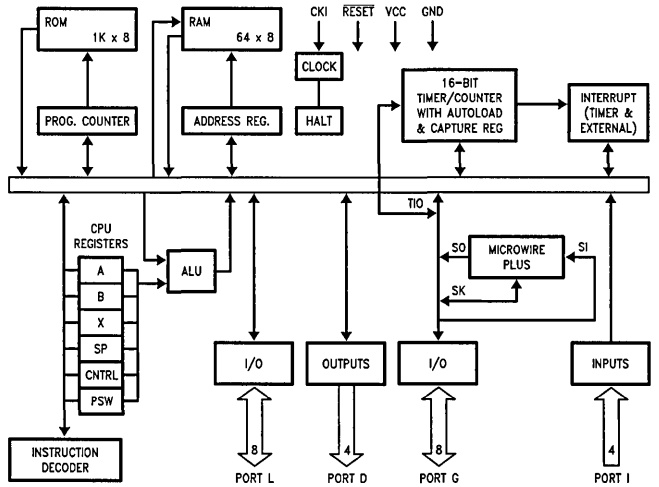
General Description

The COP820C is a member of the COPS™ microcontroller family. It is a fully static part, fabricated using double-metal silicon gate microCMOS technology. This low cost microcontroller is a complete microcomputer containing all system timing, interrupt logic, ROM, RAM, and I/O necessary to implement dedicated control functions in a variety of applications. Features include an 8-bit memory mapped architecture, MICROWIRE serial I/O, a 16-bit timer/counter with capture register and a multi-sourced interrupt. Each I/O pin has software selectable options to adapt the COP820C to the specific application. The part operates over a voltage range of 2.5 to 6.0V. High throughput is achieved with an efficient, regular instruction set operating at a 1 microsecond per instruction rate. The part may be operated in the ROMless mode to provide for accurate emulation and for applications requiring external program memory.

Features

- Low Cost 8-bit microcontroller
- Fully static CMOS
- 1 μs instruction time (20 MHz clock)
- Low current drain (3.5 mA at 3 μs instruction rate)
  Extra-low current static HALT mode (Typically < 1 μA)
- Single supply operation: 2.5 to 6.0V
- 1024 bytes on-chip ROM
  — Expandable to 32k bytes in ROMless mode
- 64 bytes on-chip RAM
- 16-bit read/write timer operates in a variety of modes
  — Timer with 16-bit auto reload register
  — 16-bit external event counter
  — Timer with 16-bit capture register (selectable edge)
- Multi-source interrupt
  — Reset master clear
  — External interrupt with selectable edge
  — Timer interrupt or capture interrupt
  — Software/"watch-dog" interrupt
- 8-bit stack pointer (stack in RAM)
- Powerful instruction set, most instruction single byte
- BCD arithmetic instructions
- MICROWIRE PLUS™ serial I/O
- 28 pin package (optionally 24 or 20 pin package)
- 24 input/output pins
- Software selectable I/O options (TRI-STATE®, push-pull, weak pull-up)
- Schmitt trigger inputs on Port G
- Extended temperature ranges: −40°C to +85°C (−55°C to +125°C to be available)
- ROMless mode for accurate emulation and external program capability
- Form, fit and function EEPROM emulation device (COP8720C)
- Fully supported by National's MOLE™ development system

Block Diagram



TL/DD/9103-1

FIGURE 1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage (V_{CC})	7V
Voltage at any Pin	−0.3V to V_{CC} + 0.3V
ESD Susceptibility (Note 4)	2000V
Total Current into V_{CC} Pin (Source)	50 mA
Total Current out of GND Pin (Sink)	60 mA
Storage Temperature Range	−65°C to +150°C

Note: Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.

DC Electrical Characteristics −40°C < T_A < +85°C unless otherwise specified

Parameter	Condition	Min	Typ	Max	Units
Operating Voltage		2.5		6.0	V
Power Supply Ripple (Note 1)	Peak to Peak			0.1 V_{CC}	V
Supply Current (see page 17)					
High Speed Mode, CKI = 20 MHz	V_{CC} = 6V, tc = 1 μs			15	mA
Normal Mode, CKI = 5 MHz	V_{CC} = 6V, tc = 2 μs			5	mA
Normal Mode, CKI = 2 MHz	V_{CC} = 2.5V, tc = 5 μs			1	mA
(Note 2)					
HALT Current	V_{CC} = 6V, CKI = 0 MHz		<1		μA
(Note 3)					
Input Levels					
RESET, CKI					
Logic High		0.9 V_{CC}			V
Logic Low				0.1 V_{CC}	V
All Other Inputs					
Logic High		0.7 V_{CC}			V
Logic Low				0.2 V_{CC}	V
Hi-Z Input Leakage	V_{CC} = 6.0V, V_{IN} = 0V	−2		+2	μA
Input Pullup Current	V_{CC} = 6.0V, V_{IN} = 0V	TBD		TBD	μA
G Port Input Hysteresis			0.05 V_{CC}		V
Output Current Levels					
D Outputs					
Source	V_{CC} = 4.5V, V_{OH} = 3.8V	0.4			mA
	V_{CC} = 2.5V, V_{OH} = 1.8V	TBD			mA
Sink	V_{CC} = 4.5V, V_{OL} = 1.0V	10			mA
	V_{CC} = 2.5V, V_{OL} = 0.4V	TBD			mA
All Others					
Source (Weak Pull-Up)	V_{CC} = 4.5V, V_{OH} = 3.2V	10		100	μA
	V_{CC} = 2.5V, V_{OH} = 1.8V	TBD		TBD	μA
Source (Push-Pull Mode)	V_{CC} = 4.5V, V_{OH} = 3.8V	0.4			mA
	V_{CC} = 2.5V, V_{OH} = 1.8V	TBD			
Sink (Push-Pull Mode)	V_{CC} = 4.5V, V_{OL} = 0.4V	1.6			mA
	V_{CC} = 2.5V, V_{OL} = 0.4V	TBD			
TRI-STATE Leakage		−2.5		+2.5	μA
Allowable Sink/Source Current Per Pin					
D Outputs (Sink)				15	mA
All Others				3	mA
Maximum Input Current Without Latchup			TBD		mA
RAM Retention Voltage, Vr			TBD		V
Input Capacitance				7	pF
Load Capacitance on D2				1000	pF

Note 1: Rate of voltage change must be less than 0.5V/ms.

Note 2: Supply current is measured after running 2000 cycles with a square wave CKI input, CKO open, inputs at rails and outputs open.

Note 3: The HALT mode will stop CKI from oscillating in the RC and the Crystal configurations. Test conditions: All inputs tied to V_{CC}, L and G ports TRI-STATED and tied to ground, all outputs low and tied to ground.

Note 4: Human body mode, 100 pF through 1500Ω.

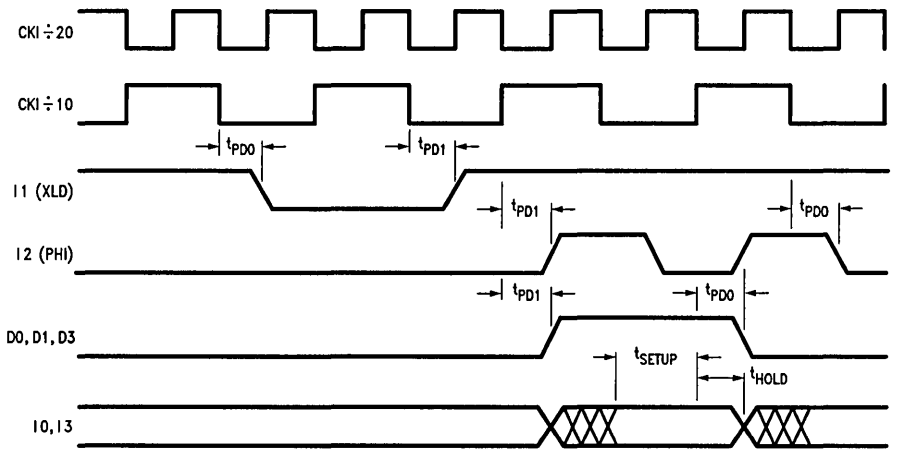
AC Electrical Characteristics $-40°C < T_A < +85°C$ unless otherwise specified

Parameter	Condition	Min	Typ	Max	Units
Instruction Cycle Time (tc)					
High Speed Mode	$V_{CC} \geq 4.5V$	1		DC	μs
(Div-by 20)	$2.5V < V_{CC} < 4.5V$	2.5		DC	μs
Normal Mode	$V_{CC} \geq 4.5V$	2		DC	μs
(Div-by 10)	$2.5V < V_{CC} < 4.5V$	5		DC	μs
R/C Oscillator Mode	$V_{CC} \geq 4.5V$	3		DC	μs
(Div-by 10)					
(See Page 16)	$2.5V < V_{CC} < 4.5V$	7.5		DC	μs
CKI Clock Duty Cycle (Note 5)	fr = Max	33		66	%
Rise Time (Note 5)	fr = 20 MHz Ext Clock			12	ns
Fall Time (Note 5)	fr = 20 MHz Ext Clock			8	ns
Inputs					
t_{SETUP}	$V_{CC} \geq 4.5V$	200			ns
	$2.5V < V_{CC} < 4.5V$	500			ns
t_{HOLD}	$V_{CC} \geq 4.5V$	60			ns
	$2.5V < V_{CC} < 4.5V$	150			ns
Output Propagation Delay t_{PD1}, t_{PD0}	R_L = 2.2k, C_L = 100 pF				
SO, SK	$V_{CC} \geq 4.5V$			0.7	μs
	$2.5V < V_{CC} < 4.5V$			1.75	μs
All Others	$V_{CC} \geq 4.5V$			1	μs
	$2.5V < V_{CC} < 4.5V$			2.5	μs
Input Pulse Width					
Interrupt Input High Time		TBD			ns
Interrupt Input Low Time		TBD			
Timer Input High Time		TBD			ns
Timer Input Low Time		TBD			
Reset Pulse Width		TBD			ns

Note 5: Parameter sampled but not 100% tested.

AC Electrical Characteristics in ROMless Mode $-40°C < T_A < 85°C$ unless otherwise specified

Parameter	Condition	Min	Typ	Max	Units
Instruction Cycle Time (tc)					
High Speed Mode	$V_{CC} \geq 4.5V$		2	DC	μs
(Div-by 20)	$2.5V < V_{CC} < 4.5V$		5	DC	μs
Normal Mode	$V_{CC} \geq 4.5V$		4	DC	μs
(Div-by 10)	$2.5V < V_{CC} < 4.5V$		10	DC	μs
R/C Oscillator Mode	$V_{CC} \geq 4.5V$		6	DC	μs
(See Page 8)	$2.5V < V_{CC} < 4.5V$		15	DC	μs
CKI Clock Duty Clock	fr = Max	40		60	%
Rise Time	fr = 10 MHz Ext Clock		24		ns
Fall Time	fr = 10 MHz Ext Clock		16		ns
Inputs					
t_{SETUP}	$V_{CC} \geq 4.5V$		400		ns
	$2.5V < V_{CC} < 4.5V$		800		ns
t_{HOLD}	$V_{CC} \geq 4.5V$		120		ns
	$2.5V < V_{CC} < 4.5V$		300		ns
Output Propagation Delay t_{PD1}, t_{PD0}	RL = 2.2k, C_L = 100 pF				
SO, SK	$V_{CC} \geq 4.5V$		1.4		μs
	$2.5V < V_{CC} < 4.5V$		3.5		μs
All Others	$V_{CC} \geq 4.5V$		2		μs
	$2.5V < V_{CC} < 4.5V$		5		μs
Minimum Pulse Width					
Interrupt Input			TBD		ns
Timer Input			TBD		ns
Reset Pulse Width					ns

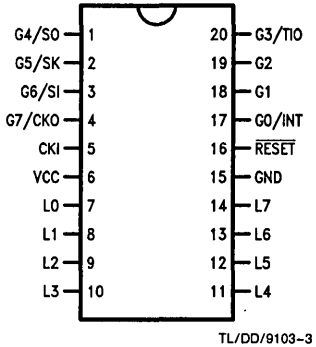


TL/DD/9103-2

FIGURE 2. AC Timing Diagrams in ROMless Mode

Connection Diagrams

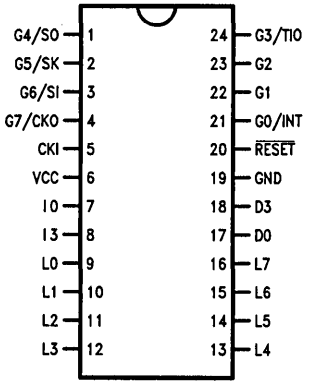
COP822C



TL/DD/9103-3

Order Number COP822C-XXXN
See NS Package Number N20A

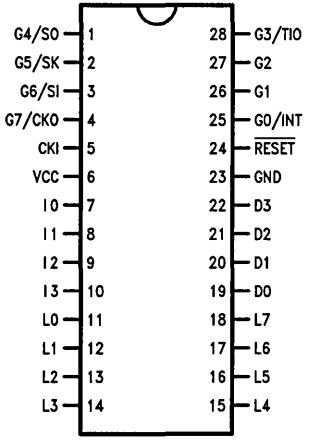
COP821C



TL/DD/9103-4

Order Number COP821C-XXXN
See NS Package Number N24A

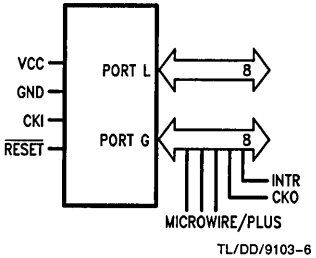
COP820C



TL/DD/9103-5

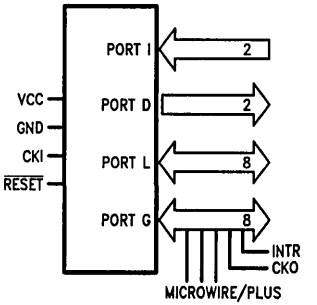
Order Number COP820C-XXXN
See NS Package Number N28B

COP822C



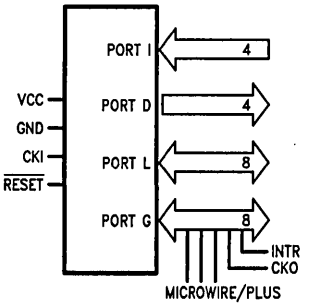
TL/DD/9103-6

COP821C



TL/DD/9103-7

COP820C



TL/DD/9103-8

FIGURE 3

Pin Descriptions

V_{CC} and GND are the power supply pins.

CKI is the clock input. This can come from an external source, a R/C generated oscillator or a crystal (in conjunction with CKO). See Oscillator description.

$\overline{\text{RESET}}$ is the master reset input. See Reset description.

PORT I is a four bit Hi-Z input port.

PORT L is an 8-bit I/O port.

There are two registers associated with each L I/O port: a data register and a configuration register. Therefore, each L I/O bit can be individually configured under software control as shown below:

Port L Config.	Port L Data	Port L Setup
0	0	Hi-Z Input (TRI-STATE)
0	1	Input With Weak Pull-Up
1	0	Push-Pull "0" Output
1	1	Push-Pull "1" Output

Three data memory address locations are allocated for these ports, one for data register, one for configuration register and one for the input pins.

PORT G is an 8-bit port with 6 I/O pins (G0–G5) and 2 input pins (G6, G7). All eight G-pins have Schmitt Triggers on the inputs. The G7 pin functions as an input pin under normal operation and as the continue pin to exit the HALT mode. There are two registers with each I/O port: a data register and a configuration register. Therefore, each I/O bit can be individually configured under software control as shown before.

Port G Config.	Port G Data	Port G Setup
0	0	Hi-Z Input (TRI-STATE)
0	1	Input With Weak Pull-Up
1	0	Push-Pull "0" Output
1	1	Push-Pull "1" Output

Three data memory address locations are allocated for these ports, one for data register, one for configuration register and one for the input pins. Since G6 and G7 are input only pins, any attempt by the user to set them up as outputs by writing a one to the configuration register will be disregarded. Reading the G6 and G7 configuration bits will return zeros. Note that the chip will be placed in the HALT mode by setting the G7 data bit.

Six bits of Port G have alternate features:

G0 INTR (an external interrupt)

G3 TIO (timer/counter input/output)

G4 SO (MICROWIRE serial data output)

G5 SK (MICROWIRE clock I/O)

G6 SI (MICROWIRE serial data input)

G7 CKO crystal oscillator output (selected by mask option) or HALT restart input (general purpose input)

Pins G1 and G2 currently do not have any alternate functions.

PORT D is a four bit output port that is set high when $\overline{\text{RESET}}$ goes low. The user can Wire-OR the D-ports in order to get a higher drive.

The D2 pin is sampled at reset. If it is held low at reset the COP820C enters the ROMless mode of operation.

Functional Description

Figure 1 shows the block diagram of the internal architecture. Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other in implementing the instruction set of the device.

ALU AND CPU REGISTERS

The ALU can do an 8-bit addition, subtraction, logical or shift operation in one cycle time.

There are five CPU registers:

A is the 15-bit Program Counter register

PU is the upper 7 bits of the program counter (PC)

PL is the lower 8 bits of the program counter (PC)

B is the 8-bit address register, can be auto incremented or decremented.

X is the 8-bit alternate address register, can be incremented or decremented.

SP is the 8-bit stack pointer, points to subroutine stack (in RAM).

B, X and SP registers are mapped into the on chip RAM. The B and X registers are used to address the on chip RAM. The SP register is used to address the program counter stack in RAM during subroutine calls and returns.

PROGRAM MEMORY

Program memory for the COP820C consists of 1024 bytes of ROM. These bytes may hold program instructions or constant data. The program memory is addressed by the 15-bit program counter (PC). ROM can be indirectly read by the LAID instruction for table lookup.

DATA MEMORY

The data memory address space includes on chip RAM, I/O and registers. Data memory is addressed directly by the instruction or indirectly by the B, X and SP registers.

The COP820C has 64 bytes of RAM. Sixteen bytes of RAM are mapped as "registers" that can be loaded immediately, decremented or tested. Three specific registers: B, X and SP are mapped into this space, the other bytes are available for general usage.

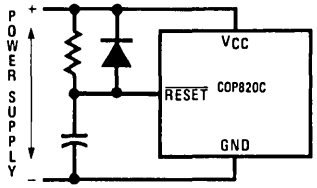
The instruction set of the COP820C permits any bit in memory to be set, reset or tested. All I/O and registers on the COP820C (except the A & PC) are memory mapped; therefore, I/O bits and register bits can be directly and individually set, reset and tested.

RESET

The $\overline{\text{RESET}}$ input when pulled low initializes the microcontroller. Initialization will occur whenever the $\overline{\text{RESET}}$ input is pulled low. Upon initialization, the ports L and G are placed in the TRI-STATE mode and the Port D is set high. The PC, PSW and CNTRL registers are cleared. The data and configuration registers for Ports L & G are cleared.

The external RC network shown in Figure 4 should be used to ensure that the $\overline{\text{RESET}}$ pin is held low until the power supply to the chip stabilizes. It is recommended that the components of the RC network be selected to provide a $\overline{\text{RESET}}$ delay of at least five times the power supply rise time or the minimum $\overline{\text{RESET}}$ pulse width, whichever is greater.

Functional Description (Continued)



TL/DD/9103-9

FIGURE 4. Recommended Reset Circuit

OSCILLATOR CIRCUITS

Figure 5 shows the three clock oscillator configurations available for the COP820C.

A. CRYSTAL OSCILLATOR

The COP820C can be driven by a crystal clock. The crystal network is connected between the pins CKI and CKO.

Table I shows the component values required for various standard crystal values.

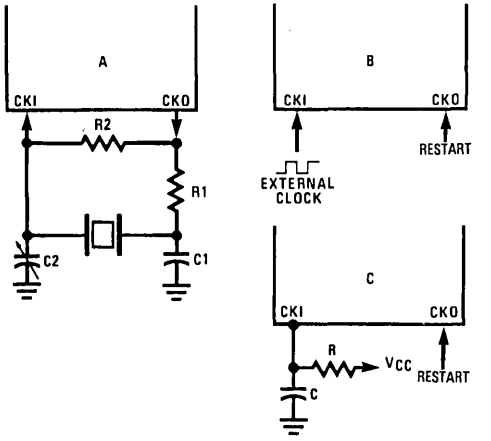
B. EXTERNAL OSCILLATOR

CKI can be driven by an external clock signal. CKO is available as a general purpose input and/or HALT restart control.

C. R/C OSCILLATOR

CKI is configured as a single pin RC controlled Schmitt trigger oscillator. CKO is available as a general purpose input and/or HALT restart control.

Table II shows the variation in the oscillator frequencies as functions of the component (R and C) values.



TL/DD/9103-10

FIGURE 5. Crystal and R-C Connection Diagrams

MASK OPTIONS

The COP820C can be driven by clock inputs between DC and 20 MHz. For low input clock frequencies (< 5 MHz) the instruction cycle frequency can be selected to be the input clock frequency divided by 10. This mode is known as the Normal Mode.

For oscillator frequencies that are greater than 5 MHz the chip must run with a divide by 20. This is known as the High Speed mode.

TABLE I. Crystal Oscillator Configuration, $T_A = 25°C$

R1 (kΩ)	R2 (kΩ)	C1 (pF)	C2 (pF)	CKI Freq (MHz)	Conditions
TBD	TBD	TBD	TBD	20	$V_{CC} = 5V$
TBD	TBD	TBD	TBD	10	$V_{CC} = 5V$
TBD	TBD	TBD	TBD	4	$V_{CC} = 3V$
TBD	TBD	TBD	TBD	0.455	$V_{CC} = 3V$

TABLE II. RC Oscillator Configuration, $T_A = 25°C$

R (kΩ)	C (pF)	CKI Freq. (MHz)	Instr. Cycle (μs)	Conditions
TBD	82	3.3 to 1.1	3 to 9	$V_{CC} = 5V$
TBD	100	1 to 0.5	10 to 20	$V_{CC} = 5V$
TBD	100	1.33 to 0.57	7.5 to 17.5	$V_{CC} = 3V$

Functional Description (Continued)

The COP820C microcontroller provides five mask options for configuring the clock input. The CKI and CKO pins are automatically configured upon selecting a particular option.

— High Speed Crystal (CKI/20) CKO for crystal configuration
— Normal Mode Crystal (CKI/10) CKO for crystal configuration
— High Speed External (CKI/20) CKO available as G7 input
— Normal Mode External (CKI/10) CKO available as G7 input
— R/C
(CKI/10) CKO available as G7 input

Where, G7 can be used either as a general purpose input or as a control input to continue from the HALT mode.

CURRENT DRAIN

The total current drain of the chip depends on:

1) Oscillator operating mode—I1
2) Internal switching current—I2
3) Internal leakage current—I3
4) Output source current—I4
5) DC current caused by external input not at V_{CC} or GND—I5

Thus the total current drain, It is given as

$$It = I1 + I2 + I3 + I4 + I5$$

To reduce the total current drain, each of the above components must be minimum.

The chip will draw the least current when in the normal mode. The high speed mode will draw additional current. The R/C mode will draw the most. Operating with a crystal network will draw more current than an external square-wave. Switching current, governed by the equation below, can be reduced by lowering voltage and frequency. Leakage current can be reduced by lowering voltage and temperature. The other two items can be reduced by carefully designing the end-user's system.

$I2 = C \times V \times f$

Where

C = equivalent capacitance of the chip. (TBD)

V = operating voltage

f = CKI frequency

The typical capacitance for the COP820C is TBD pF.

Some sample current drain values at V_{CC} = 6V are:

CKI (MHz)	Inst. Cycle (μS)	It (mA)
20	1	15
3.58	3	3.6
2	5	2
0.3	3	0.3
0 (HALT)	—	<0.0001

HALT MODE

The COP820C supports a power saving mode of operation: HALT. The COP820C is placed in the HALT mode by setting the G7 data bit, alternatively the user can stop the clock input. In the HALT mode all internal processor activities including the clock oscillator are stopped. The fully static architecture of the COP820C freezes the state of the controller and retains all information until continuing. In the HALT mode, the COP820C power requirements are minimal as it draws only leakage currents and output current. The applied voltage (V_{CC}) may be decreased down to Vr (minimum RAM retention voltage) without altering the state of the machine.

There are two ways to exit the HALT mode: via the $\overline{\text{RESET}}$ or by the CKO pin. A low on the $\overline{\text{RESET}}$ line reinitializes the microcontroller and start executing from the address 0000H. A low to high transition on the CKO pin causes the microcontroller to continue with no reinitialization from the address following the HALT instruction.

INTERRUPTS

The COP820C provides a sophisticated interrupt structure to allow easy interface to the real word. There are three possible interrupt sources, as shown below.

A maskable interrupt on external G0 input (positive or negative edge sensitive under software control)

A maskable interrupt on timer carry or timer capture

A non-maskable software/error interrupt on opcode zero

INTERRUPT CONTROL

The GIE (global interrupt enable) bit enables the interrupt function. This is used in conjunction with ENI and ENTI to select one or both of the interrupt sources. This bit is reset when interrupt is acknowledged.

ENI and ENTI bits select external and timer interrupt respectively. Thus the user can select either or both sources to interrupt the microcontroller when GIE is enabled.

IEDG selects the external interrupt edge (0 = rising edge, 1 = falling edge). The user can get an interrupt on both rising and falling edges by toggling the state of IEDG bit after each interrupt.

IPND and TPND bits signal which interrupt is pending. After interrupt is acknowledged, the user can check these two bits to determine which interrupt is pending. This permits the interrupts to be prioritized under software. The pending flags have to be cleared by the user. Setting the GIE bit high inside the interrupt subroutine allows nested interrupts.

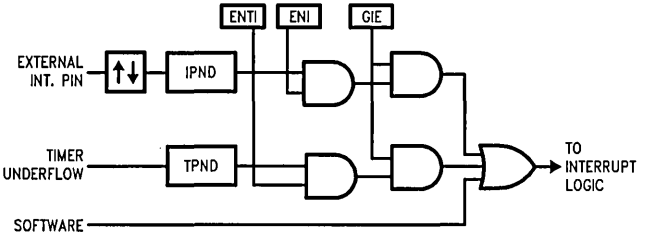
The software interrupt does not reset the GIE bit. This means that the COP820C can be interrupted by other interrupt sources while servicing the software interrupt.

INTERRUPT PROCESSING

The interrupt, once acknowledged, pushes the program counter (PC) onto the stack and the stack pointer (SP) is decremented twice. The Global Interrupt Enable (GIE) bit is reset to disable further interrupts. The microcontroller then vectors to the address 00FFH and continues from that address. At the end of the interrupt subroutine, any of the following three instructions return the processor back to the main program: RET, RETSK or RETI. Either one of the three instructions will pop the stack into the program counter (PC). The stack pointer is then incremented twice. The RETI instruction additionally sets the GIE bit to re-enable further interrupts.

Either of the three instructions can be used to return from a hardware interrupt subroutine. The RETSK instruction should be used when returning from a software interrupt subroutine to avoid entering an infinite loop.

Functional Description (Continued)



TL/DD/9103–11

FIGURE 6. Interrupt Block Diagram

DETECTION OF ILLEGAL CONDITIONS

The COP820C incorporates a hardware mechanism that allows it to detect illegal conditions which may occur from coding errors, noise and 'brown out' voltage drop situations. Specifically it detects cases of executing out of undefined ROM area and unbalanced stack situations.

Reading an undefined ROM location returns 00 (hexadecimal) as its contents. The opcode for a software interrupt is also '00'. Thus a program accessing undefined ROM will cause a software interrupt.

Reading an undefined RAM location returns a FF (hexadecimal). The subroutine stack on the COP820C grows down for each subroutine call. By initializing the stack pointer to 02FH the first unbalanced return instruction will cause the stack pointer to address undefined RAM. As a result the program will attempt to execute from FFFF (hexadecimal), which is an undefined ROM location and will trigger a software interrupt.

MICROWIRE/PLUS™

MICROWIRE/PLUS is a serial synchronous communications interface. The MICROWIRE/PLUS capability enables the COP820C to interface with any of National Semiconductor's Microwire peripherals (i.e. A/D converters, display drivers, EEPROMS, etc.) and with other microcontrollers which support the Microwire interface. It consists of an 8-bit serial shift register (SIO) with serial data input (SI), serial data output (SO) and serial shift clock (SK). *Figure 7* shows the block diagram of the MICROWIRE/PLUS interface.

The shift clock can be selected from either an internal source or an external source. Operating the Microwire arrangement with the internal clock source is called the Master mode of operation. Similarly, operating the Microwire arrangement with an external shift clock is called the Slave mode of operation.

The CNTRL register is used to configure and control the MICROWIRE mode. To use the MICROWIRE, the MSEL bit in the CNTRL register is set to one. The SK clock rate is selected by the two bits, S0 and S1, in the CNTRL register. Table III details the different clock rates that may be selected.

TABLE III

S1	S0	SK Rate
0	0	$2xt_C$
0	1	$4xt_C$
1	x	$8xt_C$

where,

t_C is the instruction cycle clock.

MICROWIRE PLUS OPERATION

Setting the BUSY bit in the PSW register causes the Microwire arrangement to start shifting the data. It gets reset when eight data bits have been shifted. The user may reset the BUSY bit by software to allow less than 8 bits to shift. The COP820C may enter the MICROWIRE PLUS mode either as a Master or as a Slave. *Figure 8* shows how two COP820C microcontrollers and several peripherals may be interconnected using the MICROWIRE/PLUS arrangement.

Master MICROWIRE PLUS Operation

In the MICROWIRE PLUS Master mode of operation the shift clock (SK) is generated internally by the COP820C. The Microwire Master always initiates all data exchanges. (See *Figure 8*). The MSEL bit in the CNTRL register must be set to enable the SO and SK functions onto the G Port. The SO and SK pins must also be selected as outputs by setting appropriate bits in the Port G configuration register. Table IV summaries the bit settings required for Master mode of operation.

SLAVE MICROWIRE PLUS OPERATION

In the Microwire Plus Slave mode of operation the SK clock is generated by an external source. Setting the MSEL bit in the CNTRL register enables the SO and SK functions onto the G Port. The SK pin must be selected as an input and the SO pin is selected as an output pin by setting and resetting the appropriate bit in the Port G configuration register. Table IV summarizes the settings required to enter the Slave mode of operation.

The user must set the BUSY flag immediately upon entering the Slave mode. This will ensure that all data bits sent by the Master will be shifted properly. After eight clock pulses the BUSY flag will be cleared and the sequence may be repeated. (See *Figure 8*.)

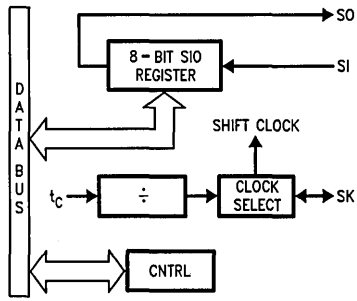
Functional Description (Continued)

TABLE IV

G4 Config. Bit	G5 Config. Bit	G4 Fun.	G5 Fun.	G6 Fun.	Operation
1	1	SO	Int. SK	SI	MICROWIRE Master
0	1	TRI-STATE	Int. SK	SI	MICROWIRE Master
1	0	SO	Ext. SK	SI	MICROWIRE Slave
0	0	TRI-STATE	Ext. SK	SI	MICROWIRE Slave

TIMER/COUNTER

The COP820C has a powerful 16-bit timer with an associated 16-bit register enabling it to perform extensive timer functions. The timer T1 and its register R1 are each organized as two 8-bit read/write registers. Control bits in the register CNTRL allow the timer to be started and stopped under software control. The timer-register pair can be operated in one of three possible modes.



TL/DD/9103-12

FIGURE 7. MICROWIRE Block Diagram

MODE 1. TIMER WITH AUTO-LOAD REGISTER

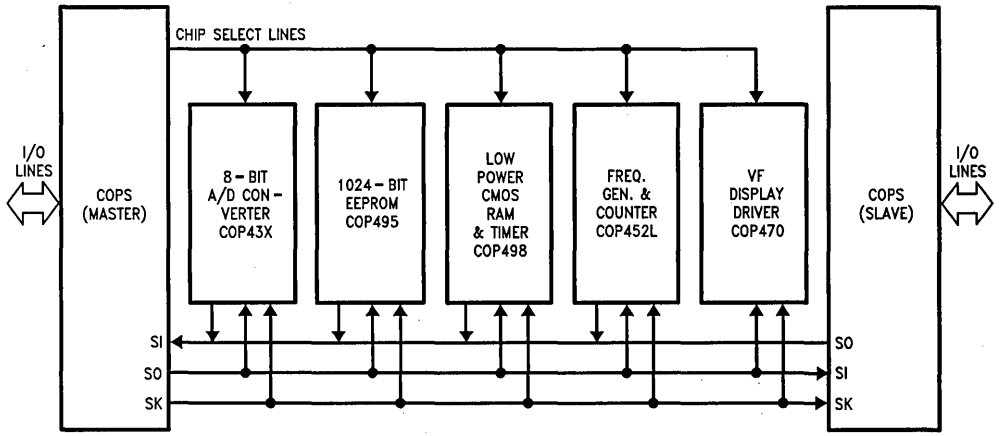
In this mode of operation the timer T1 counts down at the instruction cycle rate. Upon underflow the value in the register R1 gets automatically reloaded into the timer which continues to count down. The timer underflow can be programmed to interrupt the microcontroller. A bit in the control register CNTRL enables the TIO (G3) pin to toggle upon timer underflows. This allow the generation of square-wave outputs or pulse width modulated outputs under software control. (See *Figure 9*)

MODE 2. EXTERNAL COUNTER

In this mode, the timer T1 becomes a 16-bit external event counter. The counter counts down upon an edge on the TIO pin. Control bits in the register CNTRL program the counter to decrement either on a positive edge or on a negative edge. Upon underflow the contents of the register R1 are automatically copied into the counter. The underflow can also be programmed to generate an interrupt. (See *Figure 9*)

MODE 3. TIMER WITH CAPTURE REGISTER

Timer T1 can be used to precisely measure external frequencies or events in this mode of operation. The timer T1 counts down at the instruction cycle rate. Upon the occurrence of a specified edge on the TIO pin the contents of the timer T1 are copied into the register R1. Bits in the control register CNTRL allow the trigger edge to be specified either as a positive edge or as a negative edge. In this mode the user can elect to be interrupted on the specified trigger edge. (See *Figure 10*.)



TL/DD/9103-13

FIGURE 8. Microwire Application

Functional Description (Continued)

TABLE V. Timer Operating Modes

CNTRL Bits 7 6 5	Operation Mode	T Interrupt	Timer Counts On
0 0 0	External Counter W/Auto-Load Reg.	Timer Carry	TIO Pos. Edge
0 0 1	External Counter W/Auto-Load Reg.	Timer Carry	TIO Neg. Edge
0 1 0	Not Allowed	Not Allowed	Not Allowed
0 1 1	Not Allowed	Not Allowed	Not Allowed
1 0 0	Timer W/Auto-Load Reg.	Timer Carry	t_C
1 0 1	Timer W/Auto-Load Reg./Toggle TIO Out	Timer Carry	t_C
1 1 0	Timer W/Capture Register	TIO Pos. Edge	t_C
1 1 1	Timer W/Capture Register	TIO Neg. Edge	t_C

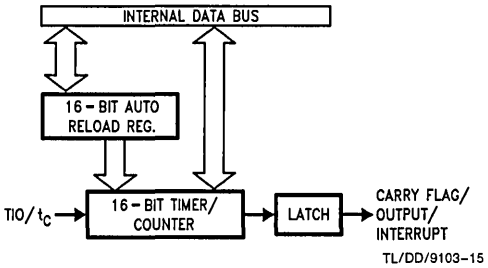


FIGURE 9. Timer/Counter Auto Reload Mode Block Diagram

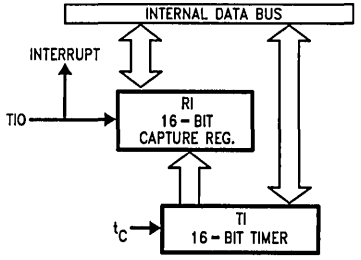


FIGURE 10. Timer Capture Mode Block Diagram

TIMER PWM APPLICATION

Figure 11 shows how a minimal component D/A converter can be built out of the Timer-Register pair in the Auto-Reload mode. The timer is placed in the "Timer with auto reload" mode and the TIO pin is selected as the timer output. At the outset the TIO pin is set high, the timer T1 holds the on time and the register R1 holds the signal off time. Setting TRUN bit starts the timer which counts down at the instruction cycle rate. The underflow toggles the TIO output and copies the off time into the timer, which continues to run. By alternately loading in the on time and the off time at each successive interrupt a PWM frequency can be easily generated.

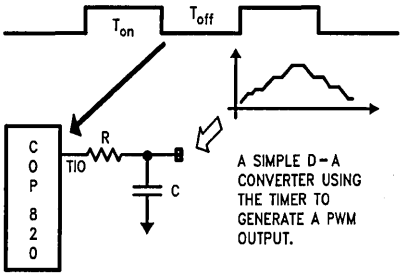


FIGURE 11. Timer Application

Control Registers

CNTRL REGISTER (ADDRESS X'OOEE)

The Timer and MICROWIRE control register contains the following bits:

S1 & S0 Select the MICROWIRE clock divide-by

IEDG External interrupt edge polarity select (0 = rising edge, 1 = falling edge)

MSEL Enable MICROWIRE functions S0 and SK

TRUN Start/Stop the Timer/Counter (1 = run, 0 = stop)

TEDG Timer input edge polarity select (0 = rising edge, 1 = falling edge)

CSEL Selects the capture mode

TSEL Selects the timer mode

TSEL	CSEL	TEDG	TRUN	MSEL	IEDG	S1	S0

BIT 7 BIT 0

PSW REGISTER (ADDRESS x'00EF)

The PSW register contains the following select bits:

GIE Global interrupt enable

ENI External interrupt enable

BUSY MICROWIRE busy shifting

IPND External interrupt pending

ENTI Timer interrupt enable

TPND Timer interrupt pending

C Carry Flag

HC Half carry Flag

HC	C	TPND	ENTI	IPND	BUSY	ENI	GIE

Bit 7 Bit 0

Operating Modes

The COP820C offers the user two operating modes: Single Chip mode and the ROMless mode. The operating mode is determined by the state of the D2 pin at power on reset.

SINGLE CHIP MODE

In the Single Chip mode, the COP820C functions as a self contained microcontroller. It can address internal memory consisting of 64 bytes of RAM and 1024 bytes of ROM. All ports configured as memory mapped I/O ports.

ROMLESS MODE

The COP820C enters the ROMless mode of operation if the D2 pin is held at logical "0" at reset. In this case the internal ROM is disabled and the COP820C can now address up to 32 kbytes of external program memory. It continues to use the on board 64 bytes of RAM. The ports D and I are used to access the external program memory. By providing a serial interface to external program memory the COP820C manages a large address space without the penalty of losing a large number of I/O pins in the process. *Figure 12* shows in schematic form the logic required for the ROMless mode operation and all support logic required to recreate the I/O.

Memory Map

All RAM, ports and registers (except A and PC) are mapped into data memory address space.

Address	Contents
00 to 2F	On Chip RAM Bytes
30 to 7F	Unused RAM Address Space (Reads as all Ones)
80 to BF	Expansion Space for on Chip EERAM
C0 to CF	Expansion Space for I/O and Registers
D0 to DF	On Chip I/O and Registers
D0	Port L Data Register
D1	Port L Configuration Register
D2	Port L Input Pins (Read Only)
D3	Reserved for Port L
D4	Port G Data Register
D5	Port G Configuration Register
D6	Port G Input Pins (Read Only)
D7	Port I Input Pins (Read Only)
D8-DB	Reserved for Port C
DC-DF	Port D
E0 to EF	On Chip Functions and Registers
E0-E7	Reserved for Future Parts
E8	Reserved
E9	MICROWIRE Shift Register
EA	Timer Lower Byte
EB	Timer Upper Byte
EC	Timer Autoload Register Lower Byte
ED	Timer Autoload Register Upper Byte
EE	CNTRL Control Register
EF	PSW Register
F0 to FF	On Chip RAM Mapped as Registers
FC	X Register
FD	SP Register
FE	B Register

Reading memory locations 30H-7FH will return all ones. Reading other unused memory locations will return undefined data.

Addressing Modes

REGISTER INDIRECT

This is the "normal" mode of addressing for COP820C. The operand is the memory addressed by the B register or X register.

DIRECT

The instruction contains an 8-bit address field that directly points to the data memory for the operand.

IMMEDIATE

The instruction contains an 8-bit immediate field as the operand.

REGISTER INDIRECT (AUTO INCREMENT AND DECREMENT)

This is a register indirect mode that automatically increments or decrements the B or X register after executing the instruction.

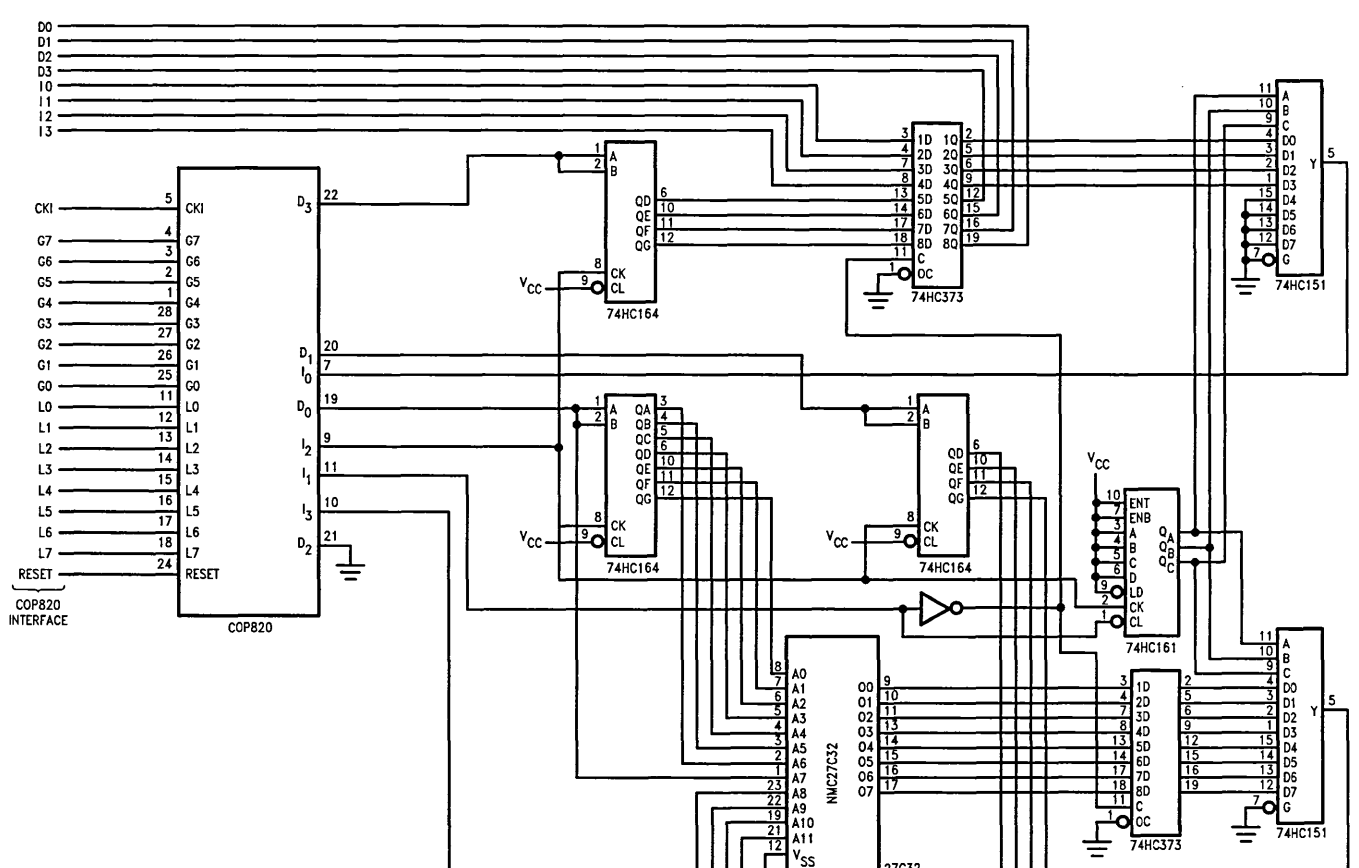


FIGURE 12. COP820C ROMless Mode Schematic

Addressing Modes (Continued)

RELATIVE

This mode is used for the JP instruction, the instruction field is added to the program counter to get the new program location. JP has a range of from −31 to +32 to allow a one byte relative jump (JP + 1 is implemented by a NOP instruction). There are no 'pages' when using JP, all 15 bits of PC are used.

Instruction Set

REGISTER AND SYMBOL DEFINITIONS

Registers

A	8-bit Accumulator register
B	8-bit Address register
X	8-bit Address register
SP	8-bit Stack pointer register
PC	15-bit Program counter register
PU	upper 7 bits of PC
PL	lower 8 bits of PC
C	1-bit of PSW register for carry
HC	Half Carry
GIE	1-bit of PSW register for global interrupt enable

Symbols

[B]	Memory indirectly addressed by B register
[X]	Memory indirectly addressed by X register
Mem	Direct address memory or [B]
MemI	Direct address memory or [B] or Immediate data
Imm	8-bit Immediate data
Reg	Register memory: addresses F0 to FF (Includes B, X and SP)
Bit	Bit number (0 to 7)
←	Loaded with
⟷	Exchanged with

Instruction Set

ADD	add	A ← A + MemI
ADC	add with carry	A ← + MemI + C, C ← Carry HC ← Half Carry
SUBC	subtract with carry	A ← A + $\overline{\text{MemI}}$ +C, C ← Carry HC ← Half Carry
AND	Logical AND	A ← A and MemI
OR	Logical OR	A ← A or MemI
XOR	Logical Exclusive-OR	A ← A xor MemI
IFEQ	IF equal	Compare A and MemI, Do next if A = MemI
IFGT	IF greater than	Compare A and MemI, Do next if A > MemI
IFBNE	IF B not equal	Do next if lower 4 bits of B ≠ Imm
DRSZ	Decrement Reg. ,skip if zero	Reg ← Reg − 1, skip if Reg goes to 0
SBIT	Set bit	1 to bit, Mem (bit= 0 to 7 immediate)
RBIT	Reset bit	0 to bit, Mem
IFBIT	If bit	If bit, Mem is true, do next instr.
X	Exchange A with memory	A ⟷ Mem
LD A	Load A with memory	A ← MemI
LD mem	Load Direct memory Immed.	Mem ← Imm
LD Reg	Load Register memory Immed.	Reg ← Imm
X	Exchange A with memory [B]	A ⟷ [B] (B ← B±1)
X	Exchange A with memory [X]	A ⟷ [X] (X ← X±1)
LD A	Load A with memory [B]	A ← [B] (B ← B±1)
LD A	Load A with memory [X]	A ← [X] (X ← X±1)
LD M	Load Memory Immediate	[B] ← Imm (B ← B±1)
CLRA	Clear A	A ← 0
INCA	Increment A	A ← A + 1
DECA	Decrement A	A ← A − 1
LAID	Load A indirect from ROM	A ← ROM(PU,A)
DCORA	DECIMAL CORRECT A	A ← BCD correction (follows ADC, SUBC)
RRCA	ROTATE A RIGHT THRU C	C → A7 → ... → A0 → C
SWAPA	Swap nibbles of A	A7...A4 ⟷ A3...A0
SC	Set C	C ← 1, HC ← 1
RC	Reset C	C ← 0, HC ← 0
IFC	If C	If C is true, do next instruction
IFNC	If not C	If C is not true, do next instruction
JMPL	Jump absolute long	PC ← ii (ii = 15 bits, 0 to 32k)
JMP	Jump absolute	PC11..0 ← i (i = 12 bits)
JP	Jump relative short	PC ← PC + r (r is −31 to +32, not 1)
JSRL	Jump subroutine long	[SP] ← PL,[SP-1] ← PU,SP-2,PC ← ii
JSR	Jump subroutine	[SP] ← PL,[SP-1] ← PU,SP-2,PC11..0 ← i
JID	Jump indirect	PL ← ROM(PU,A)
RET	Return from subroutine	SP+2,PL ← [SP],PU ← [SP-1]
RETSK	Return and Skip	SP+2,PL ← [SP],PU ← [SP-1],Skip next instruction
RETI	Return from Interrupt	SP+2,PL ← [SP],PU ← [SP-1],GIE ← 1
INTR	Generate an interrupt	[SP] ← PL,[SP−1] ← PU,SP-2,PC ← 0FF
NOP	No operation	PC ← PC + 1

Opcode List

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
JP -15	JP -31	LD 0F0,#i	DRSZ 0F0	RRCA	RC	ADC A,#i	ADC A,[B]	IFBIT 0,[B]	*	LD B, 0F	IFBNE 0	JSR 0000-00FF	JMP 0000-00FF	JP + 17	INTR	0
JP -14	JP -30	LD 0F1,#i	DRSZ 0F1	*	SC	SUBC A,#i	SUBC A,[B]	IFBIT 1,[B]	*	LD B, 0E	IFBNE 1	JSR 0100-01FF	JMP 0100-01FF	JP + 18	JP + 2	1
JP -13	JP -29	LD 0F2,#i	DRSZ 0F2	X A,[X+]	X A,[B+]	IFEQ A,#i	IFEQ A,[B]	IFBIT 2,[B]	*	LD B, 0D	IFBNE 2	JSR 0200-02FF	JMP 0200-02FF	JP + 19	JP + 3	2
JP -12	JP -28	LD 0F3,#i	DRSZ 0F3	X A,[X−]	X A,[B−]	IFGT A,#i	IFGT A,[B]	IFBIT 3,[B]	*	LD B, 0C	IFBNE 3	JSR 0300-03FF	JMP 0300-03FF	JP + 20	JP + 4	3
JP -11	JP -27	LD 0F4,#i	DRSZ 0F4	*	LAID	ADD A,#i	ADD A,[B]	IFBIT 4,[B]	CLRA	LD B, 0B	IFBNE 4	JSR 0400-04FF	JMP 0400-04FF	JP + 21	JP + 5	4
JP -10	JP -26	LD 0F5,#i	DRSZ 0F5	*	JID	AND A,#i	AND A,[B]	IFBIT 5,[B]	SWAPA	LD B,#0A	IFBNE 5	JSR 0500-05FF	JMP 0500-05FF	JP + 22	JP + 6	5
JP -9	JP -25	LD 0F6,#i	DRSZ 0F6	X A,[X]	X A,[B]	XOR A,#i	XOR A,[B]	IFBIT 6,[B]	DCORA	LD B, 9	IFBNE 6	JSR 0600-06FF	JMP 0600-06FF	JP + 23	JP + 7	6
JP -8	JP -24	LD 0F7,#i	DRSZ 0F7	*	*	OR A,#i	OR A,[B]	IFBIT 7,[B]	*	LD B, 8	IFBNE 7	JSR 0700-07FF	JMP 0700-07FF	JP + 24	JP + 8	7
JP -7	JP -23	LD 0F8,#i	DRSZ 0F8	NOP	*	LD A,#i	IFC	SBIT 0,[B]	RBIT 0,[B]	LD B, 7	IFBNE 8	JSR 0800-08FF	JMP 0800-08FF	JP + 25	JP + 9	8
JP -6	JP -22	LD 0F9,#i	DRSZ 0F9	*	*	*	IFNC	SBIT 1,[B]	RBIT 1,[B]	LD B, 6	IFBNE 9	JSR 0900-09FF	JMP 0900-09FF	JP + 26	JP + 10	9
JP -5	JP -21	LD 0FA,#i	DRSZ 0FA	LD A, [X+]	LD A,[B+]	LD [B+],#i	INCA	SBIT 2,[B]	RBIT 2,[B]	LD B, 5	IFBNE 0A	JSR 0A00-0AFF	JMP 0A00-0AFF	JP + 27	JP + 11	A
JP -4	JP -20	LD 0FB,#i	DRSZ 0FB	LD A,[X−]	LD A,[B−]	LD [B−],#i	DECA	SBIT 3,[B]	RBIT 3,[B]	LD B, 4	IFBNE 0B	JSR 0B00-0BFF	JMP 0B00-0BFF	JP + 28	JP + 12	B
JP -3	JP -19	LD 0FC,#i	DRSZ 0FC	LD Md,#i	JMPL	X A,Md	*	SBIT 4,[B]	RBIT 4,[B]	LD B, 3	IFBNE 0C	JSR 0C00-0CFF	JMP 0C00-0CFF	JP + 29	JP +13	C
JP -2	JP -18	LD 0FD,#i	DRSZ 0FD	DIR	JSRL	LD A,Md	RETSK	SBIT 5,[B]	RBIT 5,[B]	LD B, 2	IFBNE 0D	JSR 0D00-0DFF	JMP 0D00-0DFF	JP + 30	JP +14	D
JP -1	JP -17	LD 0FE,#i	DRSZ 0FE	LD A,[X]	LD A,[B]	LD [B], #;	RET	SBIT 6, [B]	RBIT 6, [B]	LD B, 1	IFBNE 0E	JSR 0E00−0EFF	JMP 0E00−0EFF	JP + 31	JP +15	E
JP -0	JP -16	LD 0FF,#1	DRSZ 0FF	*	*	*	RETI	SBIT 7,[B]	RBIT 7,[B]	LD B, 0	IFBNE 0F	JSR 0F00-0FFF	JMP 0F00-0FFF	JP + 32	JP + 16	F

where, i is the immediate data Md is a directly addressed memory location * is an unused opcode (see following table)

Instruction Execution Time

Most instructions are single byte (with immediate addressing mode instruction taking two bytes).

Most single instructions take one cycle time (1 μs at 20 MHz) to execute.

See the BYTES and CYCLES per INSTRUCTION table for details.

BYTES and CYCLES per INSTRUCTION

The following table shows the number of bytes and cycles for each instruction in the format of byte/cycle (a cycle is 1 μs at 20 MHz).

	[B]	Direct	Immed.
ADD	1/1	3/4	2/2
ADC	1/1	3/4	2/2
SUBC	1/1	3/4	2/2
AND	1/1	3/4	2/2
OR	1/1	3/4	2/2
XOR	1/1	3/4	2/2
IFEQ	1/1	3/4	2/2
IFGT	1/1	3/4	2/2
IFBNE	1/1		
DRSZ		1/3	
SBIT	1/1	3/4	
RBIT	1/1	3/4	
IFBIT	1/1	3/4	

Memory Transfer Instructions

	Register Indirect		Direct	Immed.	Register Indirect Auto Incr & Decr		
	[B]	[X]			[B+, B−]	[X+, X−]	
X A,*	1/1	1/3	2/3		1/2	1/3	
LD A,*	1/1	1/3	2/3	2/2	1/2	1/3	
LD B,Imm				1/1			(If B < 16)
LD B,Imm				2/3			(If B > 15)
LD Mem,Imm	2/2		3/3		2/2		
LD Reg,Imm				2/3			

* => Memory location addressed by B or X or directly.

Instructions Using A & C

CLRA	1/1
INCA	1/1
DECA	1/1
LAID	1/3
DCORA	1/1
RRCA	1/1
SWAPA	1/1
SC	1/1
RC	1/1
IFC	1/1
IFNC	1/1

Transfer of Control Instructions

JMPL	3/4
JMP	2/3
JP	1/3
JSRL	3/5
JSR	2/5
JID	1/3
RET	1/5
RETSK	1/5
RETI	1/5
INTR	1/7
NOP	1/1

The following table shows the instructions assigned to unused opcodes.

Unused Opcode	Instruction	Unused Opcode	Instruction
60	NOP	A9	NOP
61	NOP	AF	LD A, [B]
62	NOP	B1	NOP
63	NOP	B4	NOP
67	NOP	B5	NOP
8C	RET	B7	X A, [X]
99	NOP	B9	NOP
9F	LD [B], #i	BF	LD A, [X]
A7	X A, [B]		
A8	NOP		

Development Support

The MOLE (Microcomputer On Line Emulator) is a low cost development system and emulator for all microcontroller products. These include COPs, TMP, 8050U and the HPC family of products. The MOLE consists of a BRAIN Board, Personality Board and optional host software.

The purpose of the MOLE is to provide the user with a tool to write and assemble code, emulate code for the target microcontroller and assist in both software and hardware debugging of the system.

It is a self contained computer with its own firmware which provides for all system operation, emulation control, communication, PROM programming and diagnostic operations.

It contains three serial ports to optionally connect to a terminal, a host system, a printer or a modem, or to connect to other MOLEs in a multi-MOLE environment.

MOLE can be used in either a stand alone mode or in conjunction with a selected host system ie those using CP/M or PC-DOS communicating via a RS-232 port.

Dial-A-Helper is a service provided by the MOLE applications group. If a user is having difficulty in getting a MOLE to operate in a particular mode or it is acting peculiar, he can contact the Microcontroller Group, NSC via his system and a modem. He can leave messages on the Dial-A-Helper Bulletin Board System.

The Dial-A-Helper also maintains an accurate and up to date software library that the user can access freely. Software subroutines, math packages and assemblers are examples of the variety offered over the Dial-A-Helper.

Single Chip Emulator Device

The COP820C is fully supported by a form, fit and function emulator device, the COP8720C.

Option List

The COP820C/COP821C/COP822C mask programmable options are listed out below. The options are programmed at the same time as the ROM pattern to provide the user with hardware flexibility to use a variety of oscillator configuration.

OPTION 1: CKI INPUT

= 1 Normal Mode Crystal	(CKI/10) CKO for crystal configuration
= 2 Normal Mode External	(CKI/10) CKO available as G7 input
= 3 R/C	(CKI/10) CKO available as G7 input
= 4 High Speed Crystal	(CKI/20) CKO for crystal configuration
= 5 High Speed External	(CKI/20) CKO available as G7 input

OPTION 2: COP820C BONDING

= 1 28 pin package
= 2 24 pin package
= 3 20 pin package

National Semiconductor Corporation

PRELIMINARY

COP840C 2k Single-Chip 8-Bit Low Cost Microcontroller

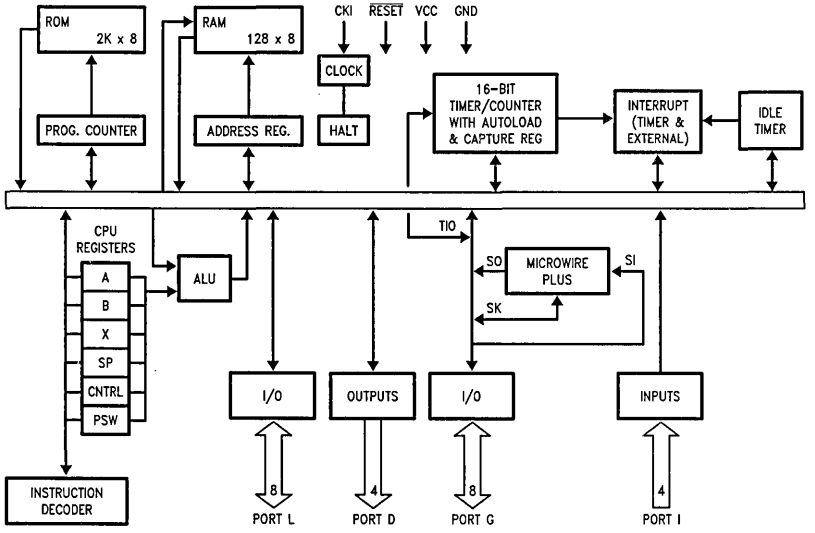
General Description

The COP840C is a member of the COPS™ microcontroller family. It is a fully static part, fabricated using double-metal silicon gate microCMOS technology. This low cost microcontroller is a complete microcomputer containing all system timing, interrupt logic, ROM, RAM, and I/O necessary to implement dedicated control functions in a variety of applications. Features include an 8-bit memory mapped architecture, MICROWIRE™ serial I/O, a 16-bit timer/counter, and a multi-sourced interrupt. Each I/O pin has software selectable options to adapt the COP840 to the specific application. The part operates over a voltage range of 2.5V to 6.0V. High throughput is achieved with an efficient, regular instruction set operating at a 1 μs per instruction rate. The part may be operated in the ROMless mode to provide for accurate emulation and for applications requiring external program memory.

Features

- Low cost 8-bit microController
- Fully static CMOS
- 1 μs instruction time (20 MHz clock)
- Low current drain (2 mA at 4 μs instruction rate)
- Extra-low current static HALT mode (5 μA)
- Single supply operation: 2.5V to 6.0V
- 2048 x 8 ROM, 128 x 8 RAM
- 16-bit read/write timer operates in a variety of modes
  - — Timer with 16-bit auto reload register
  - — 16-bit external event counter
  - — Timer with 16-bit capture register (selectable edge)
- Multi-source vectored interrupt
  - — Reset master clear
  - — External interrupt with selectable edge
  - — Timer interrupt or capture interrupt
  - — Software/"watch-dog" interrupt
- Idle timer with interrupt
- Power saving IDLE and HALT modes
- On chip comparator
- 8-bit stack pointer (stack in RAM)
- Powerful instruction set, most instructions single byte
- MICROWIRE/PLUS™ serial I/O
- 28 pin package
- 24 input/output pins
- Software selectable I/O options (TRI-STATE®, push-pull, weak pull-up)
- I/O ports and function compatible with the COP400 family
- Extended temperature ranges: −40°C to +85°C (−55°C to +125°C to be available)
- ROMless mode to get accurate emulation and external program capability
- Fully supported by National's MOLE™ development system

Block Diagram



TL/DD/9106–1

National Semiconductor Corporation

PRELIMINARY

COP8620C 1k Single-Chip 8-Bit Low Cost Microcontroller with E²PROM Data Memory

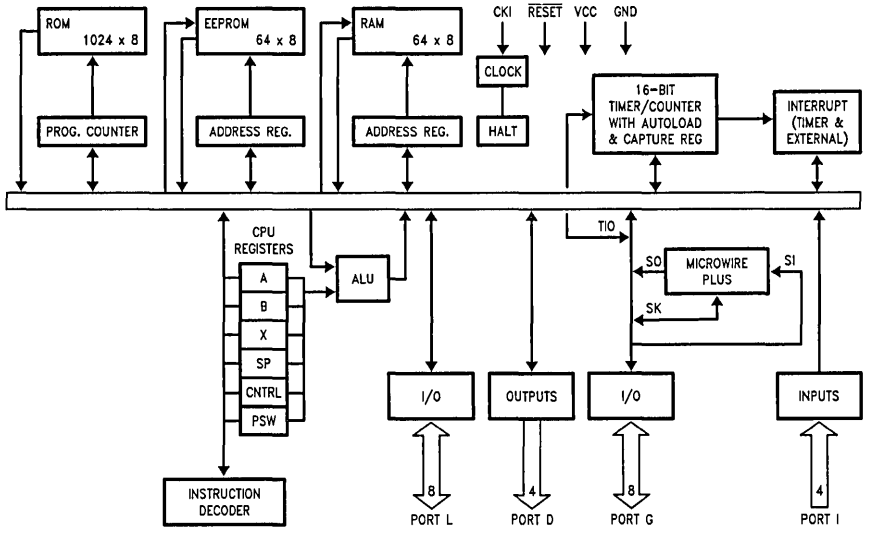
General Description

The COP8620C is a member of the COPS™ microcontroller family. It is a fully static part, fabricated using double-metal silicon gate microCMOS technology. This low cost microcontroller is a complete microcomputer containing all system timing, interrupt logic, ROM, RAM, EEPROM and I/O necessary to implement dedicated control functions in a variety of applications. Features include an 8-bit memory mapped architecture, MICROWIRE™ serial I/O, a 16-bit timer/counter, and a multi-sourced interrupt. Each I/O pin has software selectable options to adapt the COP8620 to the specific application. The part operates over a voltage range of 2.7V to 6.0V. High throughput is achieved with an efficient, regular instruction set operating at a 1 μs per instruction rate. The part may be operated in the ROMless mode to provide for accurate emulation and for applications requiring external program memory.

Features

- Low cost 8-bit microController
- Fully static CMOS
- 1 μs instruction time (20 MHz clock)
- Low current drain (2 mA at 4 μs instruction rate)
- Extra-low current static HALT mode (5 μA)
- Single supply operation: 2.7V to 6.0V
- 1024 x 8 ROM, 64 x 8 RAM, 64 x 8 EEPROM
- 16-bit read/write timer operates in a variety of modes
  - — Timer with 16-bit auto reload register
  - — 16-bit external event counter
  - — Timer with 16-bit capture register (selectable edge)
- Multi-source interrupt
  - — Reset master clear
  - — External interrupt with selectable edge
  - — Timer interrupt or capture interrupt
  - — Software/"watch-dog" interrupt
- 8-bit stack pointer (stack in RAM)
- Powerful instruction set, most instructions single byte
- MICROWIRE/PLUS™ serial I/O
- 28 pin package (optionally 24 or 20 pin package)
- 24 input/output pins
- Software selectable I/O options (TRI-STATE®, push-pull, weak pull-up)
- I/O ports and function compatible with the COP400 family
- Extended temperature ranges: −40°C to +85°C (−55°C to +125°C to be available)
- ROMless mode to get accurate emulation and external program capability
- Fully supported by National's MOLE™ development system

Block Diagram



TL/DD/9104-1

National Semiconductor Corporation

PRELIMINARY

COP8640C 2k Single-Chip 8-Bit Low Cost Microcontroller with E²PROM Data Memory

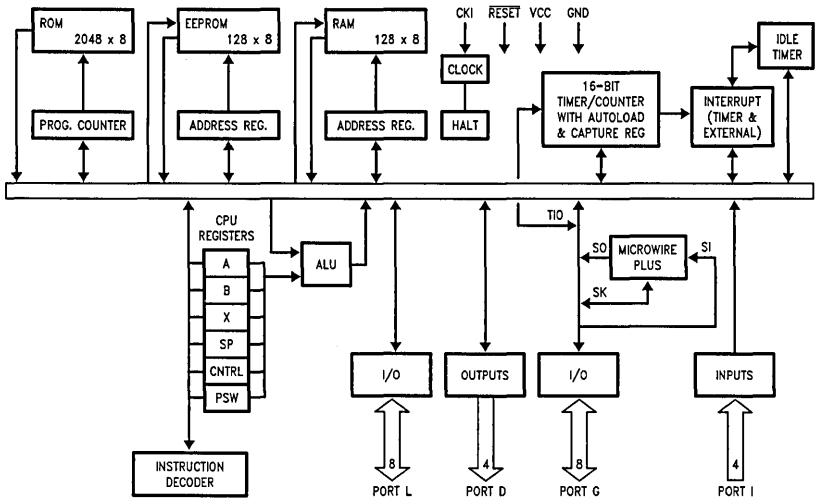
General Description

The COP8640C is a member of the COPS™ microcontroller family. It is a fully static part, fabricated using double-metal silicon gate microCMOS technology. This low cost microcontroller is a complete microcomputer containing all system timing, interrupt logic, ROM, RAM, EEPROM and I/O necessary to implement dedicated control functions in a variety of applications. Features include an 8-bit memory mapped architecture, MICROWIRE™ serial I/O, a 16-bit timer/counter, and a multi-sourced interrupt. Each I/O pin has software selectable options to adapt the COP8640 to the specific application. The part operates over a voltage range of 2.7V to 6.0V. High throughput is achieved with an efficient, regular instruction set operating at a 1 μs per instruction rate. The part may be operated in the ROMless mode to provide for accurate emulation and for applications requiring external program memory.

Features

- Low cost 8-bit microController
- Fully static CMOS
- 1 μs instruction time (20 MHz clock)
- Low current drain (2 mA at 4 μs instruction rate)
- Extra-low current static HALT mode (5 μA)
- Single supply operation: 2.7V to 6.0V
- 2048 x 8 ROM, 128 x 8 RAM, 128 x 8 EEPROM
- 16-bit read/write timer operates in a variety of modes
  - — Timer with 16-bit auto reload register
  - — 16-bit external event counter
  - — Timer with 16-bit capture register (selectable edge)
- Multi-source vectored interrupt
  - — Reset master clear
  - — External interrupt with selectable edge
  - — Timer interrupt or capture interrupt
  - — Software/"watch-dog" interrupt
- Idle timer with interrupt
- Power saving IDLE and HALT modes
- On-chip comparator
- 8-bit stack pointer (stack in RAM)
- Powerful instruction set, most instructions single byte
- MICROWIRE/PLUS™ serial I/O
- 28 pin package
- 24 input/output pins
- Software selectable I/O options (TRI-STATE®, push-pull, weak pull-up)
- I/O ports and function compatible with the COP400 family
- Extended temperature ranges: −40°C to +85°C (−55°C to +125°C to be available)
- ROMless mode to get accurate emulation and external program capability
- Fully supported by National's MOLE™ development system

Block Diagram



TL/DD/9105–1

National Semiconductor Corporation

PRELIMINARY

COP8720C 1k Single-Chip 8-Bit Low Cost Microcontroller with E²PROM Program and Data Memory

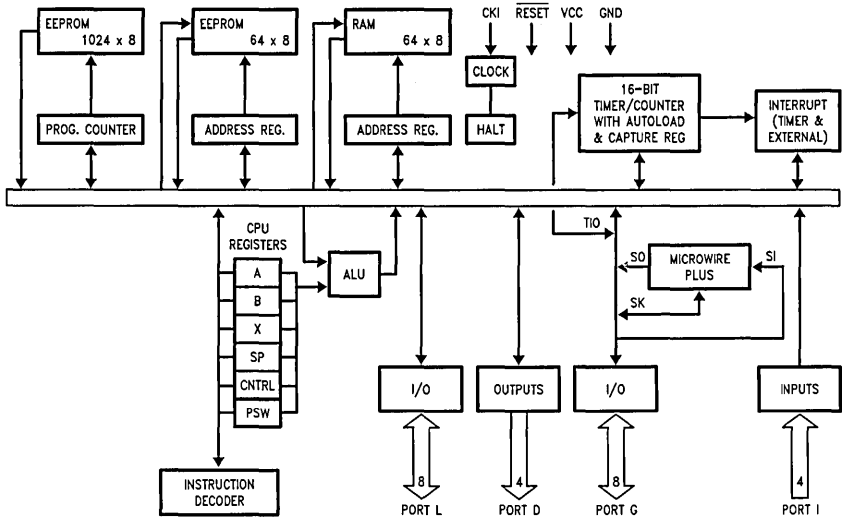
General Description

The COP8720C is a member of the COPS™ microcontroller family. It is a fully static part, fabricated using double-metal silicon gate microCMOS technology. This low cost microcontroller is a complete microcomputer containing all system timing, interrupt logic, EEPROM in ROM address space, RAM, EEPROM and I/O necessary to implement dedicated control functions in a variety of applications. Features include an 8-bit memory mapped architecture, MICROWIRE™ serial I/O, a 16-bit timer/counter, and a multi-sourced interrupt. Each I/O pin has software selectable options to adapt the COP800 to the specific application. The part operates over a voltage range of 2.7V to 6.0V. High throughput is achieved with an efficient, regular instruction set operating at a 1 μs per instruction rate. This part is the E² full form, fit, function emulator for the COP820C and COP8620C family parts.

Features

- Low cost 8-bit microController
- Fully static CMOS
- 1 μs instruction time (20 MHz clock)
- Low current drain (2 mA at 4 μs instruction rate)
- Extra-low current static HALT mode (5 μA)
- Single supply operation: 2.7V to 6.0V
- 1024 x 8 EEPROM, 64 x 8 RAM, 64 x 8 EEPROM
- 16-bit read/write timer operates in a variety of modes
  - — Timer with 16-bit auto reload register
  - — 16-bit external event counter
  - — Timer with 16-bit capture register (selectable edge)
- Multi-source interrupt
  - — Reset master clear
  - — External interrupt with selectable edge
  - — Timer interrupt or capture interrupt
  - — Software/"watch-dog" interrupt
- 8-bit stack pointer (stack in RAM)
- Powerful instruction set, most instructions single byte
- MICROWIRE/PLUS™ serial I/O
- 28 pin package (optionally 24 or 20 pin package)
- 24 input/output pins
- Software selectable I/O options (TRI-STATE®, push-pull, weak pull-up)
- I/O ports and function compatible with the COP400 family
- Extended temperature ranges: −40°C to +85°C (−55°C to +125°C to be available)
- Fully supported by National's MOLE™ development system

Block Diagram



TL/DD/9108-1

National Semiconductor Corporation

PRELIMINARY

COP8740C 2k Single-Chip 8-Bit Low Cost Microcontroller with E²PROM Program and Data Memory

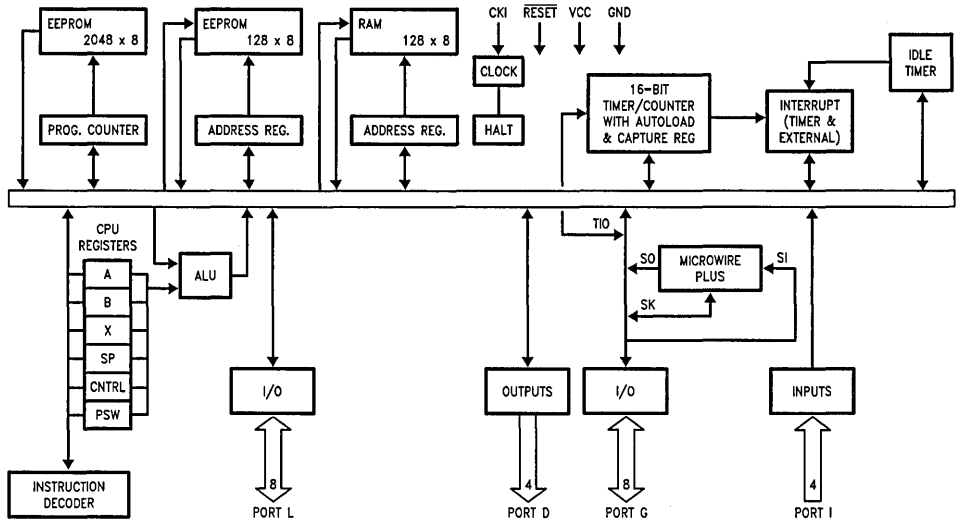
General Description

The COP8740C is a member of the COPS™ microcontroller family. It is a fully static part, fabricated using double-metal silicon gate microCMOS technology. This low cost microcontroller is a complete microcomputer containing all system timing, interrupt logic, EEPROM on ROM address space, RAM, EEPROM and I/O necessary to implement dedicated control functions in a variety of applications. Features include an 8-bit memory mapped architecture, MICROWIRE™ serial I/O, a 16-bit timer/counter, and a multi-sourced interrupt. Each I/O pin has software selectable options to adapt the COP8740C to the specific application. The part operates over a voltage range of 2.7V to 6.0V. High throughput is achieved with an efficient, regular instruction set operating at a 1 μs per instruction rate. This part is the E² full form, function emulator for the COP840C and COP8640C Family Parts.

Features

- Low cost 8-bit microController
- Fully static CMOS
- 1 μs instruction time (20 MHz clock)
- Low current drain (2 mA at 4 μs instruction rate)
- Extra-low current static HALT mode (5 μA)
- Single supply operation: 2.7V to 6.0V
- 2048 x 8 EEPROM, 128 x RAM, 128 x 8 EEPROM
- 16-bit read/write timer operates in a variety of modes
  - — Timer with 16-bit auto reload register
  - — 16-bit external event counter
  - — Timer with 16-bit capture register (selectable edge)
- Multi-source vectored interrupt
  - — Reset master clear
  - — External interrupt with selectable edge
  - — Timer interrupt or capture interrupt
  - — Software/"watch-dog" interrupt
- Idle timer with interrupt
- Power saving IDLE and HALT modes
- On-chip Comparator
- 8-bit stack pointer (stack in RAM)
- Powerful instruction set, most instructions single byte
- MICROWIRE/PLUS™ serial I/O
- 28 pin package
- 24 input/output pins
- Software selectable I/O options (TRI-STATE®, push-pull, weak pull-up)
- I/O ports and function compatible with the COP400 family
- Extended temperature ranges: −40°C to +85°C (−55°C to +125°C to be available)
- Fully supported by National's MOLE™ development system

Block Diagram



TL/DD/9107-1

Section 3

COPS Peripherals

Section 3 Contents

National Semiconductor Corporation

ADC0831/ADC0832/ADC0834 and ADC0838 (COP431/COP432/COP434 and COP438) 8-Bit Serial I/O A/D Converters with Multiplexer Options

General Description

The ADC0831 series are 8-bit successive approximation A/D converters with a serial I/O and configurable input multiplexers with up to 8 channels. The serial I/O is configured to comply with the NSC MICROWIRE™ serial data exchange standard for easy interface to the COPS™ family of processors, and can interface with standard shift registers or μPs.

The 2-, 4- or 8-channel multiplexers are software configured for single-ended or differential inputs as well as channel assignment.

The differential analog voltage input allows increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

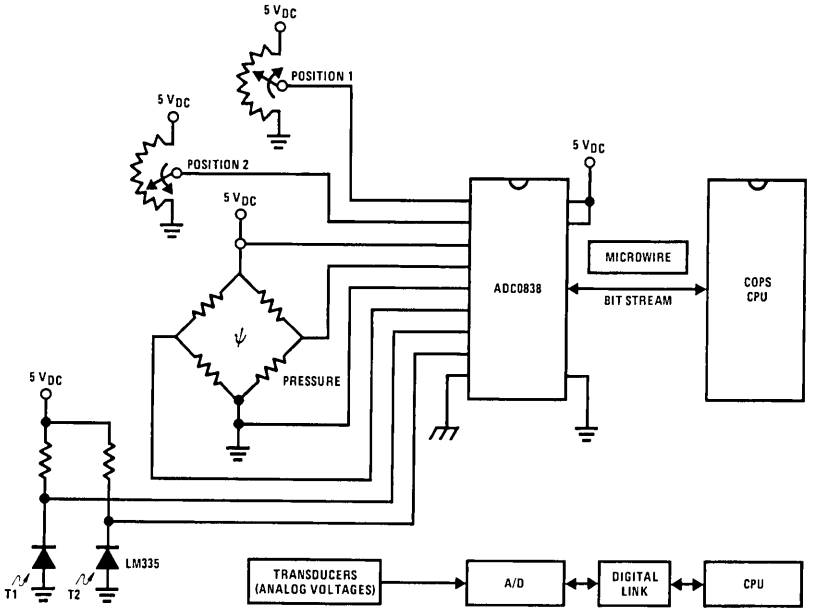
Features

- NSC MICROWIRE compatible—direct interface to COPS family processors
- Easy interface to all microprocessors, or operates "stand-alone"
- Operates ratiometrically or with 5 V_{DC} voltage reference
- No zero or full-scale adjust required
- 2-, 4- or 8-channel multiplexer options with address logic
- Shunt regulator allows operation with high voltage supplies
- 0V to 5V input range with single 5V power supply
- Remote operation with serial digital data link
- T^2L/MOS input/output compatible
- 0.3" standard width, 8-, 14- or 20-pin DIP package

Key Specifications

■ Resolution	8 Bits
■ Total Unadjusted Error	±½ LSB and ±1 LSB
■ Single Supply	5 V_{DC}
■ Low Power	15 mW
■ Conversion Time	32 μs

Typical Application



TL/H/5583–1

Absolute Maximum Ratings (Notes 1 & 2)

Current into V+ (Note 3)	15 mA
Supply Voltage, V_{CC} (Note 3)	6.5V
Voltage	
Logic Inputs	−0.3V to +15V
Analog Inputs	−0.3V to V_{CC} +0.3V
Input Current per Pin	±5 mA
Package	±20 mA
Storage Temperature	−65°C to +150°C
Package Dissipation at T_A = 25°C (Board Mount)	0.8W
Lead Temp. (Soldering, 10 seconds)	300°C

Operating Ratings (Notes 1 & 2)

Supply Voltage, V_{CC}	4.5 V_{DC} to 6.3 V_{DC}
Temperature Range	$T_{MIN} \le T_A \le T_{MAX}$
ADC0831/2/4/8BJ	−55°C to +125°C
ADC0831/2/4/8CJ	
ADC0831/2/4/8BCJ	−40°C to +85°C
ADC0831/2/4/8CCJ	
ADC0831/2/4/8BCN	−0°C to +70°C
ADC0831/2/4/8CCN	

Converter and Multiplexer Electrical Characteristics

The following specifications apply for V_{CC} = V+ = 5V, $T_A = T_j$ = 25°C, and f_{CLK} = 250 kHz unless otherwise specified. **Boldface limits apply from T_{MIN} to T_{MAX}.**

Parameter	Conditions	BCJ and CCJ Devices			BCN and CCN Devices			Units
		Typ (Note 9)	Tested Limit (Note 10)	Design Limit (Note 11)	Typ (Note 9)	Tested Limit (Note 10)	Design Limit (Note 11)	
CONVERTER AND MULTIPLEXER CHARACTERISTICS								
Total Unadjusted Error	V_{REF} = 5.00 V (Note 4)							
ADC0831/2/4/8BCN			+½			±½	±½	LSB
ADC0831/2/4/8BJ			±½			±½		
ADC0831/2/4/8BCJ			±½					
ADC0831/2/4/8CCN						±1	±1	
ADC0831/2/4/8CJ			±1					
ADC0831/2/4/8CCJ			±1					
Minimum Reference Input Resistance		2.4	**1.3**		2.4	1.3	**1.3**	kΩ
Maximum Reference Input Resistance		2.4	**5.9**		2.4	5.4	**5.4**	kΩ
Maximum Common-Mode Input Range (Note 5)			**V_{CC} +0.05**			V_{CC} +0.05	**V_{CC} +0.05**	V
Minimum Common-Mode Input Range (Note 5)			**GND −0.05**			GND −0.05	**GND −0.05**	V
DC Common-Mode Error		±1/16	±¼	±¼	±1/16	±¼	±¼	LSB
Change in zero error from V_{CC} = 5V to internal zener operation (Note 3)	15 mA into V+ V_{CC} = N.C. V_{REF} = 5V		**1**			1	**1**	LSB
V_Z, internal diode breakdown (at V+) (Note 3) MIN	15 mA into V+		**6.3**			6.3	**6.3**	V
MAX			**8.5**			8.5	**8.5**	
Power Supply Sensitivity	V_{CC} = 5V ±5%	±1/16	±¼	±¼	±1/16	±¼	±¼	LSB
I_{OFF}, Off Channel Leakage Current (Note 6)	On Channel = 5V Off Channel = 0V		**−1**			−1	**−1**	µA
	On Channel = 0V Off Channel = 5V		**+1**			+1	**+1**	µA
I_{ON}, On Channel Leakage Current (Note 6)	On Channel = 0V Off Channel = 5V		**−1**			−1	**−1**	µA
	On Channel = 5V Off Channel = 0V		**+1**			+1	**+1**	µA

Note 1: Absolute Maximum Ratings are those values beyond which the life of the device may be impaired.

Note 2: All voltages are measured with respect to ground.

Note 3: Internal zener diodes (6.3 to 8.5V) are connected from V+ to GND and V_{CC} to GND. The zener at V+ can operate as a shunt regulator and is connected to V_{CC} via a conventional diode. Since the zener voltage equals the A/D's breakdown voltage, the diode insures that V_{CC} will be below breakdown when the device is powered from V+. Functionality is therefore guaranteed for V+ operation even though the resultant voltage at V_{CC} may exceed the specified Absolute Max of 6.5V. It is recommended that a resistor be used to limit the max current into V+. (See *Figure 3*)

Note 4: Total unadjusted error includes offset, full-scale, linearity, and multiplexer errors.

Converter and Multiplexer Electrical Characteristics (Continued)

The following specifications apply for $V_{CC} = V+ = 5V$, $T_A = T_j = 25°C$, and $f_{CLK} = 250$ kHz unless otherwise specified. **Boldface limits apply from T_{MIN} to T_{MAX}.**

Parameter	Conditions	BCJ and CCJ Devices			BCN and CCN Devices			Units
		Typ (Note 9)	Tested Limit (Note 10)	Design Limit (Note 11)	Typ (Note 9)	Tested Limit (Note 10)	Design Limit (Note 11)	
DIGITAL AND DC CHARACTERISTICS								
$V_{IN(1)}$, Logical "1" Input Voltage (Min)	$V_{CC} = 5.25V$		**2.0**			2.0	**2.0**	V
$V_{IN(0)}$, Logical "0" Input Voltage (Max)	$V_{CC} = 4.75V$		**0.8**			0.8	**0.8**	V
$I_{IN(1)}$, Logical "1" Input Current (Max)	$V_{IN} = 5.0V$	0.005	**1**		0.005	1	**1**	μA
$I_{IN(0)}$, Logical "0" Input Current (Max)	$V_{IN} = 0V$	−0.005	**−1**		−0.005	−1	**−1**	μA
$V_{OUT(1)}$, Logical "1" Output Voltage (Min)	$V_{CC} = 4.75V$ $I_{OUT} = -360$ μA $I_{OUT} = -10$ μA		 **2.4** **4.5**			 2.4 4.5	 **2.8** **4.6**	 V V
$V_{OUT(0)}$, Logical "0" Output Voltage (Max)	$V_{CC} = 4.75V$ $I_{OUT} = 1.6$ mA		**0.4**			0.4	**0.34**	V
I_{OUT}, TRI-STATE Output Current (Max)	$V_{OUT} = 0V$ $V_{OUT} = 5V$	−0.01 0.01	**−3** **3**		−0.01 0.01	−3 +3	**−3** **+3**	μA μA
I_{SOURCE}, Output Source Current (Min)	$V_{OUT} = 0V$	−14	**−6.5**		−14	−6.5	**−7.5**	mA
I_{SINK}, Output Sink Current (Min)	$V_{OUT} = V_{CC}$	16	**8.0**		16	8.0	**9.0**	mA
I_{CC}, Supply Current (Max) ADC0831, ADC0834, ADC0838		1	**2.5**		1	2.5	**2.5**	mA
ADC0832	Includes Ladder Current	3	**7.2**		3	7.2	**7.2**	mA

AC Characteristics

The following specifications apply for $V_{CC} = 5V$, $t_r = t_f = 20$ ns and 25°C unless otherwise specified.

Parameter		Conditions	Typ (Note 4)	Tested Limit (Note 5)	Design Limit (Note 6)	Limit Units
f_{CLK}, Clock Frequency	Min Max			10	 400	kHz kHz
t_C, Conversion Time		Not including MUX Addressing Time		8		$1/f_{CLK}$
Clock Duty Cycle (Note 7)	Min Max				40 60	% %
t_{SET-UP}, CS Falling Edge or Data Input Valid to CLK Rising Edge					250	ns
t_{HOLD}, Data Input Valid after CLK Rising Edge					90	ns
t_{pd1}, t_{pd0}—CLK Falling Edge to Output Data Valid (Note 8)		$C_L = 100$ pF Data MSB First Data LSB First	 650 250		 1500 600	 ns ns
t_{1H}, t_{0H},—Rising Edge of CS to Data Output and SARS Hi-Z		$C_L = 10$ pF, $R_L = 10k$ (see TRI-STATE® Test Circuits)	125		250	ns
		$C_L = 100$ pf, $R_L = 2k$		500		ns
C_{IN}, Capacitance of Logic Input			5			pF
C_{OUT}, Capacitance of Logic Outputs			5			pF

Note 5: For $V_{IN}(-) \geq V_{IN}(+)$ the digital output code will be 0000 0000. Two on-chip diodes are tied to each analog input (see Block Diagram) which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater then the V_{CC} supply. Be careful, during testing at low V_{CC} levels (4.5V), as high level analog inputs (5V) can cause this input diode to conduct—especially at elevated temperatures, and cause errors for analog inputs near full-scale. The spec allows 50 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 50 mV, the output code will be correct. To achieve an absolute 0 V_{DC} to 5 V_{DC} input voltage range will therefore require a minimum supply voltage of 4.950 V_{DC} over temperature variations, initial tolerance and loading.

Note 6: Leakage current is measured with the clock not switching.

Note 7: A 40% to 60% clock duty cycle range insures proper operation at all clock frequencies. In the case that an available clock has a duty cycle outside of these limits, the minimum, time the clock is high or the minimum time the clock is low must be at least 1 μs.

Note 8: Since data, MSB first, is the output of the comparator used in the successive approximation loop, an additional delay is built in (see Block Diagram) to allow for comparator response time.

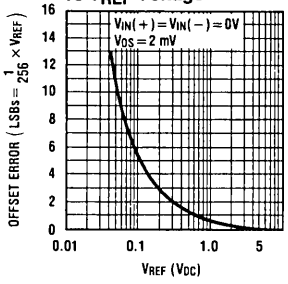
Note 9: Typicals are at 25°C and represent most likely parametric values.

Note 10: Guaranteed and 100% production tested.

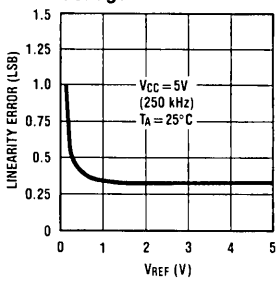
Note 11: Guaranteed but not 100% production tested. These limits are not used to calculate outgoing quality levels.

Typical Performance Characteristics

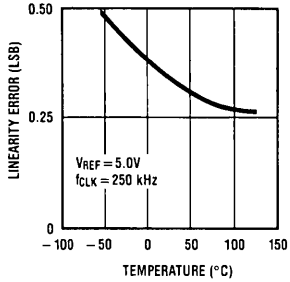
Unadjusted Offset Error vs V_{REF} Voltage



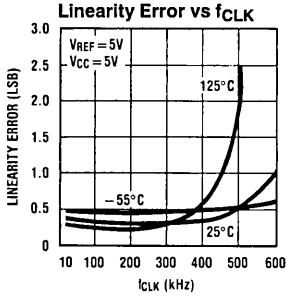
Linearity Error vs V_{REF} Voltage



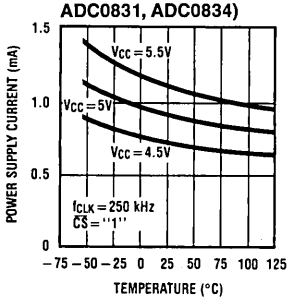
Linearity Error vs Temperature



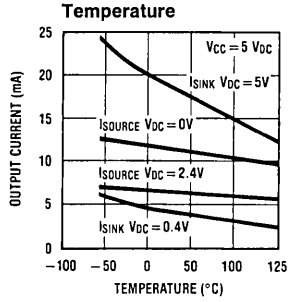
Linearity Error vs f_{CLK}



Power Supply Current vs Temperature (ADC0838, ADC0831, ADC0834)

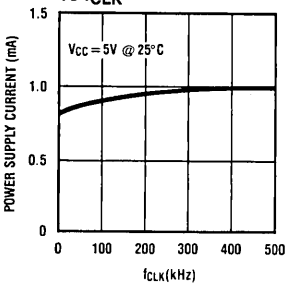


Output Current vs Temperature



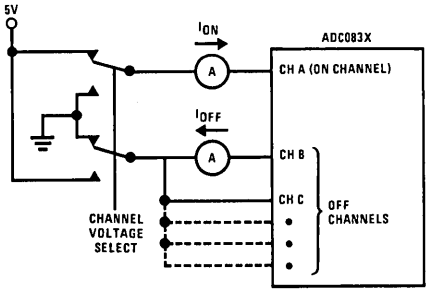
Note: For ADC0832 and I_{REF}

Power Supply Current vs f_{CLK}



TL/H/5583–2

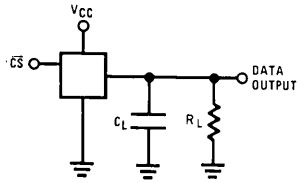
Leakage Current Test Circuit



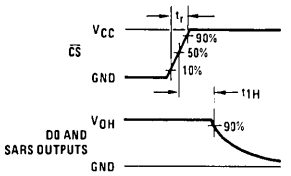
TL/H/5583–3

TRI-STATE Test Circuits and Waveforms

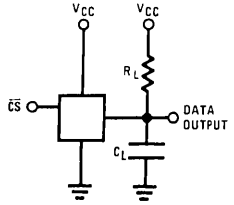
t_{1H}



t_{1H}

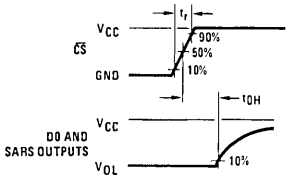


t_{0H}



TL/H/5583-4

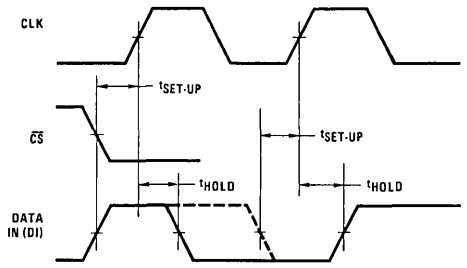
t_{0H}



TL/H/5583-23

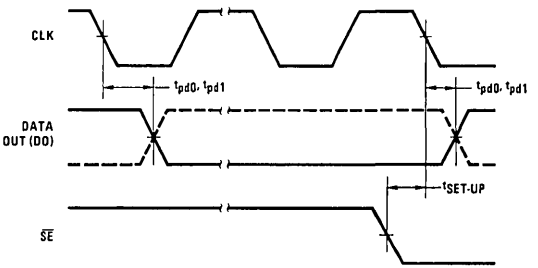
Timing Diagrams

Data Input Timing



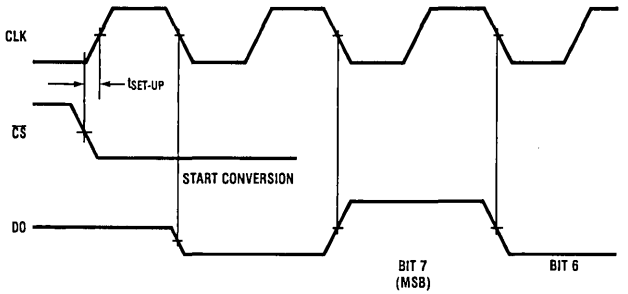
TL/H/5583-24

Data Output Timing



TL/H/5583-25

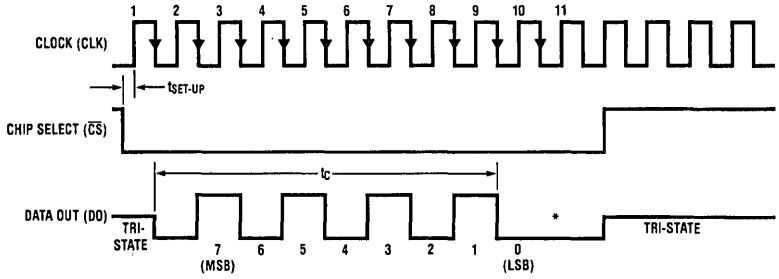
ADC0831 Start Conversion Timing



TL/H/5583-26

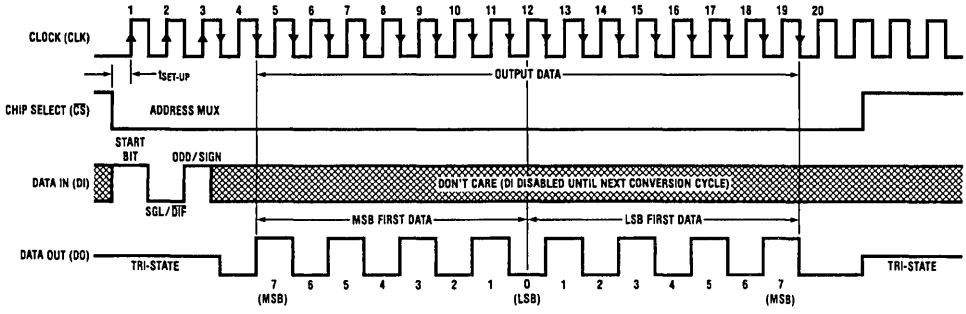
Timing Diagrams (Continued)

ADC0831 Timing



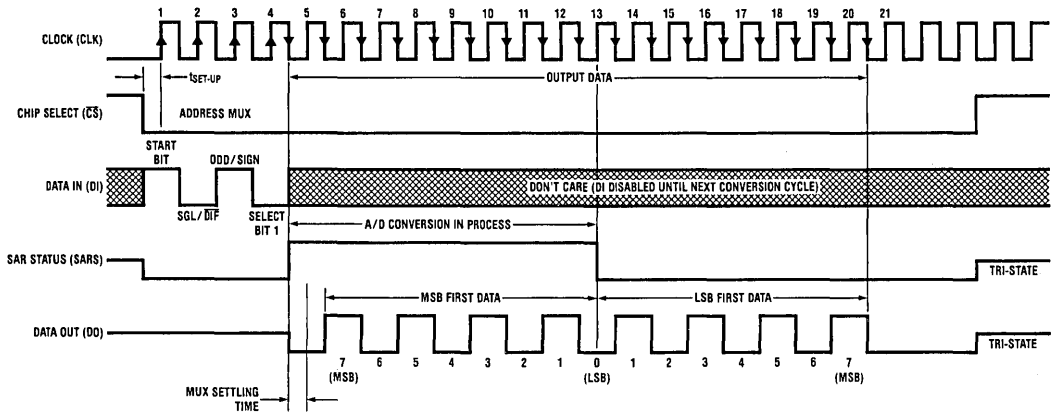
TL/H/5583–27

ADC0832 Timing



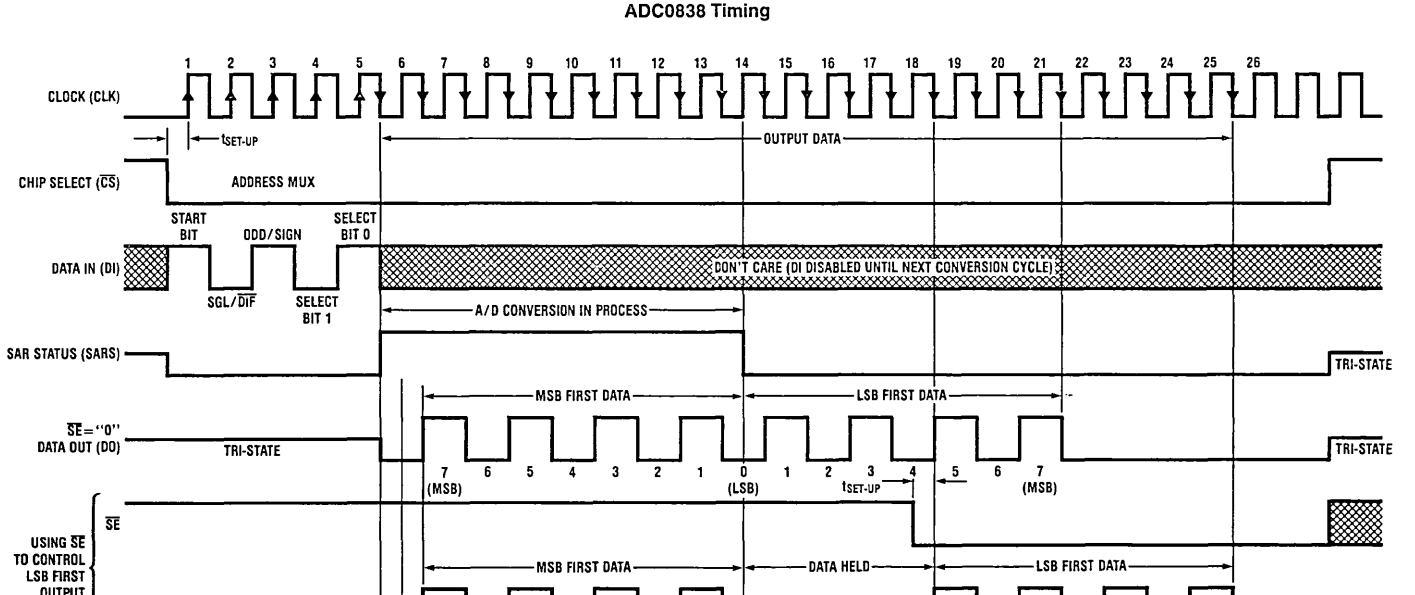
TL/H/5583–28

ADC0834 Timing



TL/H/5583–5

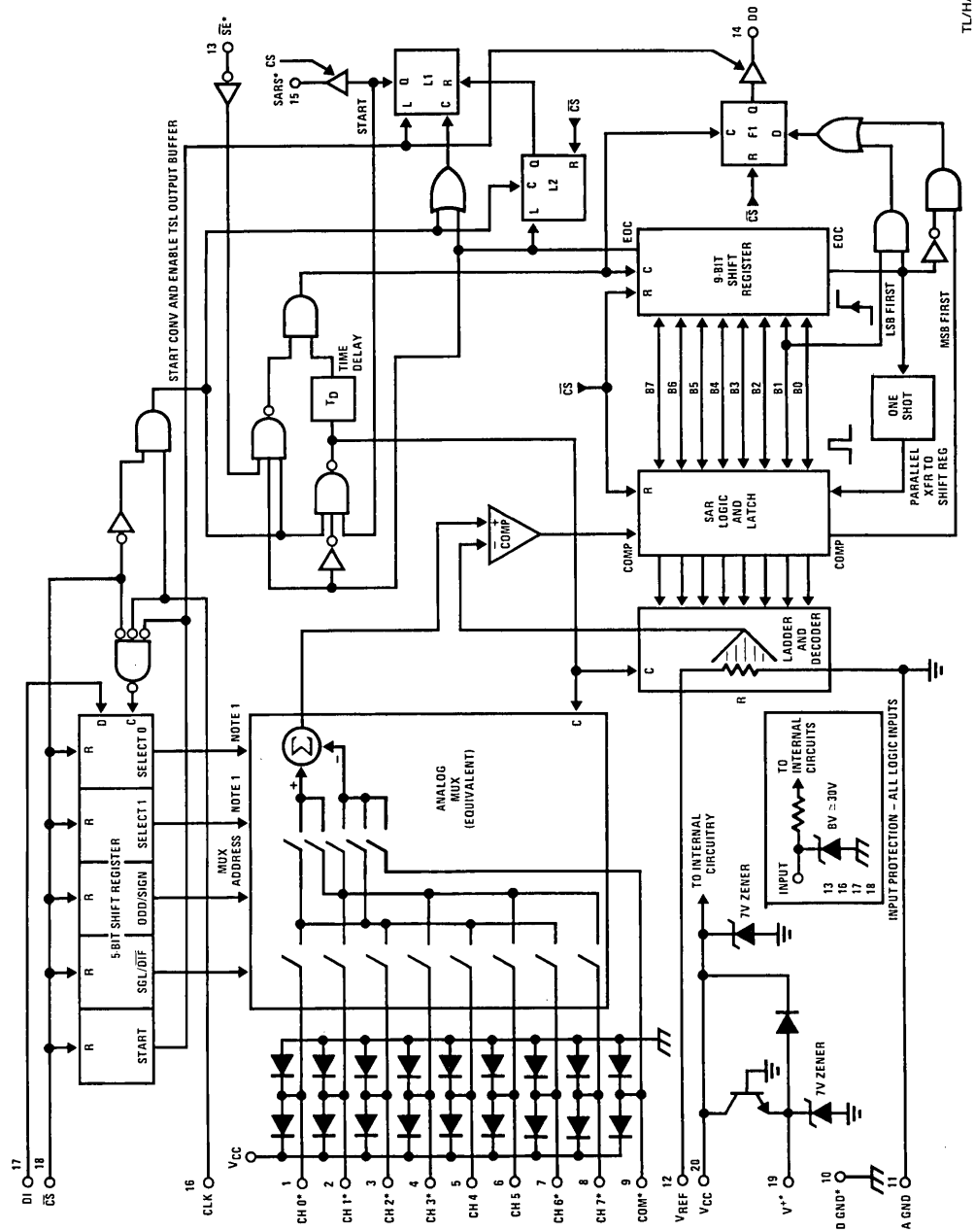
ADC0838 Timing



TL/H/5583–6

* Make sure clock edge #18 clocks in the LSB before $\overline{SE}$ is taken low

ADC0838 Functional Block Diagram



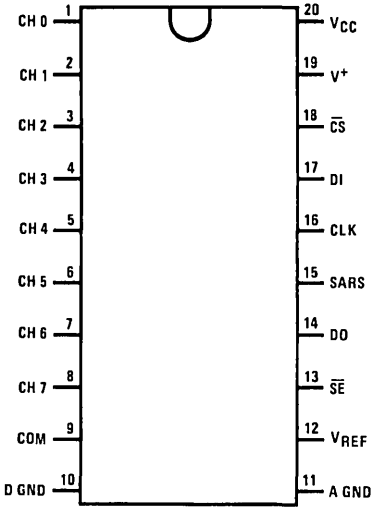
*Some of these functions/pins are not available with other options.

Note 1: For the ADC0834, DI is input directly to the D input of SELECT 1. SELECT 0 is forced to a "1". For the ADC0832, DI is input directly to the DI input of ODD/SIGN. SELECT 0 is forced to a "0" and SELECT 1 is forced to a "1".

Connection Diagrams

ADC0838 8-Channel MUX

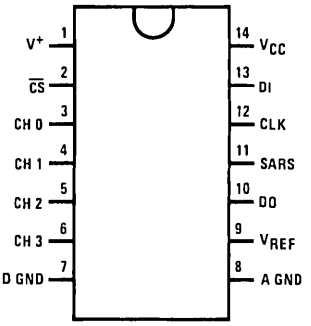
Dual-In-Line Package



TOP VIEW

ADC0834 4-Channel MUX

Dual-In-Line Package

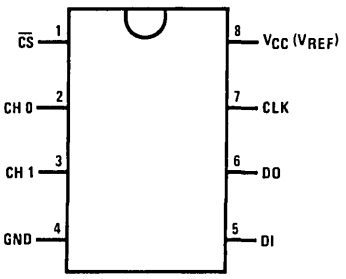


TOP VIEW

COM internally connected to A GND

ADC0832 2-Channel MUX

Dual-In-Line Package



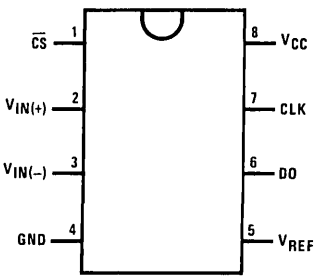
TOP VIEW

COM internally connected to GND.

REF internally connected to V_{CC}.

ADC0831 Single Differential Input

Dual-In-Line Package



TOP VIEW

TL/H/5583–8

3

Functional Description

1.0 MULTIPLEXER ADDRESSING

The design of these converters utilizes a sample-data comparator structure which provides for a differential analog input to be converted by a successive approximation routine.

The actual voltage converted is always the difference between an assigned "+" input terminal and a "−" input terminal. The polarity of each input terminal of the pair being converted indicates which line the converter expects to be the most positive. If the assigned "+" input is less than the "−" input the converter responds with an all zeros output code.

A unique input multiplexing scheme has been utilized to provide multiple analog channels with software-configurable single-ended, differential, or a new pseudo-differential option which will convert the difference between the voltage at any analog input and a common terminal. The analog signal conditioning required in transducer-based data acquisition systems is significantly simplified with this type of input flexibility. One converter package can now handle ground referenced inputs and true differential inputs as well as signals with some arbitrary reference voltage.

A particular input configuration is assigned during the MUX addressing sequence, prior to the start of a conversion. The MUX address selects which of the analog inputs are to be enabled and whether this input is single-ended or differential. In the differential case, it also assigns the polarity of the channels. Differential inputs are restricted to adjacent channel pairs. For example channel 0 and channel 1 may be selected as a different pair but channel 0 or 1 cannot act differentially with any other channel. In addition to selecting differential mode the sign may also be selected. Channel 0 may be selected as the positive input and channel 1 as the negative input or vice versa. This programmability is best illustrated by the MUX addressing codes shown in the following tables for the various product options.

The MUX address is shifted into the converter via the DI line. Because the ADC0831 contains only one differential input channel with a fixed polarity assignment, it does not require addressing.

The common input line on the ADC0838 can be used as a pseudo-differential input. In this mode, the voltage on this pin is treated as the "−" input for any of the other input channels. This voltage does not have to be analog ground; it can be any reference potential which is common to all of the inputs. This feature is most useful in single-supply application where the analog circuitry may be biased up to a potential other than ground and the output signals are all referred to this potential.

TABLE I. Multiplexer/Package Options

Part Number	Alternate Part Number	Number of Analog Channels		Number of Package Pins
		Single-Ended	Differential	
ADC0831	COP431	1	1	8
ADC0832	COP432	2	1	8
ADC0834	COP434	4	2	14
ADC0838	COP438	8	4	20

Functional Description (Continued)

TABLE II. MUX Addressing: ADC0838

Single-Ended MUX Mode

MUX Address				Analog Single-Ended Channel #								
SGL/$\overline{DIF}$	ODD/SIGN	SELECT 1	SELECT 0	0	1	2	3	4	5	6	7	COM
1	0	0	0	+								−
1	0	0	1			+						−
1	0	1	0					+				−
1	0	1	1							+		−
1	1	0	0		+							−
1	1	0	1				+					−
1	1	1	0						+			−
1	1	1	1								+	−

Differential MUX Mode

MUX Address				Analog Differential Channel-Pair #							
SGL/$\overline{DIF}$	ODD/SIGN	SELECT 1	SELECT 0	0		1		2		3	
				0	1	2	3	4	5	6	7
0	0	0	0	+	−						
0	0	0	1			+	−				
0	0	1	0					+	−		
0	0	1	1							+	−
0	1	0	0	−	+						
0	1	0	1			−	+				
0	1	1	0					−	+		
0	1	1	1							−	+

TABLE III. MUX Addressing: ADC0834

Single-Ended MUX Mode

MUX Address			Channel #			
SGL/DIF	ODD/SIGN	SELECT 1	0	1	2	3
1	0	0	+			
1	0	1			+	
1	1	0		+		
1	1	1				+

COM is internally tied to A GND

Differential MUX Mode

MUX Address			Channel #			
SGL/DIF	ODD/SIGN	SELECT 1	0	1	2	3
0	0	0	+	−		
0	0	1			+	−
0	1	0	−	+		
0	1	1			−	+

TABLE IV. MUX Addressing: ADC0832

Single-Ended MUX Mode

MUX Address		Channel #	
SGL/$\overline{DIF}$	ODD/SIGN	0	1
1	0	+	
1	1		+

COM is internally tied to A GND

Differential MUX Mode

MUX Address		Channel #	
SGL/$\overline{DIF}$	ODD/SIGN	0	1
0	0	+	−
0	1	−	+

3

Functional Description (Continued)

Since the input configuration is under software control, it can be modified, as required, at each conversion. A channel can be treated as a single-ended, ground referenced input for one conversion; then it can be reconfigured as part of a differential channel for another conversion. *Figure 1* illustrates the input flexibility which can be achieved.

The analog input voltages for each channel can range from 50 mV below ground to 50 mV above V_{CC} (typically 5V) without degrading conversion accuracy.

2.0 THE DIGITAL INTERFACE

A most important characteristic of these converters is their serial data link with the controlling processor. Using a serial communication format offers two very significant system improvements; it allows more function to be included in the converter package with no increase in package size and it can eliminate the transmission of low level analog signals by locating the converter right at the analog sensor; transmitting highly noise immune digital data back to the host processor.

To understand the operation of these converters it is best to refer to the Timing Diagrams and Functional Block Diagram and to follow a complete conversion sequence. For clarity a separate diagram is shown of each device.

1. A conversion is initiated by first pulling the $\overline{CS}$ (chip select) line low. This line must be held low for the entire conversion. The converter is now waiting for a start bit and its MUX assignment word.

2. A clock is then generated by the processor (if not provided continuously) and output to the A/D clock input.

3. On each rising edge of the clock the status of the data in (DI) line is clocked into the MUX address shift register. The start bit is the first logic "1" that appears on this line (all leading zeros are ignored). Following the start bit the converter expects the next 2 to 4 bits to be the MUX assignment word.

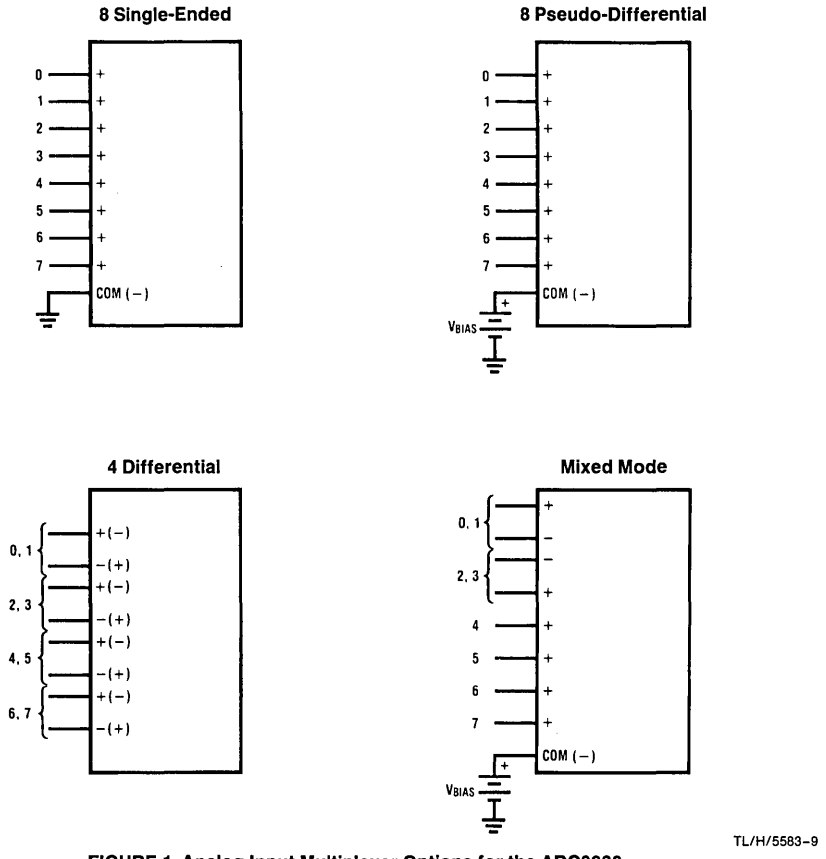


FIGURE 1. Analog Input Multiplexer Options for the ADC0838

Functional Description (Continued)

4. When the start bit has been shifted into the start location of the MUX register, the input channel has been assigned and a conversion is about to begin. An interval of ½ clock period (where nothing happens) is automatically inserted to allow the selected MUX channel to settle. The SAR status line goes high at this time to signal that a conversion is now in progress and the DI line is disabled (it no longer accepts data).

5. The data out (DO) line now comes out of TRI-STATE and provides a leading zero for this one clock period of MUX settling time.

6. When the conversion begins, the output of the SAR comparator, which indicates whether the analog input is greater than (high) or less than (low) each successive voltage from the internal resistor ladder, appears at the DO line on each falling edge of the clock. This data is the result of the conversion being shifted out (with the MSB coming first) and can be read by the processor immediately.

7. After 8 clock periods the conversion is completed. The SAR status line returns low to indicate this ½ clock cycle later.

8. If the programmer prefers, the data can be provided in an LSB first format [this makes use of the shift enable ($\overline{SE}$) control line]. All 8 bits of the result are stored in an output shift register. On devices which do not include the $\overline{SE}$ control line, the data, LSB first, is automatically shifted out the DO line, after the MSB first data stream. The DO line then goes low and stays low until $\overline{CS}$ is returned high. On the ADC0838 the $\overline{SE}$ line is brought out and if held high, the value of the LSB remains valid on the DO line. When $\overline{SE}$ is forced low, the data is then clocked out LSB first. The ADC0831 is an exception in that its data is only output in MSB first format.

9. All internal registers are cleared when the $\overline{CS}$ line is high. If another conversion is desired, $\overline{CS}$ must make a high to low transition followed by address information.

The DI and DO lines can be tied together and controlled through a bidirectional processor I/O bit with one wire. This is possible because the DI input is only "looked-at" during the MUX addressing interval while the DO line is still in a high impedance state.

All of the logic inputs can be taken to 15V independent of the magnitude of the supply voltage, V_{CC}.

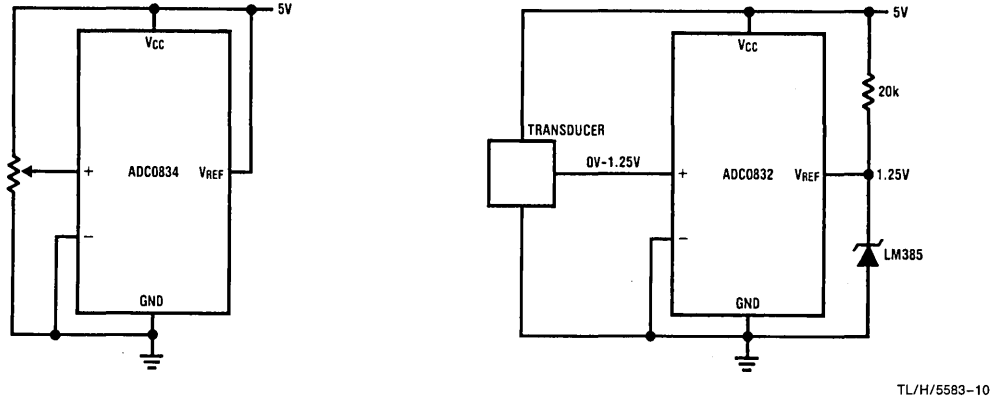
3.0 REFERENCE CONSIDERATIONS

The voltage applied to the reference input to these converters defines the voltage span of the analog input (the difference between $V_{IN(MAX)}$ and $V_{IN(MIN)}$) over which the 256 possible output codes apply. The devices can be used in either ratiometric applications or in systems requiring absolute accuracy. The reference pin must be connected to a voltage source capable of driving the reference input resistance of typically 2.4 kΩ. This pin is the top of a resistor divider string used for the successive approximation conversion.

In a ratiometric system, the analog input voltage is proportional to the voltage used for the A/D reference. This voltage is typically the system power supply, so the V_{REF} pin can be tied to V_{CC} (done internally on the ADC0832). This technique relaxes the stability requirements of the system reference as the analog input and A/D reference move together maintaining the same output code for a given input condition.

For absolute accuracy, where the analog input varies between very specific voltage limits, the reference pin can be biased with a time and temperature stable voltage source. The LM385 and LM336 reference diodes are good low current devices to use with these converters.

The maximum value of the reference is limited to the V_{CC} supply voltage. The minimum value, however, can be quite small (see Typical Performance Characteristics) to allow direct conversions of transducer outputs providing less than a 5V output span. Particular care must be taken with regard to noise pickup, circuit layout and system error voltage sources when operating with a reduced span due to the increased sensitivity of the converter (1 LSB equals V_{REF}/256).



a) Ratiometric

b) Absolute with a Reduced Span

FIGURE 2. Reference Examples

Functional Description (Continued)

4.0 THE ANALOG INPUTS

The most important feature of these converters is that they can be located right at the analog signal source and through just a few wires can communicate with a controlling processor with a highly noise immune serial bit stream. This in itself greatly minimizes circuitry to maintain analog signal accuracy which otherwise is most susceptible to noise pickup. However, a few words are in order with regard to the analog inputs should the input be noisy to begin with or possibly riding on a large common-mode voltage.

The differential input of these converters actually reduces the effects of common-mode input noise, a signal common to both selected "+" and "−" inputs for a conversion (60 Hz is most typical). The time interval between sampling the "+" input and then the "−" input is ½ of a clock period. The change in the common-mode voltage during this short time interval can cause conversion errors. For a sinusoidal common-mode signal this error is:

$$V_{error}(max) = V_{peak}(2\pi f_{CM}) \left(\frac{0.5}{f_{CLK}} \right)$$

where f_{CM} is the frequency of the common-mode signal,

V_{peak} is its peak value

and f_{CLK}, is the A/D clock frequency.

For a 60 Hz common-mode signal to generate a ¼ LSB error (≈5 mV) with the converter running at 250 kHz, its peak value would have to be 6.63V which would be larger than allowed as it exceeds the maximum analog input limits.

Due to the sampling nature of the analog inputs short spikes of current enter the "+" input and exit the "−" input at the clock edges during the actual conversion. These currents decay rapidly and do not cause errors as the internal comparator is strobed at the end of a clock period. Bypass capacitors at the inputs will average these currents and cause an effective DC current to flow through the output resistance of the analog signal source. Bypass capacitors should not be used if the source resistance is greater than 1 kΩ.

This source resistance limitation is important with regard to the DC leakage currents of input multiplexer as well. The worst-case leakage current of ±1 μA over temperature will create a 1 mV input error with a 1 kΩ source resistance. An op amp RC active low pass filter can provide both impedance buffering and noise filtering should a high impedance signal source be required.

5.0 OPTIONAL ADJUSTMENTS

5.1 Zero Error

The zero of the A/D does not require adjustment. If the minimum analog input voltage value, $V_{IN(MIN)}$, is not ground, a zero offset can be done. The converter can be made to output 0000 0000 digital code for this minimum input voltage by biasing any V_{IN} (−) input at this $V_{IN(MIN)}$ value. This utilizes the differential mode operation of the A/D.

The zero error of the A/D converter relates to the location of the first riser of the transfer function and can be measured by grounding the $V_{IN}(-)$ input and applying a small magnitude positive voltage to the $V_{IN}(+)$ input. Zero error is the difference between the actual DC input voltage which is necessary to just cause an output digital code transition from 0000 0000 to 0000 0001 and the ideal ½ LSB value (½ LSB = 9.8 mV for $V_{REF} = 5.000\ V_{DC}$).

5.2 Full-Scale

The full-scale adjustment can be made by applying a differential input voltage which is 1 ½ LSB down from the desired analog full-scale voltage range and then adjusting the magnitude of the V_{REF} input or V_{CC} for a digital output code which is just changing from 1111 1110 to 1111 1111.

5.3 Adjusting for an Arbitrary Analog Input Voltage Range

If the analog zero voltage of the A/D is shifted away from ground (for example, to accommodate an analog input signal which does not go to ground), this new zero reference should be properly adjusted first. A V_{IN} (+) voltage which equals this desired zero reference plus ½ LSB (where the LSB is calculated for the desired analog span, 1 LSB = analog span/256) is applied to selected "+" input and the zero reference voltage at the corresponding "−" input should then be adjusted to just obtain the 00_{HEX} to 01_{HEX} code transition.

The full-scale adjustment should be made [with the proper $V_{IN}(-)$ voltage applied] by forcing a voltage to the $V_{IN}(+)$ input which is given by:

$$V_{IN}(+) \text{ fs adj} = V_{MAX} - 1.5 \left[\frac{(V_{MAX} - V_{MIN})}{256} \right]$$

where:

V_{MAX} = the high end of the analog input range

and

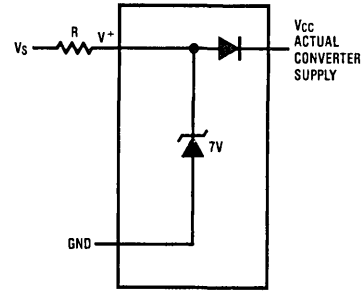
V_{MIN} = the low end (the offset zero) of the analog range.

(Both are ground referenced.)

The V_{REF} (or V_{CC}) voltage is then adjusted to provide a code change from FE_{HEX} to FF_{HEX}. This completes the adjustment procedure.

6.0 POWER SUPPLY

A unique feature of the ADC0838 and ADC0834 is the inclusion of a zener diode connected from the V^+ terminal to ground which also connects to the V_{CC} terminal (which is the actual converter supply) through a silicon diode, as shown in *Figure 3*. (See Note 3)



TL/H/5583–11

FIGURE 3. An On-Chip Shunt Regulator Diode

Functional Description (Continued)

This zener is intended for use as a shunt voltage regulator to eliminate the need for any additional regulating components. This is most desirable if the converter is to be remotely located from the system power source. *Figures 4 and 5* illustrate two useful applications of this on-board zener when an external transistor can be afforded.

An important use of the interconnecting diode between V^+ and V_{CC} is shown in *Figures 6 and 7*. Here, this diode is used as a rectifier to allow the V_{CC} supply for the converter to be derived from the clock. The low current requirements of the A/D and the relatively high clock frequencies used (typically in the range of 10k–400 kHz) allows using the small value filter capacitor shown to keep the ripple on the V_{CC} line to well under 1/4 of an LSB. The shunt zener regulator can also be used in this mode. This requires a clock voltage swing which is in excess of V_Z. A current limit for the zener is needed, either built into the clock generator or a resistor can be used from the CLK pin to the V^+ pin.

Applications

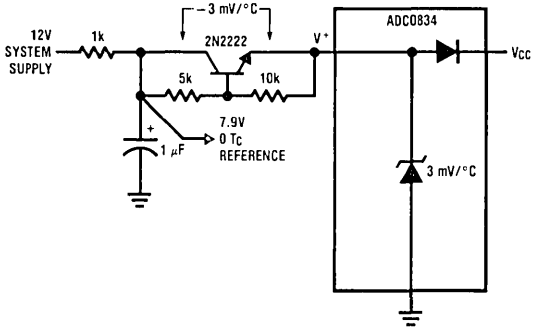


FIGURE 4. Operating with a Temperature Compensated Reference

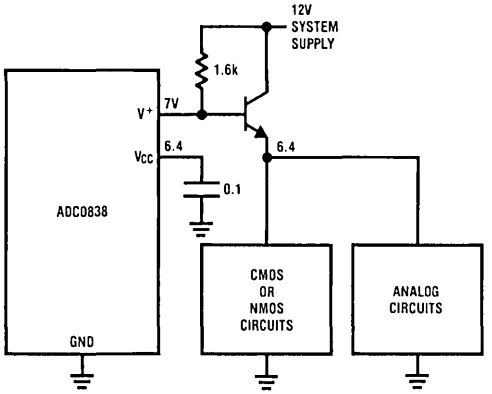


FIGURE 5. Using the A/D as the System Supply Regulator

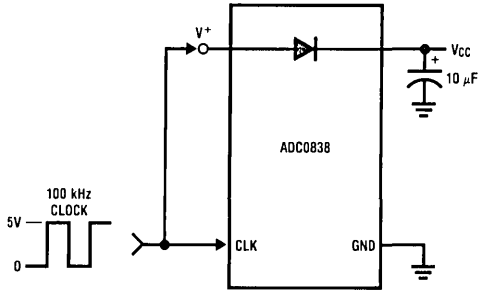
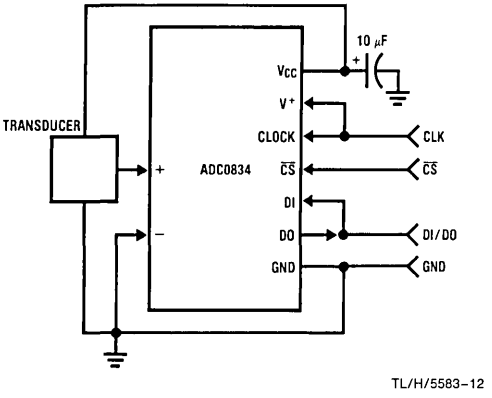


FIGURE 6. Generating V_{CC} from the Converter Clock

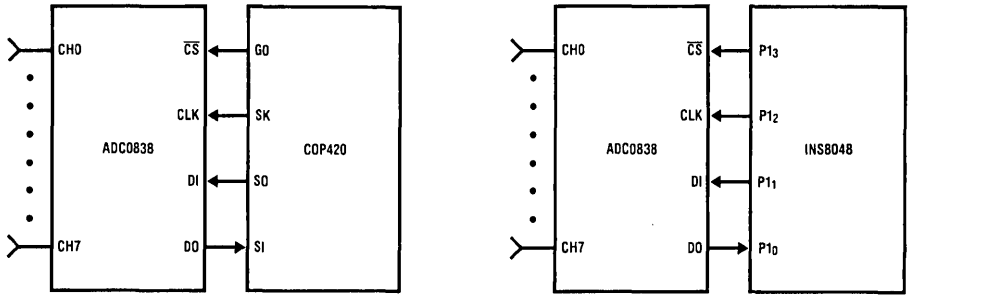


TL/H/5583–12

FIGURE 7. Remote Sensing—Clock and Power on 1 Wire

Applications (Continued)

Digital Link and Sample Controlling Software for the Serially Oriented COP420 and the Bit Programmable I/O INS8048



TL/H/5583–13

COP CODING EXAMPLE

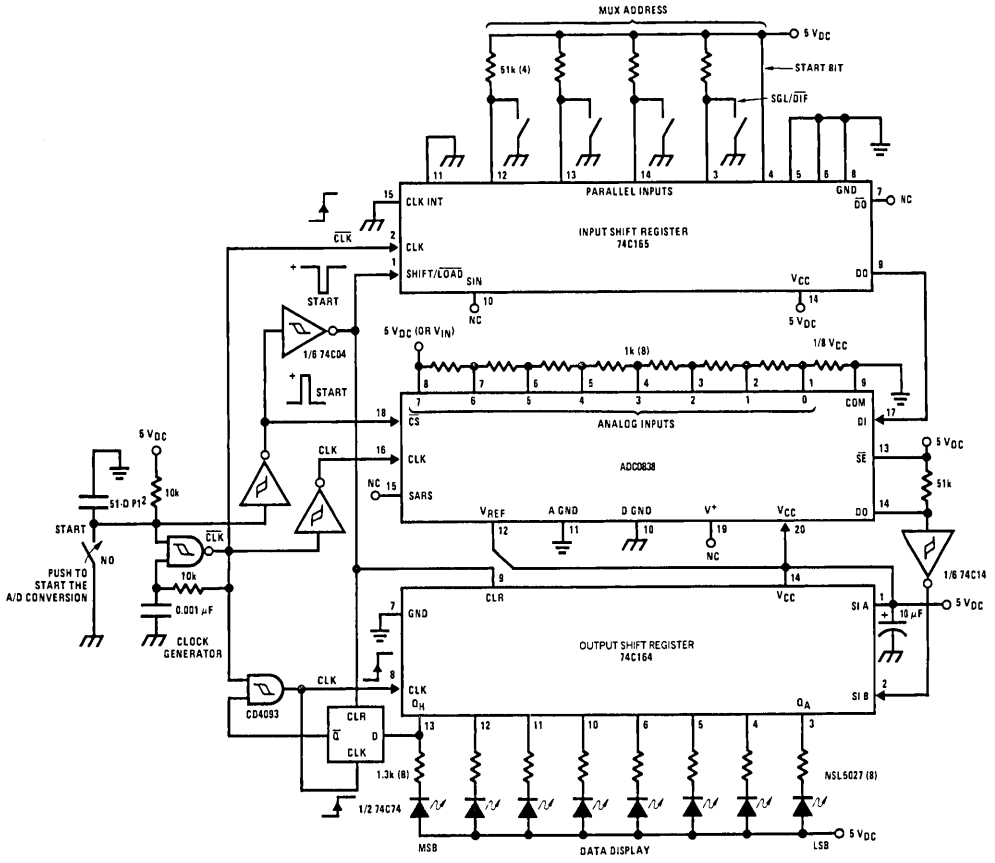
Mnemonic	Instruction
LEI	ENABLES SIO's INPUT AND OUTPUT
SC	C = 1
OGI	G0 = 0 ($\overline{CS}$ = 0)
CLR A	CLEARS ACCUMULATOR
AISC 1	LOADS ACCUMULATOR WITH 1
XAS	EXCHANGES SIO WITH ACCUMULATOR AND STARTS SK CLOCK
LDD	LOADS MUX ADDRESS FROM RAM INTO ACCUMULATOR
NOP	—
XAS	LOADS MUX ADDRESS FROM ACCUMULATOR TO SIO REGISTER
↑	
8 INSTRUCTIONS	
↓	
XAS	READS HIGH ORDER NIBBLE (4 BITS) INTO ACCUMULATOR
XIS	PUTS HIGH ORDER NIBBLE INTO RAM
CLR A	CLEARS ACCUMULATOR
RC	C = 0
XAS	READS LOW ORDER NIBBLE INTO ACCUMULATOR AND STOPS SK
XIS	PUTS LOW ORDER NIBBLE INTO RAM
OGI	G0 = 1 ($\overline{CS}$ = 1)
LEI	DISABLES SIO's INPUT AND OUTPUT

8048 CODING EXAMPLE

	Mnemonic		Instruction
START:	ANL	P1, #0F7H	;SELECT A/D ($\overline{CS}$ = 0)
	MOV	B, #5	;BIT COUNTER ← 5
	MOV	A, #ADDR	;A ← MUX ADDRESS
LOOP 1:	RRC	A	;CY ← ADDRESS BIT
	JC	ONE	;TEST BIT
			;BIT = 0
ZERO:	ANL	P1, #0FEH	;DI ← 0
	JMP	CONT	;CONTINUE
			;BIT = 1
ONE:	ORL	P1, #1	;DI ← 1
CONT:	CALL	PULSE	;PULSE SK 0 → 1 → 0
	DJNZ	B, LOOP 1	;CONTINUE UNTIL DONE
	CALL	PULSE	;EXTRA CLOCK FOR SYNC
	MOV	B, #8	;BIT COUNTER ← 8
LOOP 2:	CALL	PULSE	;PULSE SK 0 → 1 → 0
	IN	A, P1	;CY ← DO
	RRC	A	
	RRC	A	
	MOV	A, C	;A ← RESULT
	RLC	A	;A(0) ← BIT AND SHIFT
	MOV	C, A	;C ← RESULT
	DJNZ	B, LOOP 2	;CONTINUE UNTIL DONE
RETR			
			;PULSE SUBROUTINE
PULSE:	ORL	P1, #04	;SK ← 1
	NOP		;DELAY
	ANL	P1, #0FBH	;SK ← 0
	RET		

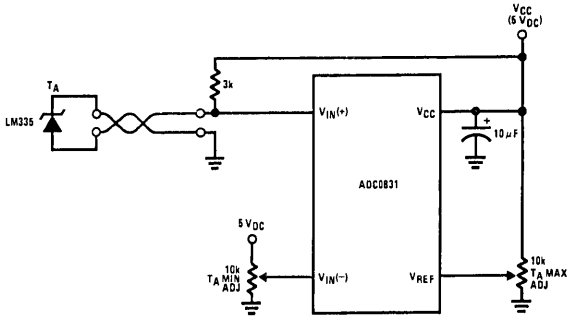
Applications (Continued)

A "Stand-Alone" Hook-Up for ADC0838 Evaluation



*Pinouts shown for ADC0838.
For all other products tie to pin functions as shown.

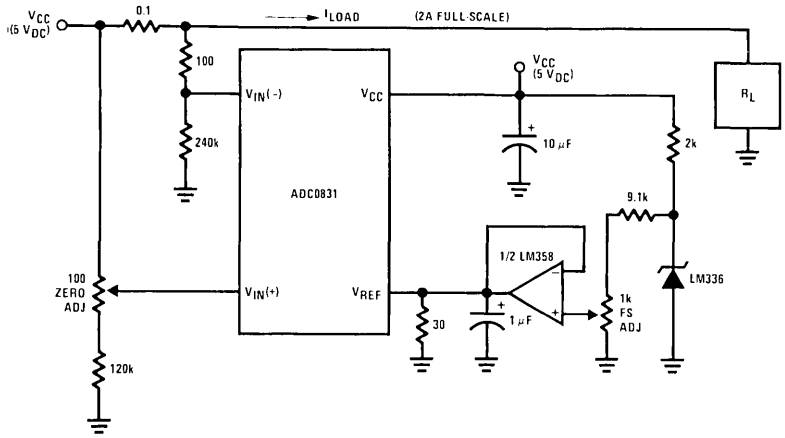
Low-Cost Remote Temperature Sensor



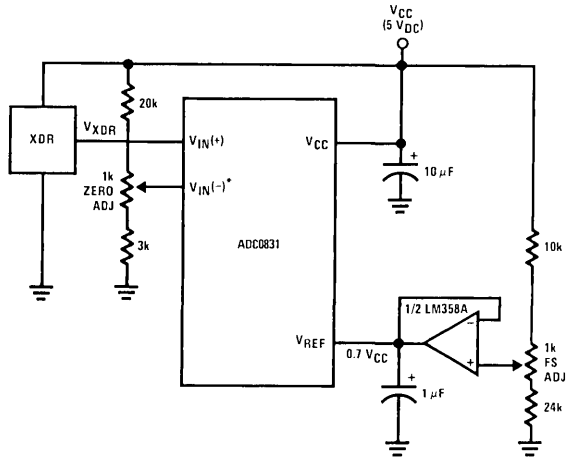
TL/H/5583–14

Applications (Continued)

Digitizing a Current Flow



Operating with Ratiometric Transducers

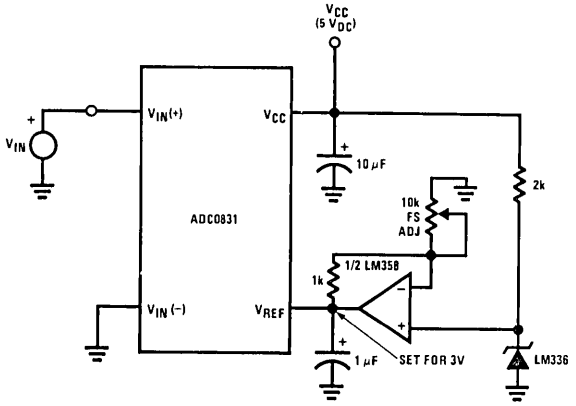


TL/H/5583–15

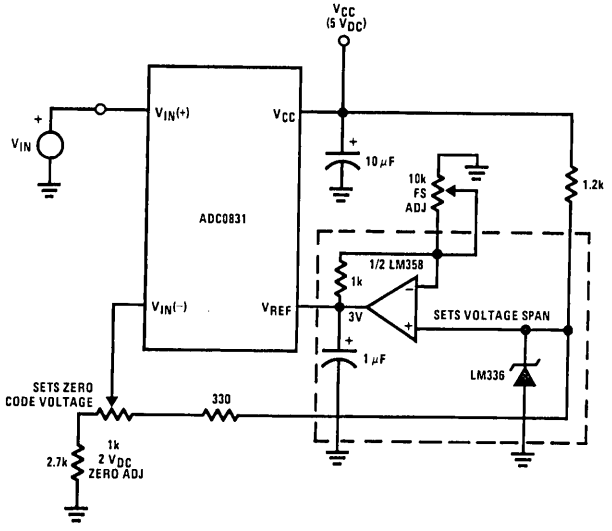
* $V_{IN}(-) = 0.15\ V_{CC}$
15% of $V_{CC} \leq V_{XDR} \leq 85\%$ of V_{CC}

Applications (Continued)

Span Adjust: $0V \leq V_{IN} \leq 3V$



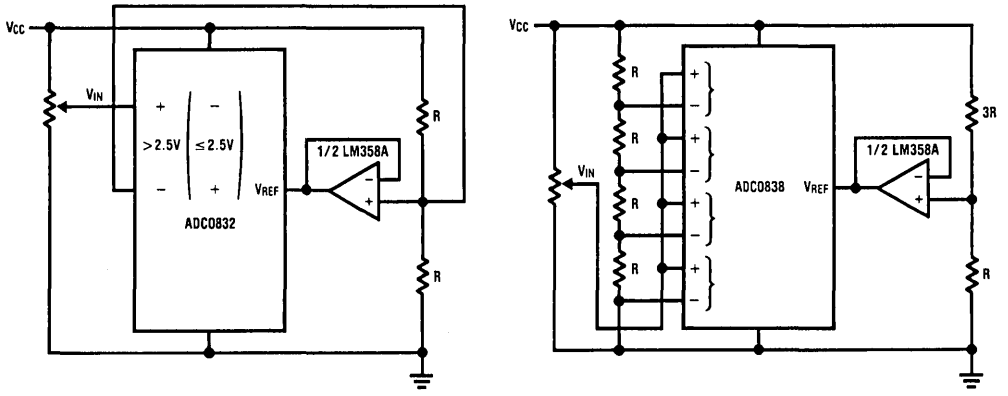
Zero-Shift and Span Adjust: $2V \leq V_{IN} \leq 5V$



TL/H/5583–16

Applications (Continued)

Obtaining Higher Resolution

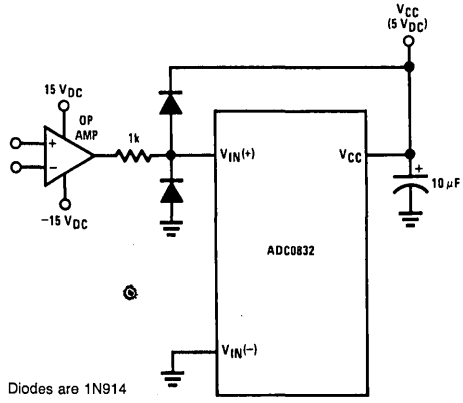


TL/H/5583–17

Controller performs a routine to determine which input polarity (9-bit example) or which channel pair (10-bit example) provides a non-zero output code. This information provides the extra bits.

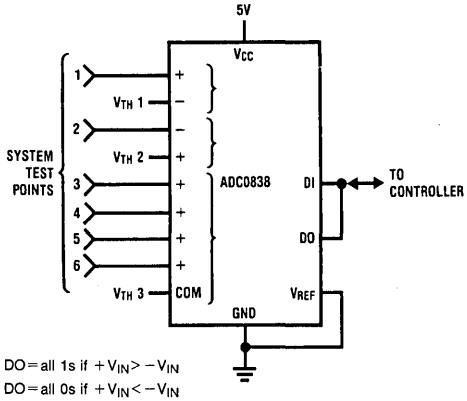
a) 9-Bit A/D **b)10-Bit A/D**

Protecting the Input



Diodes are 1N914

High Accuracy Comparators



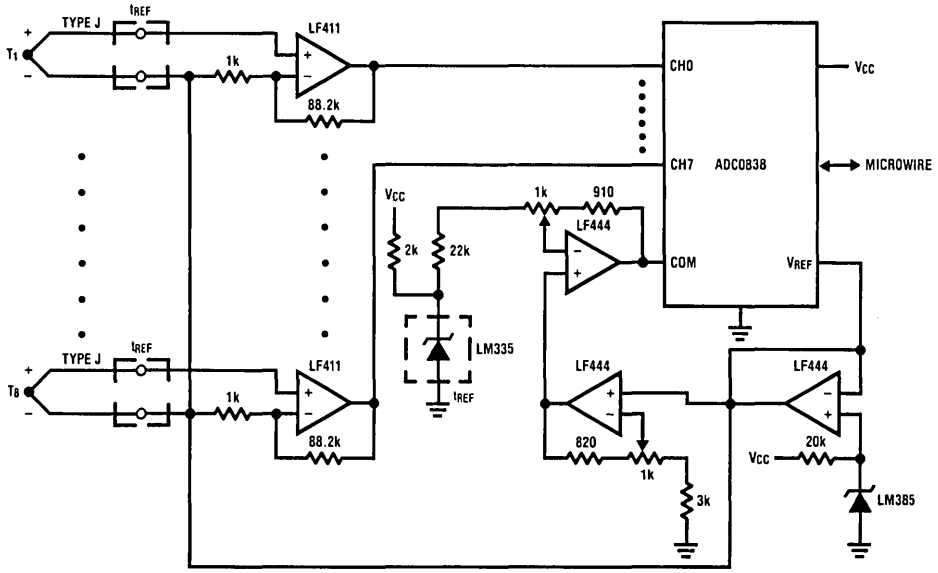
DO = all 1s if $+V_{IN} > -V_{IN}$

DO = all 0s if $+V_{IN} < -V_{IN}$

TL/H/5583–18

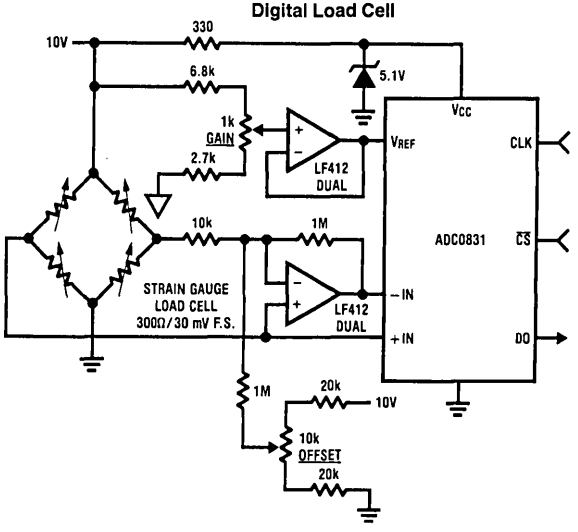
Applications (Continued)

Convert 8 Thermocouples with only One Cold-Junction Compensator



Uses the pseudo-differential mode to keep the differential inputs constant with changes in reference temperature (T_{REF}).

Digital Load Cell

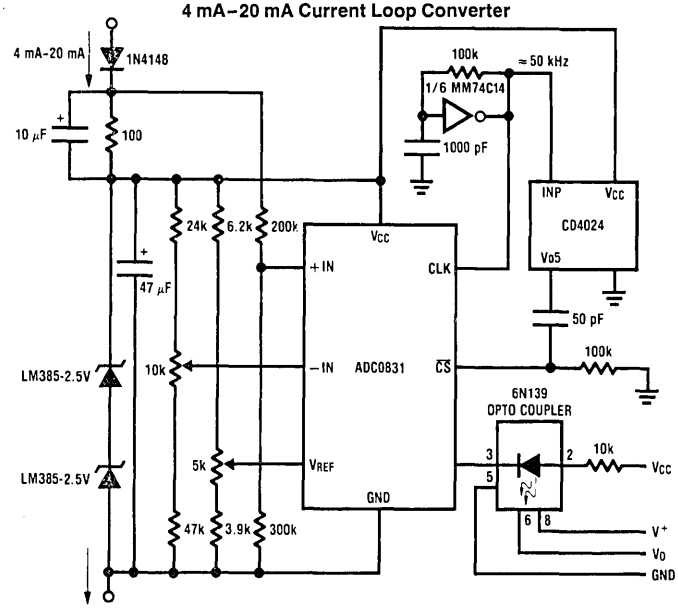


TL/H/5583–19

- Uses one more wire than load cell itself
- Two mini-DIPs could be mounted inside load cell for digital output transducer
- Electronic offset and gain trims relax mechanical specs for gauge factor and offset
- Low level cell output is converted immediately for high noise immunity

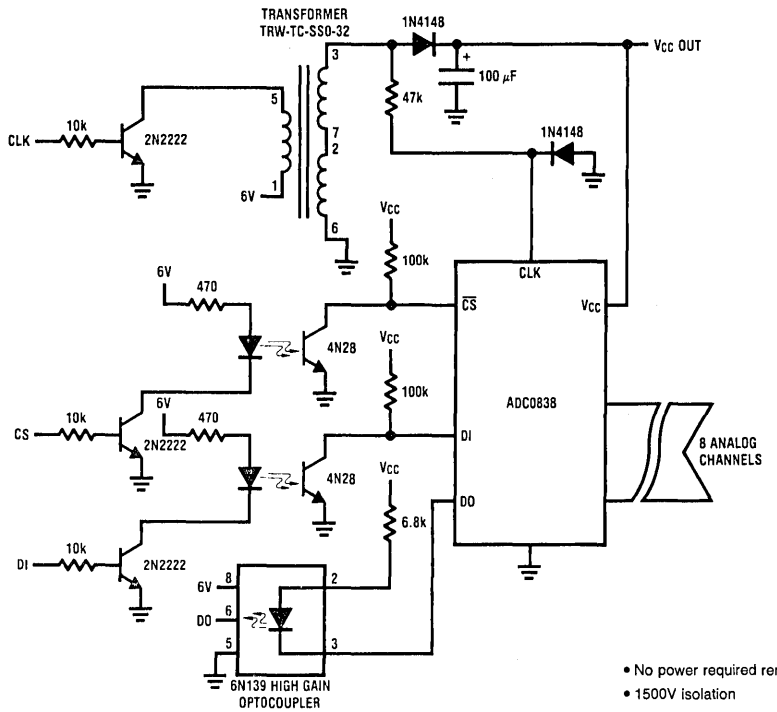
Applications (Continued)

4 mA–20 mA Current Loop Converter



- All power supplied by loop
- 1500V isolation at output

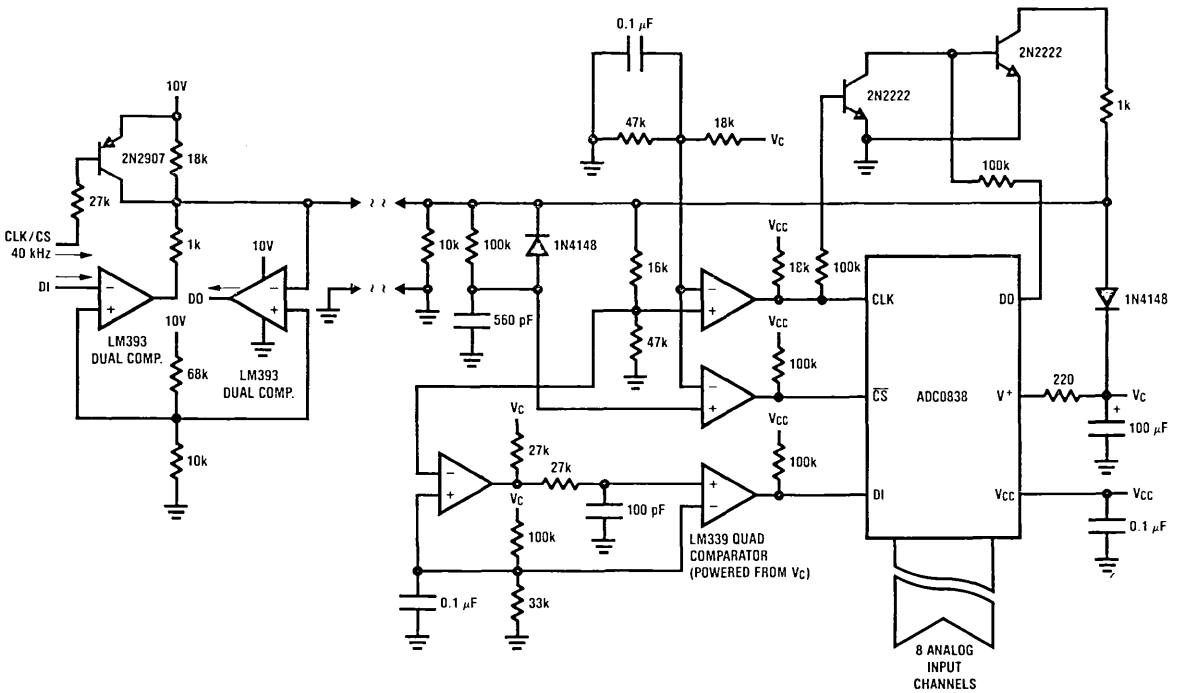
Isolated Data Converter



TL/H/5583–20

- No power required remotely
- 1500V isolation

Two Wire Interface for 8 Channels

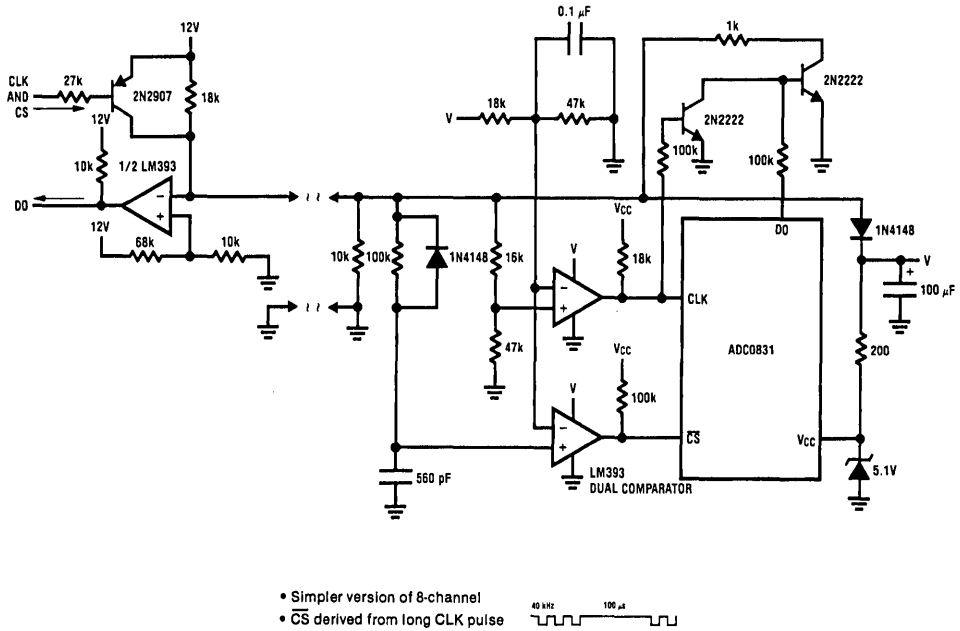


- No additional connections
- $\overline{CS}$ derived from extended high on CLK line > 100 μs
- Timing arranged for 40 kHz, could be changed up or down by component change
- 10% CLK frequency change without component change OK

TL/H/5583–21

Applications (Continued)

Two Wire 1-Channel Interface



- Simpler version of 8-channel
- $\overline{CS}$ derived from long CLK pulse

TL/H/5583–22

Ordering Information

Part Number	Analog Input Channels	Total Unadjusted Error	Package	Temperature Range
ADC0831BJ	1	±½	Hermetic (J)	−55°C to +125°C
ADC0831BCJ			Hermetic (J)	−40°C to +85°C
ADC0831BCN (COP431BN)			Molded (N)	−0°C to +70°C
ADC0831CCJ		±1	Hermetic (J)	−40°C to +85°C
ADC0831CCN (COP431CN)			Molded (N)	−0°C to +70°C
ADC0832BJ	2	±½	Hermetic (J)	−55°C to +125°C
ADC0832BCJ			Hermetic (J)	−40°C to +85°C
ADC0832BCN (COP432BN)			Molded (N)	−0°C to +70°C
ADC0832CCJ		±1	Hermetic (J)	−40°C to +85°C
ADC0832CCN (COP432CN)			Molded (N)	−0°C to +70°C
ADC0834BJ	4	±½	Hermetic (J)	−55°C to +125°C
ADC0834BCJ			Hermetic (J)	−40°C to +85°C
ADC0834BCN (COP434BN)			Molded (N)	−0°C to +70°C
ADC0834CCJ		±1	Hermetic (J)	−40°C to +85°C
ADC0834CCN (COP434CN)			Molded (N)	−0°C to +70°C
ADC0838BJ	8	±½	Hermetic (J)	−55°C to +125°C
ADC0838BCJ			Hermetic (J)	−40°C to +85°C
ADC0838BCN (COP438BN)			Molded (N)	−0°C to +70°C
ADC0838CCJ		±1	Hermetic (J)	−40°C to +85°C
ADC0838CCN (COP438CN)			Molded (N)	−0°C to +70°C

See NS Packages J08A, J14A, J20A, N08E, N14A, N20A

National Semiconductor Corporation

COP452L/COP352L Frequency Generator and Counter

General Description

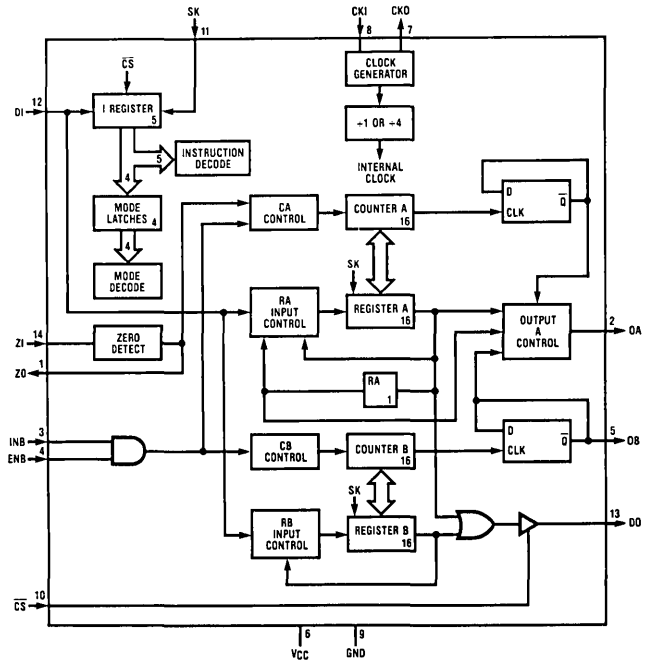
The COP452L and COP352L are peripheral members of the COPS™ family fabricated using N-channel silicon gate MOS technology. Containing two independent 16-bit counter/register pairs, they are well suited to a wide variety of tasks involving the measurement and/or generation of times and/or frequencies. Included in the features are multiple tone generation, precise duty cycle generation, event counting, waveform measurement, frequency bursts, delays, and "white noise" generation. An on-chip zero crossing detector can trigger a pulse with a programmed delay and duration. The COP352L is the extended temperature version of the COP452L. The COP352L is the functional equivalent of the COP452L.

The COP452L series peripheral devices can perform numerous functions that a microcontroller alone cannot perform. They can execute one or more complex tasks, attaining higher accuracies over a broader frequency range than a microcontroller alone. These devices remove repetitive yet demanding counting, timing, and frequency related functions from the microcontroller, thereby freeing it to perform other tasks or allowing the use of a simpler microcontroller in the system.

Features

- Unburdens microcontroller by performing "mundane" tasks
- Wider range and greater accuracy than microcontroller alone
- Generates frequencies, frequency bursts, and complex waveforms
- Measures waveform duty cycle
- Two independent pulse/event counters
- True zero crossing detector triggers output pulse
- White noise generator
- Compatible with all COP400 microcontrollers
- MICROWIRE™ compatible serial I/O
- 14-pin package
- Single supply operation
  (4.5V–6.3V, COP452L; 4.5V–5.5V, COP352L)
- Low cost
- Crystal or external clock
  (25 kHz to 2.1 MHz, COP452L)
  (64 kHz to 2.1 MHz, COP352L)
- TTL compatible

Block Diagram



TL/DD/6155–1

FIGURE 1

COP452L
Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin (except ZI) relative to GND	−0.5V to +7.0V
Voltage at Pin ZI relative to GND	−0.8 to +10V
Sink Current, Output OA	15 mA
Sink Current, All Other Outputs	5 mA
Total Sink Current	35 mA
Source Current, Outputs OA, OB	5 mA
Source Current, All Other Outputs	1 mA
Total Source Current	10 mA
Ambient Operating Temperature	0°C to 70°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C
Power Dissipation	0.5W at 25°C 0.2W at 70°C

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $0°C \le T_A + 70°C$, $4.5V \le V_{CC} \le 6.3V$ (COP452L), unless otherwise specified

Parameter	Conditions	Min	Max	Units
Operating Voltage (V_{CC})		4.5	6.3	V
Operating Supply Current	All Outputs Open		14	mA
Input Voltage Levels				
CKI Input Levels	V_{CC} = Max.	3.0		V
Logic High (V_{IH})	V_{CC} = 5.0V ±5%	2.0		V
Logic Low (V_{IL})			0.4	V
DI, INB, ENB, SK, $\overline{CS}$				
Logic High	V_{CC} = Max.	3.0		V
Logic High (V_{IH})	V_{CC} = 5.0V ±5%	2.0		V
Logic Low (V_{IL})			0.8	V
ZI Input Voltage		−0.8	+10	V
Impedance to GND at ZI		−1.6	7.8	kΩ
ZI Offset Voltage	(Note 1)		150	mV
Output Voltage Levels				
TTL Operation	V_{CC} = 5.0V ±5%			
Logic High (V_{OH})	I_{OH} = 100 μA	2.4		V
Logic Low (V_{OL})	I_{OL} = −1.6 mA		0.4	V
Maximum Allowable Output Current Levels				
Sink Current				
OA	(Note 2)		15	mA
All Other Outputs	(Note 2)		5.0	mA
Total Sink Current	(Note 3)		35	mA
Source Current				
OA, OB	(Note 2)		−5.0	mA
All Other Outputs	(Note 2)		−1.0	mA
Total Source Current	(Note 3)		−10	mA

Note 1: ZI offset voltage is the absolute value of the difference between the voltage at ZI and ground (pin 9) that will cause the zero detect circuit output to change state. This is the maximum value which takes into account the worst case effects of process, temperature, voltage, and gain variation.

Note 2: The maximum current for the specified pin must be limited to this value or less.

Note 3: The total current in the device must be limited to this value or less.

COP452L
AC Electrical Characteristics $0°C \le T_A \le 70°C$, $4.5V \le V_{CC} \le 6.3V$ unless otherwise specified

Parameter		Conditions	Min	Max	Units
CKI Input Frequency (f_{IN})		÷4 Mode	100	2100	kHz
		÷1 Mode	25	525	kHz
Duty Cycle		÷4	30	55	%
÷1		÷1	45	55	%
Rise Time (t_r)		f_{IN} = 2.1 MHz		50	ns
Fall Time (t_f)		f_{IN} = 2.1 MHz		40	ns
SK Input Frequency			25	250	kHz
SK Duty Cycle			30	70	%
Internal Clock Frequency (f_I)			25	525	kHz
Internal Count Rate			0	$f_I/2$	Hz
Output Frequency			$f_I/131072$	$f_I/2$	Hz
Inputs					
DI	t_{SETUP}		800		ns
	t_{HOLD}		1.0		µs
Outputs					
CKO	t_{pd1}	C_L = 50 pF		0.2	µs
	t_{pd0}			0.2	µs
ZO	t_{pd1}	ZI = Sine Wave *(Figure 4)*		0.7	µs
	t_{pd0}			0.6	µs
DO	t_{pd1}	C_L = 50 pF		1.0	µs
	t_{pd0}			0.6	µs
OA	t_{pd1}	C_L = 50 pF V_{OUT} = 1.5V		0.7	µs
	t_{PD0}			0.8	µs
OB	t_{pd1}			1.0	µs
	t_{pd0}			0.4	µs

COP352L
Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin (except ZI) relative to GND	−0.5V to +7.0V
Voltage at Pin ZI relative to GND	−0.8V to +10V
Sink Current, Output OA	15 mA
Sink Current, All Other Outputs	5 mA
Total Sink Current	35 mA
Source Current, Outputs OA, OB	5 mA
Source Current, All Other Outputs	1 mA
Total Source Current	10 mA
Ambient Operating Temperature	−40°C to +85°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C
Power Dissipation	0.5W at 25°C 0.125W at 85°C

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $-40°C \le T_A \le 85°C$, $4.5V \le V_{CC} \le 5.5V$ unless otherwise specified

Parameter	Conditions	Min	Max	Units
Operating Voltage (V_{CC})		4.5	5.5	V
Operating Supply Current	All Outputs Open		16	mA
Input Voltage Levels				
CKI Input Levels	V_{CC} = Max.	3.0		V
Logic High (V_{IH})	V_{CC} = 5.0V ±5%	2.2		V
Logic Low (V_{IL})			0.3	V
DI, INB, ENB, SK, $\overline{CS}$				
Logic High	V_{CC} = Max.	3.0		V
Logic High (V_{IH})	V_{CC} = 5.0V ±5%	2.2		V
Logic Low (V_{IL})			0.6	V
ZI Input Voltage		−0.8	+10	V
Impedance to GND at ZI		1.6	7.8	kΩ
ZI Offset Voltage	(Note 1)		150	mV
Output Voltage Levels				
TTL Operation	V_{CC} = 5.0V ±5%			
Logic High (V_{OH})	I_{OH} = 100 μA	2.4		V
Logic Low (V_{OL})	I_{OL} = −1.6 mA		0.4	V
Maximum Allowable Output Current Levels				
Sink Current				
OA	(Note 2)		15	mA
All Other Outputs	(Note 2)		5.0	mA
Total Sink Current	(Note 3)		35	mA
Source Current				
OA, OB	(Note 2)		−5.0	mA
All Other Outputs	(Note 2)		−1.0	mA
Total Source Current	(Note 3)		−10	mA

Note 1: ZI offset voltage is the absolute value of the difference between the voltage at ZI and ground (pin 9) that will cause the zero detect circuit output to change state. This is the maximum value which takes into account the worst case effects of process, temperature, voltage, and gain variation.

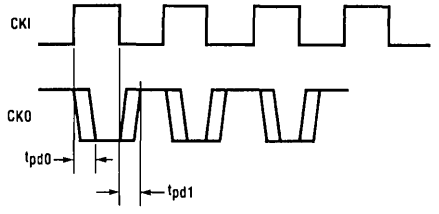
Note 2: The maximum current for the specified pin must be limited to this value or less.

Note 3: The total current in the device must be limited to this value or less.

COP325L
AC Electrical Characteristics $-40°C \le T_A \le +85°C$, $4.5V \le V_{CC} \le 5.5V$ unless otherwise specified

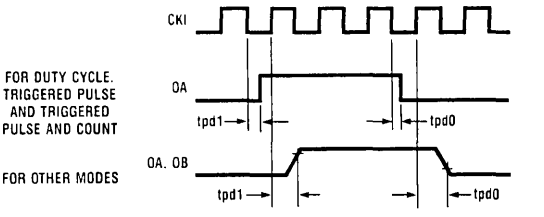
Parameter		Conditions	Min	Max	Units
CKI Input Frequency (f_{IN})		÷4 Mode	256	2100	kHz
		÷1 Mode	64	525	kHz
Duty Cycle		÷4	35	55	%
		÷1	50	55	%
Rise Time (t_r)		f_{IN} = 2.1 MHz		50	ns
Fall Time (t_f)		f_{IN} = 2.1 MHz		40	ns
SK Input Frequency			25	250	kHz
SK Duty Cycle			30	70	%
Internal Clock Frequency (f_I)			64	525	kHz
Internal Count Rate			0	$f_I/2$	Hz
Output Frequency			$f_I/131072$	$f_I/2$	Hz
Inputs					
DI	t_{SETUP}		800		ns
	t_{HOLD}		1.0		μs
Outputs					
CKO	t_{pd1}	C_L = 50 pF		0.25	μs
	t_{pd0}			0.25	μs
ZO	t_{pd1}	ZI = sine wave *(Figure 4)*		0.8	μs
	t_{pd0}			0.7	μs
DO	t_{pd1}	C_L = 50 pF		1.1	μs
	t_{pd0}			0.7	μs
OA	t_{pd1}	C_L = 50 pF V_{OUT} = 1.5V		0.7	μs
	t_{pd0}			0.8	μs
OB	t_{pd1}			1.0	μs
	t_{pd0}			0.4	μs

Timing Diagrams



TL/DD/6155-2

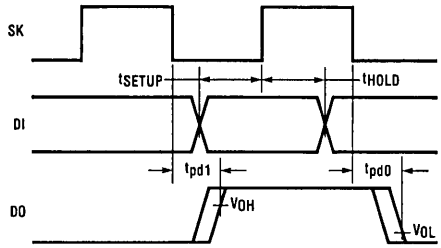
FIGURE 2a. CKO Output Timing



TL/DD/6155-3

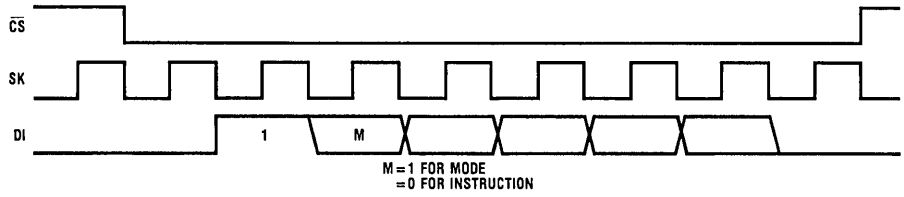
FIGURE 2b. OA and OB Output Timing

Timing Diagrams (Continued)



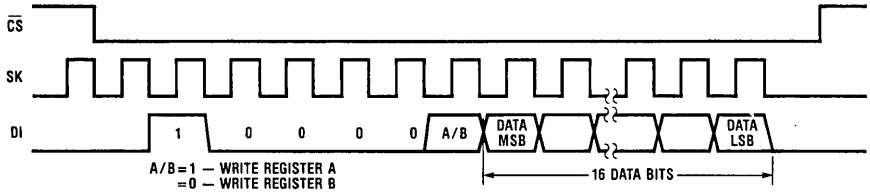
TL/DD/6155-4

FIGURE 3a. Synchronous Data Timing



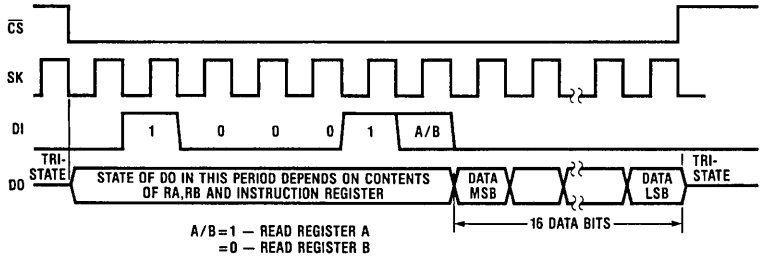
TL/DD/6155-5

FIGURE 3b. Instruction Timing (Except Read/Write)



TL/DD/6155-6

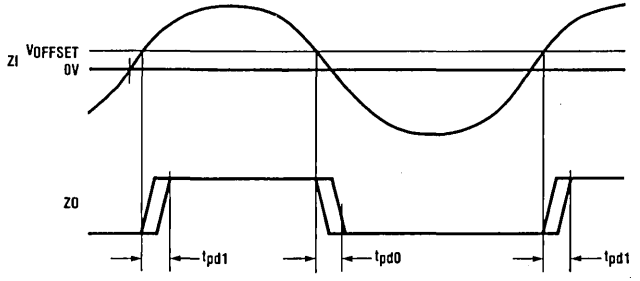
FIGURE 3c. Write Instruction Timing



TL/DD/6155-7

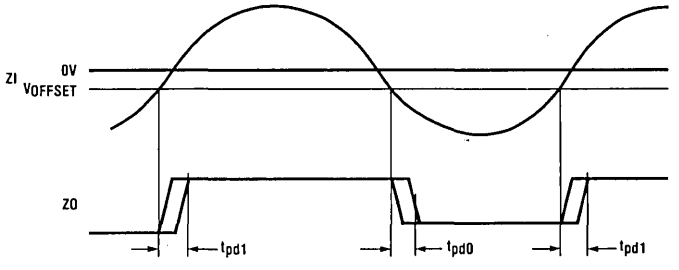
FIGURE 3d. Read Instruction Timing

Timing Diagrams (Continued)



TL/DD/6155-8

FIGURE 4a. ZO Timing, $V_{OFFSET} > 0V$



TL/DD/6155-9

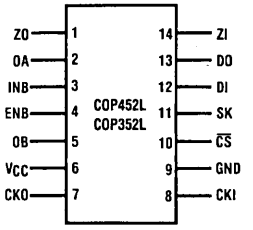
FIGURE 4b. ZO Timing, $V_{OFFSET} < 0V$

Pin Descriptions

Pin	Description	Pin	Description
ZO	Zero Cross Output Signal	CKI	Crystal Oscillator Input
OA	Counter A, Logic Controlled Output	GND	Ground
INB	Counter B, External Input	$\overline{CS}$	Chip Select
ENB	Enable for INB	SK	Serial Data I/O Clock Input
OB	Counter B Output	DI	Serial Data Input
V_{CC}	Power Supply	DO	Serial Data Output
CKO	Crystal Oscillator Output	ZI	AC Waveform Input, Counter A External Input

Connection Diagram

Dual-In-Line Package



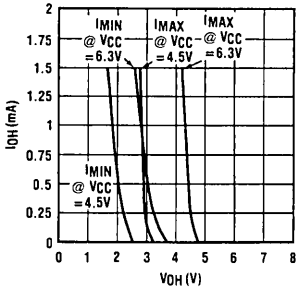
TL/DD/6155-10

FIGURE 5. Pin Connection Diagram

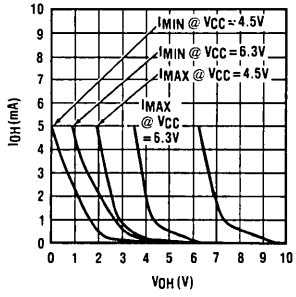
Order Number COP452D, COP352D, COP452N or COP352N
See NS Package Number D14D or N14A

Typical Performance Characteristics

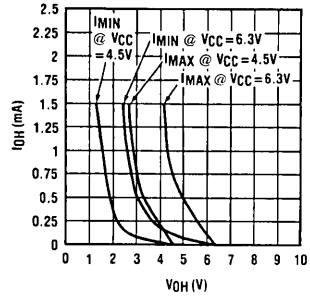
DO Source Current



OA, OB Source Current

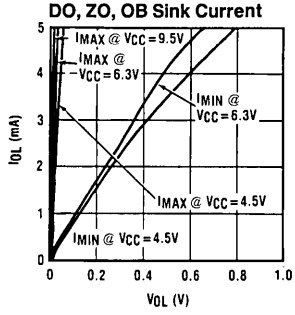


ZO Source Current

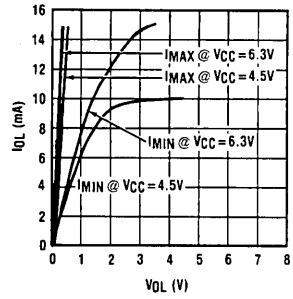


TL/DD/6155-11

DO, ZO, OB Sink Current



OA Sink Current

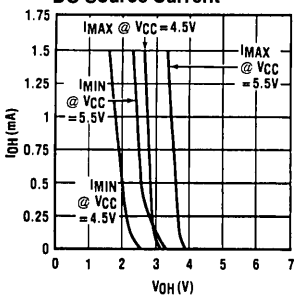


TL/DD/6155-12

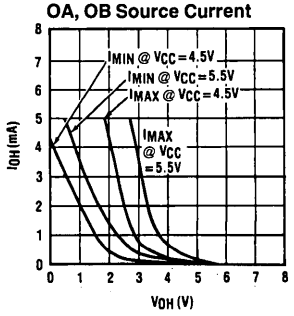
TL/DD/6155-13

FIGURE 6. COP452L

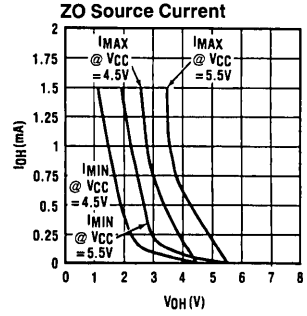
DO Source Current



OA, OB Source Current

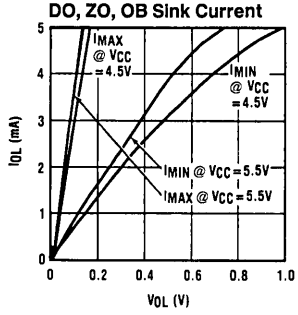


ZO Source Current

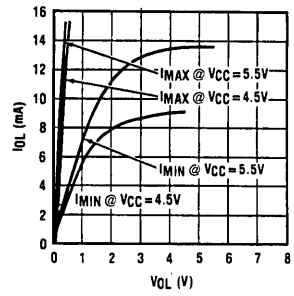


TL/DD/6155-14

DO, ZO, OB Sink Current



OA Sink Current



TL/DD/6155-15

TL/DD/6155-16

FIGURE 7. COP352L

Functional Description

The COP452L and COP352L are functionally identical devices. They differ only in V_{CC} range and/or operating temperature range, and certain electrical parameters associated with those temperature and voltage ranges. The following information will refer only to the COP452L. All the information, however, applies equally to the COP452L and COP352L.

INSTRUCTION SET AND OPERATING MODES

The COP452L has ten instructions and eleven operating modes as indicated in *Figure 8*. The information for the instruction or mode is sent to the COP452L via the serial interface. The MSB is always a "1" and is properly viewed as a start bit. The second MSB identifies the communication as an instruction or a mode. The lower four bits contain the command for the device.

Instruction	Opcode MSB LSB	Comments
LDRB	100000	Load register B from DI
LDRA	100001	Load register A from DI
RDRB	100010	Read register B to DO
RDRA	100011	Read register A to DO
TRCB	100100	Transfer register B to counter B
TRCA	100101	Transfer register A to counter A
TCRB	100110	Transfer counter B to register B
TCRA	100111	Transfer counter A to register A
CK1	101000	CKI divide by one
CK4	101001	CKI divide by four
LDM	11xxxx	Load mode latches

FIGURE 8a. COP452L Instruction Set

Operating Mode	Opcode MSB LSB
Reset	111111
Dual Frequency	110000
Frequency and Count	110100
Dual Count	110101
Number of Pulses	110010
Duty Cycle	110011
Waveform Measurement	110110
Triggered Pulse	110001
Triggered Pulse and Count	110111
White Noise and Frequency	111000
Gated White Noise	111001

FIGURE 8b. COP452L Operating Modes

A block diagram of the COP452L is given in *Figure 1*. Positive logic is used. The COP452L can execute ten instructions as indicated in *Figure 8a*, and has eleven operating modes. The operating mode is under user software control.

The device basically consists of two sixteen bit shift registers and two sixteen bit binary down counters organized as two register-counter pairs. In most operating modes, the two register-counter pairs are completely independent of one another. For frequency generation, both the register and counter of a given pair are utilized. The counter counts down to zero where a toggle flip-flop is toggled. Then the data in the register is loaded, automatically, to the counter and the process continues. A similar procedure is used in the duty cycle mode and number of pulses modes. For counting, the counters count the pulses at their respective inputs. There is no automatic counter-register transfer in the count modes. The counters wraparound from 0 to FFFF in the count modes. Data I/O is via the serial port and the registers. The counters are not involved in the input/output process at all.

The device requires a low chip select signal. When the device is selected ($\overline{CS}$ low) the driver on the DO pin is enabled and the device will accept data at DI on each SK pulse. When the device is deselected ($\overline{CS}$ high) the DO driver is TRI-STATE® and the I register is reset to 0. Note that chip select does not affect any other portion of the device. The mode latches are not affected. The COP452L will continue to operate in the mode specified by the user until the mode is changed by the user.

The COP452L contains a clock generator. The user may connect a crystal network to CKI and CKO or he may drive CKI from an external oscillator. Certain RC and LC networks may also be used. See the applications for further information.

The user also has control over whether the clock generator divides the CKI signal by 4 or 1. This allows the user to quickly get a 4 to 1 change in frequency output or input count rates. Alternatively, it allows the user to use a higher speed crystal or clock generator. The internal clock frequency (the frequency after the divider) must remain between the specified limits to guarantee proper operation. The state of the divider is not affected by $\overline{CS}$.

There is an internal power-on reset circuit which places the device in the Reset mode (mode latches all set to 1) and sets the clock divider to divide by four. If the CKI frequency is less than four times the minimum internal frequency the first access of the COP452L *must* be the command to set the divider to divide by 1. This command will be accepted and will be processed. Proper operation of the COP452L is not guaranteed if the internal frequency is less than the specified minimum. The power-on reset circuit does not affect the counter and registers of the COP452L.

When the COP452L is subjected to rapid power supply cycling, the internal power on reset will not function. Power must be removed for at least 20 seconds to allow restoration of internal reset circuitry. If the application requires power on-off cycles more frequently than once each 20 seconds the software reset with proper CKI divide by must be used to establish the initial state of the COP452L.

INSTRUCTION DESCRIPTION

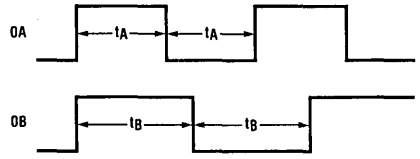
1. **Load Register (LDRA/LDRB)**—The selected register (A/B) is loaded with 16 bits of data shifted in on DI and clocked in by SK.
2. **Read Register (RDRA/RDRB)**—The data in the selected register (A/B) is shifted out serially onto DO. At the same time the data is recirculated back to the register.
3. **Load Counter (TRCA/TRCB)**—The contents of the selected register are transferred to its associated counter. (Counter A is loaded from register A; counter B is loaded from register B.) The contents of the register are unaffected.
4. **Copy Counter (TCRA/TCRB)**—The contents of the selected counter are transferred to its associated register. (Counter A loads register A; counter B loads register B.) The contents of the counter are unaffected.

Functional Description (Continued)

5. **CKI Divide by One**—The oscillator divider at the CKI input is set to divide by one. The internal frequency is therefore equal to the CKI frequency. This instruction should not be used if the CKI frequency is greater than the maximum internal frequency.
6. **CKI Divide by Four**—The oscillator divider at the CKI input is set to divide by four. The internal frequency is therefore equal to one-fourth of the CKI frequency. This instruction should not be used if the CKI frequency is less than four times the minimum internal frequency.
7. **Load Mode Latches**—The four mode latches are loaded with the lower four bits of the instruction.

MODE DESCRIPTION

1. **Reset Mode**—This mode sets OA and OB to "0". The mode latches are all set to "1". No counting occurs; the COP452L is in an idle condition. The registers and counters are not altered in any way.
2. **Dual Frequency**—Two frequencies are generated—one at output OA and one at output OB. The period of the square wave at OA is determined by the contents of register A. The period of the square wave at OB is determined by the contents of register B. In frequency generation modes, the counters count down until they reach zero. At that point the output toggles and the counters are automatically loaded from the respective registers. The counters are only loaded when they count down to zero. Therefore it may be necessary to initially load the counters. The frequency outputs at OA and OB are completely independent of one another. The respective counter inputs (INB, ZI) have no effect on the counters in this mode.



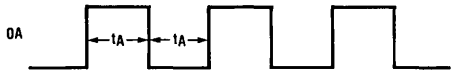
TL/DD/6155–17

$t_A = (A + 1)t$
$t_B = (B + 1)t$
$0 \le A \le 65535$; $0 \le B \le 65535$

Where: A = Contents of register A
B = Contents of register B
t = Period of internal clock
= Period of CKI oscillator (÷ mode)
= 4 × period of CKI oscillator (÷4 mode)

Period of output square wave = 2(N + 1)t
Where t is devined above
N = Contents of register
$0 \le N \le 65535$ ($0 \le N \le FFFF_{16}$)

3. **Frequency and Count**—A single frequency is output at OA. Counter B counts external pulses on INB (when ENB = 1). There is no automatic clear of the counter. Since counter B counts down from whatever state it is in it is usually desirable to preload the counter. Preloading the counter with all zeroes will give the two's complement of the count. Preloading the counter with all ones will give the one's complement of the count.



TL/DD/6155–18

$t_A = (A + 1)t$

Where: A = Contents of register A
t = Period of internal clock
(as previously defined)
$0 \le A \le 65535$ ($0 \le A \le FFFF_{16}$)

OB toggles each time counter B counts through zero.

Maximum count rate at INB = $f_I/2$

Where: f_I = Internal Clock frequency
= CKI input frequency (÷1 mode)
= CKI input frequency ÷4 (÷4 mode)

Minimum pulse width required for reliable counting = t
where t = period of internal clock.

4. **Dual Count**—In this mode counter A and counter B are enabled as external event or pulse counters. Counter A counts pulses at ZI and counter B counts pulses at INB (when ENB = 1). There is no automatic clear of either counter. Each counter counts down from whatever state it starts in. Thus, to ease reading the information, the counters should be preloaded. Preloading the counters with all zeroes will give the two's complement of the count. Preloading the counters with all ones will give the one's complement of the count. The circuitry which decrements the counters is enabled by the high to low transition at the count input. There is no interaction between the two register counter pairs.

OA toggles every time counter A counts through "0".

OB toggles every time counter B counts through "0".

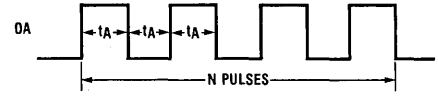
The counters, when counting, count down and wrap around from 0 to FFFF and continue counting down.

Maximum count rate = $f_I/2$
where: f_I = internal clock frequency

Minimum pulse width = t
where: t = period of internal clock
(as previously defined).

There is no requirement that the count signal be symmetrical. The pulse width low must be at least equal to t. The pulse width high must also be at least equal to t.

5. **Number of Pulses Mode**—This mode outputs at OA a specified number of pulses of a specified width. The number of pulses is specified by the contents of register B. The pulse width is specified by the contents of register A.



TL/DD/6155–19

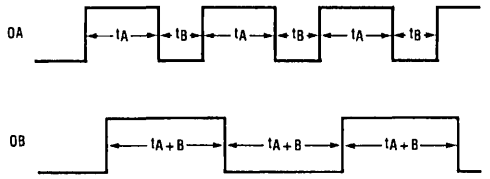
$t_A = (A + 1)t$
$N = B + 1$

Where: A = Contents of register A
B = Contents of register B
t = period of internal clock
(as previously defined)
$1 \le A \le 65535$, $A \ne 0$ ($1 \le A \le FFFF_{16}$)
$0 \le B \le 65535$ ($0 \le B \le FFFF_{16}$)

Functional Description (Continued)

OB toggles each time a pulse train is generated at OA. The pulse is generated each time the COP452L is selected and an instruction is set to the device. Counter B is automatically loaded from register B after the N pulses are generated. Counter A is automatically loaded from register A at each transition of OA. Therefore simply reloading the number of pulses mode will repeat the previous sequence.

6. Duty Cycle Mode—This mode generates a rectangular waveform at OA. The pulse width high is specified by the contents of register A. The pulse width low is specified by the contents of register B. A combination square wave signal is generated at OB.



TL/DD/6155-20

$t_A = At$

$t_B = Bt$

$t_{A+B} = (A + B)t$

Where: A = Contents of register A

B = Contents of register B

t = period of internal clock (as previously defined)

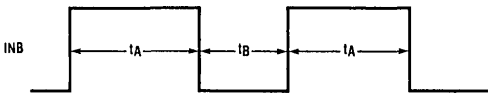
$1 \le A \le 65535, A \ne 0 \quad (1 \le A \le FFFF_{16})$

$0 \le B \le 65535, B \ne 0 \quad (1 \le B \le FFFF_{16})$

7. Waveform Measurement Mode—This mode measures the high and low times of an external waveform at INB (with ENB = 1). Counter A counts the pulse width high and counter B counts the pulse width low. On the high to low transition counter A is transferred to register A and then cleared. On the low to high transition counter B is transferred to register B and then cleared. The counters, therefore, count down from zero. Therefore the value read from the registers is a two's complement value. The transfer from the counter to register is inhibited during a read instruction.

The outputs OA and OB toggle each time the respective counter counts through zero.

The minimum pulse width, either high or low, that can be measured, is the period of the internal frequency. The maximum pulse width that can be measured is the maximum count (65535) multiplied by the period of the internal frequency.



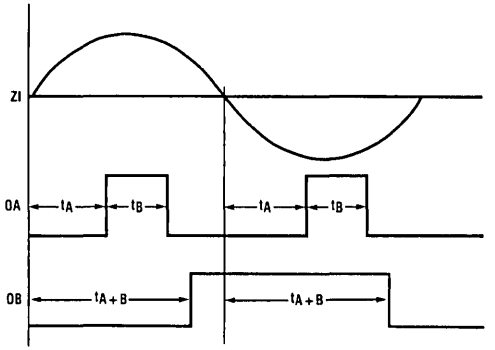
TL/DD/6155-21

$65535t \ge t_A \ge t$

$65535t \ge t_B \ge t$

Where: t = period of internal clock

8. Triggered Pulse Mode—This mode outputs a pulse triggered by the zero crossing of a signal at ZI. The delay from the zero crossing is specified by the contents of register A. The pulse width is specified by the contents of register B. Input INB is ignored. See applications section for further information.



TL/DD/6155-22

$t_A = (A + 1.5)t$

$t_B = Bt$

$t_{A+B} = (A + B + 1.5)t$

Where: A = Contents of register A

B = Contents of register B

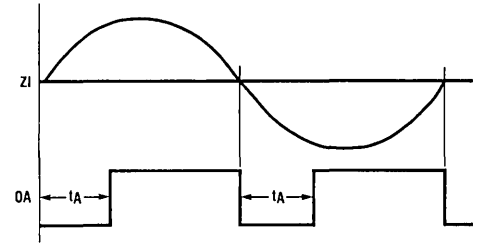
t = period of internal clock (as previously defined)

$0 \le A \le 65535 \quad (0 \le A \le FFFF_{16})$

$1 \le B \le 65535, B \ne 0 \; (1 \le B \le FFFF_{16})$

9. Triggered Pulse and Count Mode—This mode outputs a pulse at OA triggered by the zero crossing of a signal at ZI. The contents of register A specify the delay from the zero crossing. The pulse remains high until the next zero crossing of the signal at ZI.

Independently of the zero detection, counter B counts external events at INB (when ENB = 1). The conditions on the counter as described previously apply here.



TL/DD/6155-23

$t_A = (A + 1.5)t$

Where: A = Contents of register A

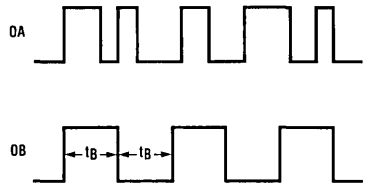
t = period of internal clock (as previously defined)

$0 \le A \le 65535 \quad (0 \le A \le FFFF_{16})$

OB toggles each time counter B counts through 0

Functional Description (Continued)

10. White Noise and Frequency Mode—Register A is converted to a 17-stage shift register generator for the generation of pseudo-random noise at output OA. OB outputs a square wave whose period is specified by the contents of register B. The shift register generator is shifted at the internal frequency (= CKI frequency or ¼ CKI frequency depending on the oscillator divider). See the applications section for more information on the white noise generator.



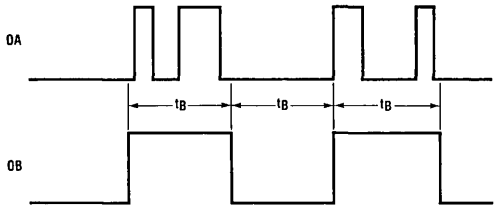
TL/DD/6155-24

$t_B = (B + 1)t$

Where: B = Contents of register B
t = period of internal clock (as previously defined)

$0 \le B \le 65535 \quad (0 \le B \le FFFF_{16})$

11. Gated White Noise Mode—This mode generates pseudo-random noise ANDed with a square wave. OA outputs this combined signal. OB outputs a square wave frequency. Register A is converted into a 17-stage shift register generator which is shifted at the internal frequency rate. Counter A is not used. Counter B and register B are used in the frequency generation. See the applications section for further information on the white noise generation.



TL/DD/6155-25

$t_B = (B + 1)t$

Where: B = Contents of register B
t = period of internal clock (as previously defined)

$0 \le B \le 65535 \quad (0 \le B \le FFFF_{16})$

GENERAL NOTES

The master timing reference in the COP452L is the internal frequency. This is the CKI frequency after it has passed through the divider. This frequency must remain within its specified limits. The maximum count rate at either input is this frequency divided by 2. The minimum pulse width that can be measured is the period of this frequency.

$\overline{CS}$, other than removing DO from the TRI-STATE condition and allowing data to come into the I register via DI, does not affect the operation of the device. $\overline{CS}$ must go high between accesses in order to clear the I register. Since the I register is cleared when $\overline{CS}$ goes high, the user must insure that $\overline{CS}$ does not go high before the COP452L has accepted the information in the I register. See the software interface section for further explanation on this point. CS does not affect the mode latches.

In those modes where there is an automatic transfer from the register to the counter (frequency generation, duty cycle, number of pulses, triggered pulse), care must be exercised when reading or writing the register. To insure proper, "glitch-free" operation, one of the two procedures below must be followed:

1. Place the COP452L in the RESET mode.
2. Read or write the appropriate register.
3. Place the COP452L back in the original mode.

Alternatively:

1. Read or write the appropriate register.
2. Send the instruction to copy the appropriate register to its counter.

WARNING: Failure to observe one or the other of these procedures can cause some faulty output conditions.

The COP452L powers up in the RESET mode and with oscillator divide by 4. If the CKI input frequency is less than 4 times the minimum internal clock frequency the user *must* set the oscillator divider to divide by 1 *before* attempting any operation with the COP452L. The instruction setting the oscillator divider will be accepted regardless of the value of the internal clock frequency.

***Caution:** Failure to observe this requirement will result in the improper operation of the COP452L.*

Applications Information

ZERO CROSS

The ZI input normally requires a resistor and diode external to the device as indicated in *Figure 9a*. The resistor is part of a voltage divider used to ensure that the voltage at pin ZI does not exceed 10V peak and to protect the diode which is required to clamp the negative voltage swing at the input to less than −0.8V. *Figure 9b* is the recommended input circuit if logic level pulses are input to ZI for counting.

As indicated above, the input voltage at ZI must not exceed 10V peak. For inputs less than 10V peak, the resistor in *Figure 9a* is required only to protect the diode. Otherwise, the resistor should be selected to guarantee that the voltage at pin ZI does not exceed 10V peak. *Figure 10* shows this resistor (R_S) and the impedance (R_{IN}) which forms the first part of the input circuit at ZI. The absolute value of R_{IN} can vary widely with process variation. The user should compute the divider with R_S and the worst case maximum of R_{IN} so that the voltage at pin ZI is 10V or less. The following relationship should be used when the input voltage is greater than 10V peak:

$$\frac{R_{IN(MAX.)}}{R_S + R_{IN(MAX.)}} \times V_{IN} \le 10V \text{ peak}$$

Substituting the maximum value for R_{IN} and solving for R_S gives:

$$R_S \le \frac{V_{IN}}{10} \times 7.8k - 7.8k$$

where: V_{IN} = peak input voltage.

Note that this equation is not valid for V_{IN} less than 10V. In this case, the value of R_S is chosen primarily for protection of the diode and not to divide the voltage down to acceptable values.

Applications Information (Continued)

ZERO CROSS OFFSET

As the electrical characteristics indicate, the ZI input has a worst case offset of 150 mV in the zero crossing detection. Therefore, the output of the zero cross detection circuit will change state within ±150 mV of zero volts. There are no directional characteristics to this, i.e., approaching zero from the positive or negative direction has no effect on where the output of the zero cross detection circuit will change state (see *Figure 4*). The offset further indicates that the voltage at pin ZI must exceed 150 mV peak in order to guarantee that the zero crossings will be detected and the appropriate signals generated.

TRIGGERED PULSE MODES

The delays from the zero crossing in the triggered pulse modes are measured from the point where the output of the zero crossing detection circuit changes state—the trip point of this circuit. As stated before, the delay time from this trip point is:

$$T = (A + 1.5)t$$

where: T = delay time from trip point

A = contents of register A

t = period of internal clock

The delay from the true zero crossing of the input waveform has other parameters that must be considered. The equation is of the form:

$$T = (A + 1.5)t \pm |X_1| + X_2 + X_3$$

where: T, A, t are as defined previously

X_1 = time for input waveform to reach the trip point of the zero cross detection circuit

X_2 = propagation delay through the zero cross detection circuit

X_3 = input synchronization delay

Parameter X_1 is dependent on the peak voltage at pin ZI and on the frequency of the input signal. The peak voltage at ZI is in turn dependent on the R_S–R_{IN} voltage divider and the input voltage. The X_1 time is added or subtracted because the trip point of the zero cross detection circuit may be either above or below zero. In the worst case, the trip point is the maximum offset of 150 mV. For a sine wave signal, X_1 is determined as follows:

$$V_{OFFSET} = V_p \sin[2\pi f(X_1)]$$

$$X_1 = \frac{1}{2\pi f} \arcsin \frac{V_{OFFSET}}{V_P}$$

and

$$V_P = V_{IN} \frac{R_{IN}}{R_S + R_{ON}}$$

substituting we have

$$X_1 = \frac{1}{2\pi f} \arcsin \left(V_{OFFSET} \frac{R_S + R_{IN}}{V_{IN} R_{IN}} \right)$$

where: V_{OFFSET} = zero crossing offset or trip point

V_P = peak input voltage at pin ZI

f = frequency of input signal

R_{IN} = internal impedance to ground at pin ZI

R_S = external series resistance at ZI

Both V_{OFFSET} and R_{IN} vary from device to device. It is clear from the equation above that the maximum value of $|X_1|$ is obtained when V_{OFFSET} is at its maximum of 150 mV and R_{IN} is at its minimum of 2.6 kΩ. The minimum value of $|X_1|$ is obtained if V_{OFFSET} is 0. Using this information, the following range of $|X_1|$ is obtained:

$$0 \le |X_1| \le \frac{1}{2\pi f} \arcsin 0.15 \frac{R_S + 2.6k}{V_{IN} \times 2.6k}$$

Parameter X_2 is the propagation delay through the zero crossing detection circuit and its range is given by:

$$0.3\ \mu s \le X_2 \le 0.6\ \mu s$$

Parameter X_3 is the internal synchronization delay and is dependent upon when the zero crossing occurs relative to the internal timing which reads the output of the zero crossing detection circuit. The range for X_3 is:

$$0 \le X_3 \le \frac{t}{2}$$

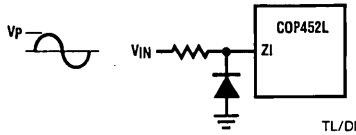
where: t = period of internal clock

With the preceding information, minimum and maximum values of the delay from true zero can be derived by simply substituting into the original equation.

$$T_{MIN} = (A+1.5)t - \frac{1}{2\pi f} \arcsin \left(0.15 \frac{R_S + 2.6k}{V_{IN} \times 2.6k} \right) + 0.3\ \mu s$$

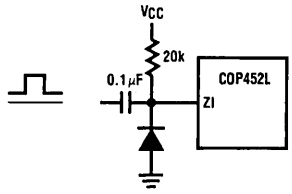
$$T_{MAX} = (A+1.50)t + \frac{1}{2\pi f} \arcsin \left(0.15 \frac{R_S + 2.6k}{V_{IN} \times 2.6k} \right) + 0.6\ \mu s + \frac{t}{2}$$

The preceding information should enable the user to determine more closely the actual delay from zero of output OA of the COP452L. This analysis applies to both of the triggered pulse modes. The three parameters, X_1, X_2, X_3, also apply in the same way in the triggered pulse and count mode when OA returns to 0 since it is the zero cross detection circuit that causes the output to return to zero in that mode.



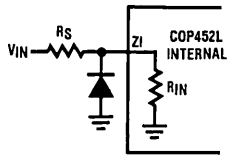
TL/DD/6155-26

FIGURE 9a



TL/DD/6155-27

FIGURE 9b



TL/DD/6155-28

FIGURE 10

Applications Information (Continued)

TRIGGERED PULSE MODES: INTERVENING ZERO CROSSINGS

In the triggered pulse modes, it is possible to specify a delay from the zero crossing which will extend beyond the next zero crossing. In the triggered pulse and count mode, the intervening zero crossing is ignored and therefore lost. The device will still continue to operate properly. The situation is somewhat different in the "pure" triggered pulse mode where both a delay and a pulse width are specified. Any zero crossing which occurs during the programmed delay time is ignored and therefore lost. However, if the delay time is counted out and the zero crossing occurs during the pulse width high time, the zero crossing will be recognized and the delay time will start counting again while the pulse width high time is being counted. This can result in a variety of possible conditions at the output—ranging from the apparent loss of that zero crossing to an effective very short delay from the zero crossing. What will occur depends on the values of the two counters and on their relationship to the times between zero crossings. Some interesting output waveforms can be produced, but their utility is questionable. Therefore, the user should exercise extreme caution in this mode and make sure that the times are such that all zero crossings occur at the "right" times. Otherwise, the user must be prepared to accept the bizarre effects that this situation can produce.

COUNT MODES

As stated before, the counters are 16-bit down counters. Preloading them when they are enabled as external event counters with ones or zeroes will give the one's or two's complement of the count. To read the counters it is necessary to first copy the counter to its respective register and then read the register.

The user can utilize the fact that the outputs toggle when the counter counts through zero. The counter can be preloaded with a value that represents the number of events the user wishes to count. When the output corresponding to that counter toggles, the specified number of events have occurred. Thus, the user can know that the required number of events have occurred without having to actually read the counter.

The counters require a pulse width greater than or equal to the period of the internal frequency in order to be reliably decremented. It is possible for a narrower pulse to decrement the counter, but it is not guaranteed. A narrower pulse will decrement the counter if it appears at the count input at the right time relative to the internal timing of the device. Since the user does not have access to this internal timing, it is impossible for him to synchronize the count input to this timing and effectively reduce the required width of the count pulse. Therefore, applying pulses at the count input of less than one period of the internal frequency in width may cause erratic counting in the sense that some of the pulses may be recognized and some may not be recognized. Reliable counting is assured only if the width of the count pulse is greater than or equal to one period of the internal frequency.

The counters decrement on a low-going pulse at the input. As stated above, the pulse must remain low at least one internal frequency period to give reliable counting. Similarly, the count signal must go high and remain high at least one internal frequency period before it goes low again. However, the count signal does *not* have to be symmetrical.

COP452L OSCILLATOR

The COP452L will operate over a wide range of oscillator input frequencies. The input frequency may be supplied from an external source or CKI and CKO can be used with a crystal or resonator to generate the oscillator frequency. *Figure 11* indicates some crystal networks for some typical crystal values.

RC and LC networks can also be connected between CKI and CKO to produce the oscillation frequency. *Figure 12* indicates some examples of such networks. *Figure 12a* is the recommended RC network for use in this manner. With C_1 = 0.005 μF, R = 1.5 kΩ, and C_2 between 10 pF and 400 pF oscillation frequencies between about 1 MHz and 3 MHz should be obtainable. The oscillation frequency decreases with increasing values of C_2. The user should feel free to experiment with the R and C values, and with the network configuration, to produce the oscillation frequency desired.

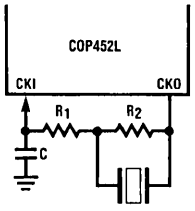
Figures 12b and *12c* indicate LC networks that can be used to produce the COP452L oscillation frequency. In *Figure 12b*, with L = 100 μH and C = 100 pF, a frequency of about 2 MHz should be produced. In *Figure 12c,* with L = 56 μH, C_2 = 27 pF, and C_1 between 25 pF and 0.01 μF, frequencies between about 1.5 MHz and 3 MHz can be produced.

There is, in effect, an inverter between CKI and CKO. This inverter was designed for use with a crystal and its associated network. It was not designed for use with the RC and LC networks previously described. However, these networks will work and are usable. The user should be prepared to experiment with the networks to determine component values, stability, oscillation frequency, etc. These networks should be viewed as the starting point for a user who wishes to use networks of this type to generate the COP452L oscillation frequency.

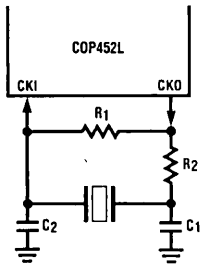
The RC networks provide an inexpensive way to generate the oscillation frequency. It is foolish, however, to expect any significant degree of frequency stability or accuracy over temperature and voltage with a simple RC network—especially if inexpensive, uncompensated components are used. LC and RLC networks can produce very stable and accurate frequencies. Regardless of the network used, the user must consider the variation of the external components in his design if accuracy and stability are important considerations in his application.

The crystal networks of *Figure 11* provide frequency stability and accuracy and are easy to use. If the application requires oscillation frequency accuracy and stability the crystal networks are recommended as the best solution.

Applications Information (Continued)



TL/DD/6155-29

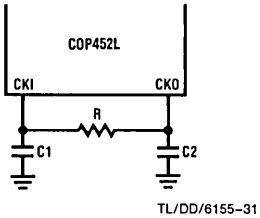


TL/DD/6155-30

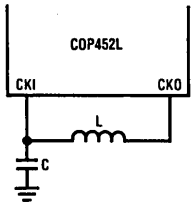
Crystal Value	Component Values		
	R_1	R_2	C
2.0 MHz	1k	1M	56 pF
1.0 MHz	1k	1M	56 pF

Crystal Value	Component Values			
	R_1	R_2	C_1	C_2
455 kHz	1M	16k	80 pF	80 pF
32 kHz	20μ	240k	100 pF	10 pF

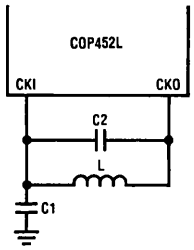
FIGURE 11. COP452 Crystal Oscillator



TL/DD/6155-31



TL/DD/6155-32



TL/DD/6155-33

a. b. c.

FIGURE 12. RC and LC Networks to Produce COP452 Oscillator Frequency

WHITE NOISE GENERATION MODES

In the two white noise modes register A is converted into a 17-stage shift register, or polynomial, generator. With feedback taps at stages 17 and 14, as indicated in *Figure 13*, a maximal length sequence is generated. With these feedback taps the characteristic polynomial of the sequence is:

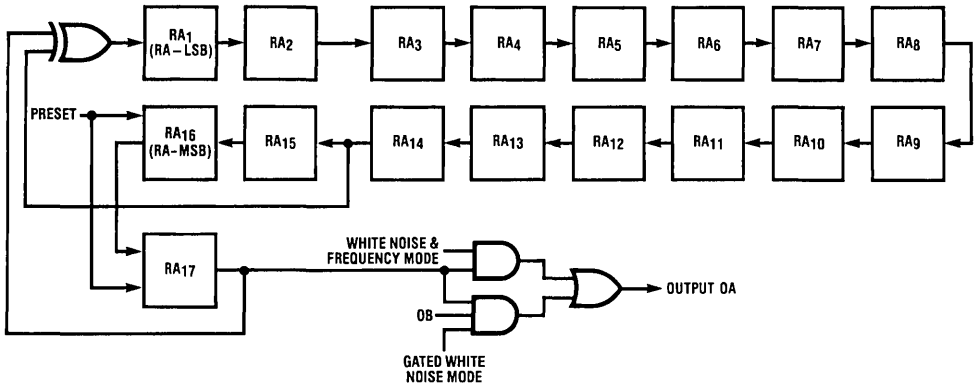
$$X^{17} + X^3 + 1.$$

The output of this generator is a pseudo-random sequence. Since the register is shifted at the internal frequency rate, the sequence repeats after a period equal to $(2^{17} - 1)t$, where t is the period of the internal frequency.

The first 16 stages of the shift register are the 16 bits of register A that the user may read or write. Entering either white noise mode presets the 16th stage to a 1 and connects the 17th stage to the shift register. If the user wishes, he can write register A and then enter the white noise and frequency mode. The output at OA will then be "1", and the lower 15 bits of the data user had written to register A. Following that, the polynomial sequence dictates the output. This injection of a 1 into the 16th stage prevents the lockup condition that occurs if all the stages are 0.

WARNING: To insure proper operation, the white noise must be entered from the Reset mode. The COP452 must be in the Reset mode before the desired white noise mode and there may be no intervening modes between Reset and the desired white noise mode. (The state of 17th stage is don't care (unknown).)

Applications Information (Continued)



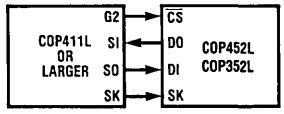
TL/DD/6155-34

Note: Setting the Register A to all 1's will result in a predictable pattern each time this mode is activated.

FIGURE 13. COP452L White Noise Generator

INTERFACE TO COPS MICROCONTROLLERS

Figure 14 indicates the typical interface between the COP452L and a COPS microcontroller. As is obvious from the figure, the interface is the standard MICROWIRE. G_2 is indicated as the chip select line because it is available on all COPS microcontrollers. Obviously, any convenient output of the microcontroller may be used as the chip select for the COP452L.



TL/DD/6155-35

FIGURE 14

The $\overline{CS}$ pin of the COP452L must be toggled between successive communications with the device. The internal I register (instruction register) is held reset (all zero) when $\overline{CS}$ is high. Since this is the only way in which the I register is cleared, failure to take $\overline{CS}$ high between accesses will result in improper operation.

The COP452L contains an internal power-on reset circuit which sets the mode latches to one, i.e., places the COP452L in the RESET mode, and sets the oscillator divider to divide by 4. The counters and registers are not affected by this reset circuit and are therefore undefined at power up.

INTERFACE SOFTWARE FOR THE COP452L

Sample software for interfacing COPS microcontrollers to the COP452L is given below. The code is completely general and will work in any COPS microcontroller. The following assumptions are made:

1. Pin G_2 is used as the chip select for the COP452L (because G_2 is available on all COPS microcontrollers).
2. G_2 is assumed high on entry to the routines.
3. The SK clock is off (0) on entry to the routines.
4. Register 0 of the microcontroller is arbitrarily chosen as the I/O register.
5. The leading digit sent out is of the form 001X where 1 is a start bit; X is 1 or 0, depending on the operation.
6. The next lower digit contains the remaining 4 bits of the command.
7. If data is being sent, it is in the next 16 bits of information sent.
8. Location GSTATE chosen as RAM address 0,15.
9. SK frequency is less than or equal to the internal frequency.

Since the COP452L is an I/O device, the code takes precautions to insure that SO is 0 prior to enabling the SK clock. (This is a wise precaution to take in any system with I/O peripherals on the serial port.)

Two versions of the WRITE routine are provided. The destructive WRITE routine destroys the information in the microcontroller as the data is being sent out to the COP452L. The nondestructive WRITE routine preserves the data in the microcontroller as that data is being sent out to the COP452L. The destructive routine is a little more code efficient than the nondestructive routine.

Applications Information (Continued)

```
WRCMND:   CLRA                ; SET UP POINTER FOR COMMAND ONLY WRITE
          AISC    1
          JP      WRITE
WRDATA:   CLRA                ; SET UP POINTER FOR COMMAND AND DATA WRITE
          AISC    5
WRITE:    LBI     GSTATE      ; GSTATE = LOCATION 0,15
          RMB     2
          OMG                 ; SEND COP452L CHIP SELECT LOW
          CAB                 ; POINT TO PROPER LOCATION FOR OUTPUT
          LEI     8           ; ENABLE SHIFT REGISTER MODE
          RC                  ; JUST TO INSURE SO = 0 BEFORE CLOCK ON
          CLRA
          XAS                 ; THESE 3 WORDS FOR SAFETY ONLY
          SC                  ; SO SK WILL TURN ON AT NEXT XAS
SEND:     LD
          XAS
          XDS
          JP      SEND
FINISH:   RC                  ; ALL DONE, SK OFF, DESELECT COP452L, AND SET
          XAS                 ; SO TO ZERO
DONE:     LBI     GSTATE
          SMB     2
          OMG
          LEI     0
          RET
;
CODE TO WRITE COP452L — DATA DESTROYED IN MICROCONTROLLER
```

TL/DD/6155–36

Applications Information (Continued)

The code below is the code to read COP452L. It is written so that the command to the COP452L is sent out nondestructively, i.e., the data in the microcontroller is preserved. A routine which sends out the data destructively could be easily generated but is not shown here. The user is referred to the techniques in the WRITE routines to determine how to modify this READ routine to send the command out destructively.

```
READ:    CLRA              ; READ INSTRUCTION IN 0, 1 AND 0, 0 AND IS
         AISC    1         ; OF THE FORM 00100010 OR 00100011 IF READ
         LBI     GSTATE    ; RA OR RB
         RMB     2
         OMG               ; SELECT THE COP452L
         CAB
         SC
         CLRA              ; SO THAT ZEROES GO OUT FIRST
         LEI     8
SEND2:   XAS
         LD
         XDS
         JP      SEND2     ; NONDESTRUCTIVE SENDING OF READ INSTRUCTION
         XAS
         CLRA              ; SET UP TO READ
         AISC    2
         CAB
         NOP               ; NOW WAIT FOR THE DATA
         NOP
         NOP
RDLOOP:  CLRA
         XAS
         XDS
         JP      RDLOOP
         RC                ; TURN OFF THE CLOCK
         XAS               ; READ LAST 4 BITS
         JP      DONE      ; COMMON EXIT WITH WRITE ROUTINE
                           ; EXITS WITH DATA IN LOWER 3 DIGITS OF RO
                           ; AND IN THE ACCUMULATOR
```

SAMPLE CODE TO READ THE COP452L

```
WRCMND:  CLRA              ; SET UP POINTER FOR COMMAND ONLY WRITE
         AISC    1
         JP      WRITE
WRDATA:  CLRA              ; SET UP POINTER FOR COMMAND AND DATA WRITE
         AISC    5
WRITE:   LBI     GSTATE
         RMB     2
         OMG               ; SELECT THE COP452L — G2 LOW
         CAB               ; LOAD THE POINTER
         RC
         CLRA
         LEI     8         ; ENABLE SHIFT REGISTER MODE
         XAS               ; SEND OUT ZEROES
         SC
         CLRA
SEND:    XAS               ; FIRST TIME THROUGH, TURNS ON CLOCK
         LD                ; THEN SENDS DATA
         XDS
         JP      SEND
         XAS               ; SEND LAST 4 BITS
         CLRA
         NOP
FINISH:  RC
         XAS               ; ALL DONE, SK OFF
DONE:    LBI     GSTATE
         SMB     2         ; DESELECT THE COP452
         OMG
         LEI     0         ; SEND SO LOW
         RET
```

CODE TO WRITE COP452L — DATA PRESERVED IN MICROCONTROLLER

TL/DD/6155–37

Applications Information (Continued)

The software interface routines provided above are general purpose routines written to work in the general case for all COPS microcontrollers. They are written as subroutines to be called by the main program. There is no question that other routines can be written to perform the required function. It is also clear that these routines can be reduced in specific applications. These routines should be viewed as providing a framework from which the user can develop routines which are optimal to a specific application.

Assumption 9 mentioned prior to the code itself presents an important requirement for the interface software. There must be a time delay greater than 3 periods of the internal frequency between the time the SK clock is turned off and the time the COP452L is deselected. This is required because the COP452L reads the instruction register with timing based on its internal frequency. When the microcontroller deselects the COP452L, CS goes high and the instruction register is automatically cleared. Therefore, depending on the relative speeds of SK and the internal frequency, it is possible that the instruction register may be cleared before the COP452L has accepted the information. The sample code provided automatically satisfies the requirement mentioned above whenever the SK frequency is less than or equal to the counter clock frequency. When SK is faster than the internal frequency, some delay may be required between the time SK is turned off and the time the COP452L is deselected. The time delay is not required when reading or writing the COP452L registers or when changing the oscillator divider.

Caution: *Failure to observe this time delay will result in improper operation of the COP452L.*

APPLICATION # 1—GENERATION OF MULTIPLE TONES

The COP452L makes the generation of two independent frequencies a simple task. This application indicates how to generate frequencies with the COP452L and also indicates other aspects of control of the device.

The requirement is to generate the following two DTMF frequencies:

$$f1 = 941 \text{ Hz}$$

$$f2 = 1336 \text{ Hz}$$

We will select the CKI frequency of the COP452L as 1 MHz primarily for ease in computation. Therefore, in divide by 1 mode, the internal frequency is 1 MHz. Since the registers in the COP452L are loaded with a number related to the period of the frequency, we need the periods of f1 and f2.

$$\frac{1}{f1} = t1 = 1062.7\ \mu s; \quad \frac{t1}{2} = 531.35\ \mu s$$

$$\frac{1}{f2} = t2 = 748.5\ \mu s; \quad \frac{t2}{2} = 374.25\ \mu s$$

As stated earlier, the period of an output frequency in the COP452L in the frequency generation mode is given by:

$$T = 2(N + 1)t$$

where:

t = period of internal clock

N = register value

Solving for N, the equation becomes:

$$N = \frac{T}{2t} - 1$$

With the internal frequency at 1 MHz, the value of t is 1 μs. Therefore, the N values with which the registers must be loaded to generate the frequencies specified above are 530 (212 hex) and 373 (175 hex). Note that the fractional parts of the numbers are lost since the COP452L cannot be loaded with fractional numbers. Note that the fractional parts may be reduced or eliminated by judicious choice of the CKI frequency. With the numbers here, the COP452L will generate a frequency with a period of 1062 μs (941.62 Hz) and a frequency with a period of 748 μs (1336.9 Hz). Note that these values are accurate to within 0.7% of the desired output frequencies.

Figure 15 indicates a connection diagram for this application. The software to accomplish this task is indicated below. The software indicates several aspects of the usage of the COP452L. The code first resets the COP452L, then loads the registers with the proper values, transfers the registers to the counters, puts the COP452L in the CKI divide by 1 state, and then loads the dual frequency mode. The output frequency generation begins when the dual frequency mode is loaded. The code as written is independent of the COP microcontroller used. The code uses the WRITE routines as described in the software interface section and assumes that these routines are located in the subroutine page.

In this, CKI (Max.) = 525 kHz ÷ 1 Mode

```
            . PAGE    0
GSTATE      =         0, 15
POWUP:      CLRA
            XAS                   ; TURN OFF SK CLOCK (C = 0 AT POWER UP)
            LBI       GSTATE
            STII      15
            LBI       GSTATE
            OMG                   ; MAKE SURE COP452 IS DESELECTED
            LBI       0, 0
            JSRP      CLEAR       ; CLEAR REGISTER 0
            LBI       0, 0        ; NOW SET UP TO SEND RESET MODE TO COP452
            STII      15
            STII      3           ; RESET COMMAND AND START BIT
            JSRP      WRCMND
```

TL/DD/6155-38

Applications Information (Continued)

```
; THE COP452L IS NOW RESET, NOW SET UP TO WRITE REGISTER A TO
; GENERATE OUTPUT FREQUENCY OF 941 HZ AT OA

                LBI     0, 0
                STII    2               ; 0212 HEX = 530, GIVE PERIOD OF 1062 µs
                STII    1
                STII    2
                STII    0
                STII    1
                STII    2               ; START BIT PLUS CODE TO WRITE RA
                JSRP    WRDATA
; REGISTER A IS NOW LOADED. NEXT TRANSFER REGISTER A TO COUNTER A
                LBI     0, 0
                STII    5
                STII    2               ; INSTRUCTION TO TRANSFER PLUS START BIT
                JSRP    WRCMND
; ALL DONE WITH REGISTER AND COUNTER A, NEXT WORK ON REGISTER B
                LBI     0, 0
                STII    5               ; WRITE REGISTER B WITH 0175 HEX (373)
                STII    7               ; TO GIVE FREQUENCY OF 1336 HZ
                STII    1
                STII    0
                STII    0               ; INSTRUCTION TO WRITE RB
                STII    2
                JSRP    WRDATA
; REGISTER B IS NOW LOADED. NEXT TRANSFER RB TO CB
                LBI     0, 0
                STII    4               ; INSTRUCTION TO TRANSER RB TO CB
                STII    2
                JSRP    WRCMND
; NOW LOAD CKI DIVIDE BY 1
                LIB     0, 0
                STII    8
                STII    2
                JSRP    WRCMND
; NOW PUT THE COP452 IN DUAL FREQUENCY MODE
                LBI     0, 0
                STII    0
                STII    3
                JSRP    WRCMND
; NOW THE CODE MAY PROCEED TO DO WHATEVER ELSE IS REQUIRED IN
; THE APPLICATION.
; THE SUBROUTINES USED IN THIS APPLICATION ARE CLEAR AND THE
; WRITE ROUTINES. THE ADD ROUTINE IS USED IN THE EXAMPLE BELOW
                . PAGE  2
CLEAR:          CLRA
                XIS
                JP      CLEAR
                RET
ADD:            SC
                LBI     2, 9            ; ROUTINE ADDS 1 TO COUNTER
ADD1:           CLRA
                ASC
                NOP
                XIS
                JP      ADD1
                RET
;
WRCMND:                                 ; SEE SOFTWARE INTERFACE FOR THIS ROUTINE
;
WRDATA:                                 ; SEE SOFTWARE INTERFACE FOR THIS ROUTINE
```

TL/DD/6155-39

Applications Information (Continued)

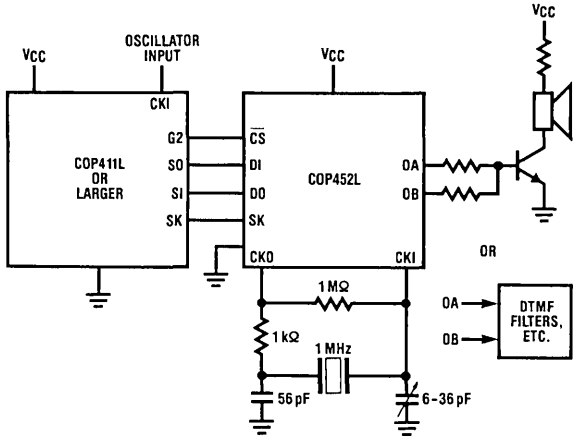
The preceding has done a lot with the COP452L. It is clear that the code can be reduced and specialized. The purpose here was to illustrate the various communications with the device.

An interesting effect can now be produced by making use of the 4 to 1 CKI divider. With the CKI frequency at 1 MHz, the internal frequency is well within the specified limits in either the divide by 1 or divide by 4 condition. Therefore, this characteristic of the device can be used to quickly multiply or divide the output frequency by 4. An interesting siren effect can thus be created. Sample code to do this is given below. This code assumes that the registers have been loaded and that the COP452 is in dual frequency mode. Again, the code is written to be independent of the COPS microcontroller used.

As is obvious from this code, it is a simple matter to create this effect. As was mentioned earlier, the code here is general purpose. This necessarily means that it can be reduced in specific applications. The user should view this code as representative of the techniques involved and then optimize or rewrite the routines to suit his particular application.

```
SIREN:  LBI     2, 9        ; USE REGISTER 2 AS COUNTER FOR DELAY TIME
        JSRP    CLEAR
        LBI     0, 0
        STII    8           ; CKI DIVIDE BY 1
        STII    2
        JSRP    WRCMND
PLUS1:  JSRP    ADD         ; INCREMENT COUNTER FOR DELAY
        SKC
        JP      PLUS1       ; EXIST DELAY LOOP WHEN COUNTER OVERFLOWS
        LBI     0, 0
        STII    9           ; CKI DIVIDE BY 4
        STII    2
        JSRP    WRCMND
        LBI     2,9
        JSRP    CLEAR
PLUS1A: JSRP    ADD
        SKC                 ; AGAIN, TIME OUT VIA THE COUNTER
        JP      PLUS1A
        JP      SIREN       ; DONE, START OVER AGAIN
```

TL/DD/6155–40



TL/DD/6155–41

FIGURE 15. Dual Frequency Application

Applications Information (Continued)

APPLICATION #2

This application makes use of the number of pulses mode of the COP452L to control a stepping motor. The technique is equally applicable in any situation where a number of pulses must be generated based upon the state of the system. *Figure 16* indicates the system interconnect. Since the oscillator frequency is 2.1 MHz max. and the CKO pin of the COP452L is being used to drive the CKI of the microcontroller, a COP420 is specified as the microcontroller. If a separate oscillator were provided, any COPS microcontroller could be used. The software is completely general and will work in any COPS microcontroller.

The application has the following specifications:

1. The pulse width required for the stepping motor is 5 ms ±5%.
2. The system has 4 return lines which indicate 4 possible variations in the number of output pulses required. These four conditions are:
   a. 10 pulses required
   b. 100 pulses required
   c. Repeat the last number of pulses sent
   d. Send one more than the last number of pulses
3. The system has a signal available indicating that the return lines contain valid information.
4. One pulse is required at power up.

A flowchart to implement this system is indicated in *Figure 17*. *Figure 16* is the interconnect used in this application. As the figure indicates, we will use a 2.1 MHz crystal as the time base for the COP452L. With the oscillator divide by 4 selection, this gives an internal frequency period of 1.11745 μs. With this information we can determine the number that needs to be loaded to register A to give a pulse width of 5 ms. From application #1 we have the following equation which is valid here:

$$T = (N + 1)t$$

where: T = pulse width

N = contents of register A

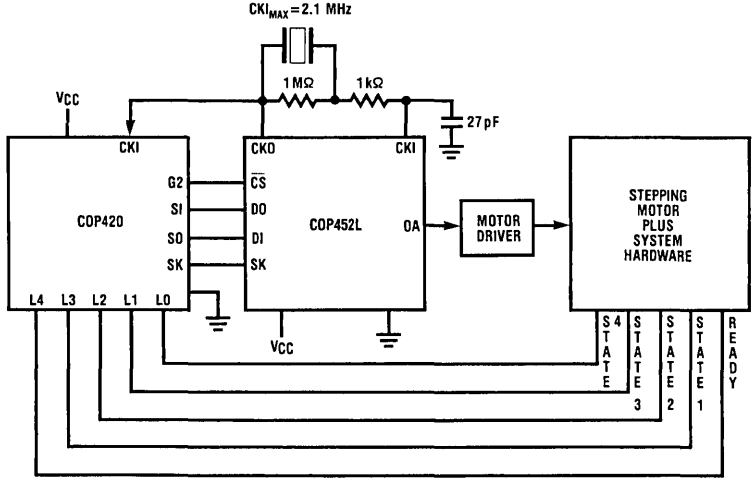
t = period of internal clock

Solving for N we have;

$$N = (T/t) - 1$$
$$= (5\ \text{ms}/1.11746\ \mu\text{s}) - 1$$
$$= 4474.34 - 1$$
$$= 4473.34$$

The fractional part is discarded, so register A must be loaded with 4473 (1179 hex) to give a 5 ms pulse. The error created by the truncation of the number is 0.5 μs. There is an error of 0.01%—well within the tolerance limits required.

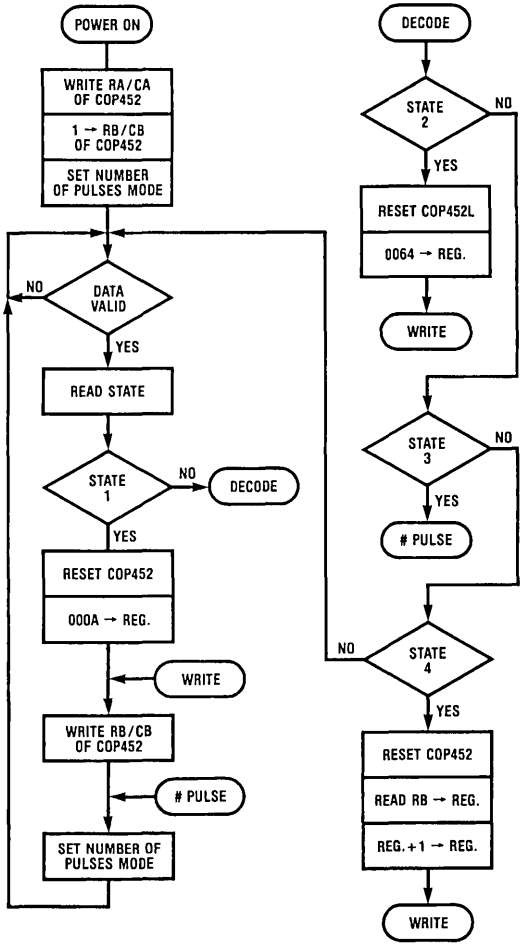
The code to operate this system is given below. The interconnect of *Figure 16* is assumed. The code uses the READ and WRITE subroutines as given in the software interface section of this data sheet. The code further assumes that those routines are located in the subroutine page.



TL/DD/6155-42

FIGURE 16. COP452 in Stepping Motor Control

Applications Information (Continued)



TL/DD/6155-43

FIGURE 17. Flow Diagram for Application #2

```
                . PAGE      0
GSTATE          =           0, 15
POWRON:         CLRA
                XAS                     ; TURN OFF SK CLOCK
                LBI         GSTATE
                STII        15
                LBI         GSTATE
                OMG                     ; DESELECT THE COP452L — G2 HIGH
                LD
                CAMQ                    ; DRIVE THE L LINES HIGH FOR READING
                LEI         4           ; ENABLE THE L OUTPUTS
                LBI         0, 0
                STII        9
                STII        7
                STII        1
                STII        1
                STII        1
                STII        2           ; WRITE RA OF COP452L WITH 1179 HEX TO GET
                JSRP        WRDATA      ; 5MS PULSE
```

TL/DD/6155-44

Applications Information (Continued)

```
                LBI     0, 0
                STII    5               ; TRANSFER RA TO COUNTER A
                STII    2
                JSRP    WRCMND
                LBI     0, 0            ; NOW WRITE RB WITH THE NUMBER OF PULSES
                STII    1
RBWRT:          STII    0               ; ONE PULSE REQUIRED AT POWER UP
RBWRT2:         STII    0
                STII    0
RBWRT3:         STII    0
                STII    2
                JSRP    WRDATA
                LBI     0, 0            ; NOW TRANSFER RB TO COUNTER B
                STII    4
                STII    2
                JSRP    WRCMND
PULSE:          LBI     0, 0
                STII    2               ; SET NUMBER OF PULSES MODE
                STII    3
                JSRP    WRCMND
; AT THIS POINT THE COP452L IS IN NUMBER OF PULSES MODE, ONE
; PULSE IS OUTPUT AT OA. NOW MUST READ THE RETURN LINES, MAKE
; THE APPROPRIATE DETERMINATION OF THE STATE OF THE SYSTEM
; AND UPDATE THE COP452L ACCORDINGLY, ALSO AT THIE POINT, THE
; COP452L IS SET UP TO AGAIN GENERATE A SINGLE PULSE 5 ms WIDE
; IF THE DEVICE IS ACCESSED AGAIN.
;
STATE:          LBI     GSTATE
                LD                      ; CONTENTS OF GSTATE = 15 HERE
                CAMQ                    ; MAKE SURE L LINES ARE HIGH AND
                LEI     4               ; ENABLED
                LBI     0, 0
                INL                     ; READ THE L LINES TO A AND M(0, 0)
                SKMBZ   0               ; TEST DATA — RETURN LINES — VALID
                JMP     STATE           ; DATA NOT VALID, WAIT FOR IT TO BE VALID
                AISC    8               ; DATA IS VALID, DECODE A
                JMP     TEST2
STATE1:         STII    15              ; POINTING AT 0, 0
                STII    3               ; RESET THE COP452L FOR STATE 1
                JSRP    WRCMND
                LBI     0, 0            ; NOW SET UP TO SEND 10 PULSES
                STII    10
                JMP     RBWRIT          ; SHARE COMMON CODE
TEST2:          AISC    4
                JMP     TEST3
STATE2:         STII    15              ; IN STATE2, MUST SEND 100 PULSES
                STII    3               ; FIRST RESET THE COP452L
                JSRP    WRCMND
                LBI     0, 0            ; WRITE 100 (0064 HEX) TO RB OF COP452L
                STII    4
                STII    6
                JMP     RBWRT2
TEST3:          AISC    2
                JMP     TEST4
STATE3:         JMP     PULSE           ; STATE 3 MERELY SENDS THE SAME NUMBER OF PULSES AGAIN.
                                        ; THEREFORE, MERELY SEND THE NUMBER OF PULSES MODE COMMAND
                                        ; AGAIN
```

TL/DD/6155–45

Applications Information (Continued)

```
TEST4:      AISC    1
            JMP     STATE       ; ALL L LINES WERE 0, JUMP BACK TO MAIN
STATE4:     STII    15          ; RESET THE COP452L
            STII    3
            JSRP    WRCMND
            LBI     0, 0        ; NOW READ THE COP452L
            STII    2
            STII    2           ; COMMAND TO READ RB
            JSRP    READ
            LBI     0, 0        ; MOVE DATA TO LAST 4 DIGITS OF R0
            XIS
            XIS
            XIS
            XIS
            LBI     0, 0        ; NOW INCREMENT THE VALUE BY 1
            SC
PLUS1:      CLRA
            ASC
            NOP
            XIS
            CBA
            AISC    12
            JP      PLUS1
            JMP     RBWRT3      ; HAVE INCREMENTED THE VALUE, SEND IT OUT
;
            . PAGE  2
READ:
                                ; SEE SOFTWARE INTERFACE SECTION FOR THESE
WRDATA:                         ; ROUTINES

WRCMND:
```

TL/DD/6155–46

These are general routines and can be reduced in specific applications. The application itself was kept general so that it can be easily adapted to particular applications. The user should view this code as the basis from which to work to optimize the code for a specific application.

APPLICATION #3

An application such as a tachometer requires the counting of external pulses that occur within a given time period. The COP452L can be used both to perform the counting and to establish the "viewing window," or time period, during which to count the pulses. By using the frequency and count mode of the COP452L, a frequency can be generated which will establish this viewing time. The other counter can then be used to count the pulses. *Figure 18* provides a diagram of the interconnect in this application.

As *Figure 18* indicates, the oscillator frequency for the COP452L has been selected as 250 kHz. With the oscillator divider set at divide by 1, the internal frequency is also 250 kHz. At this frequency, the minimum pulse width that can be reliably expected to decrement the counter is 4 μs—the period of the internal frequency.

A viewing time of 250 ms is arbitrarily selected. This means that the period of the output frequency is 500 ms—a frequency of 2 Hz. Using the equation developed earlier for determining the counter values we have:

$$N = \frac{T}{2t} - 1$$

$$= (500\ \text{ms}/8\ \mu\text{s}) - 1$$

$$= 62500 - 1$$

$$N = 62499 = \text{F423 hex}$$

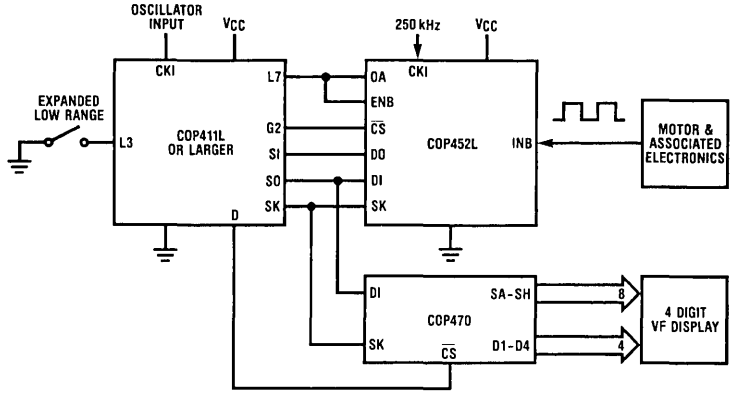
Therefore, register A must be loaded with the hex value F423 to generate a frequency of 2 Hz at OA. Counter B will count pulses when OA is high by virtue of the ENB input. When OA is low, the microcontroller will read and reset the counter and peform any necessary operations.

With the values above for the internal frequency and the viewing window, the tachometer range is 240 RPM to 62,500 RPM. By making use of the divide by 1/divide by 4 features of the oscillator divider, the range can be extended down to 60 RPM. The range when the oscillator is divided by 4 is 60 RPM to 15,625 RPM. However, a penalty is paid for this range extension. The viewing window goes from 250 ms to 1 second. The minimum reliable pulse width also increases from 4 μs to 16 μs. The added time spent counting may or may not be acceptable. It can be reduced somewhat by changing the value of RA to give a faster frequency at the reduced counter clock frequency. However, as the OA frequency increases, the low end of the range increases.

A flow chart for this application is provided in *Figure 19*. Sample code is given below. Note that the sample code includes only the COP452L interface and control. Other system requirements, e.g., display interface, arithmetic, etc., are not included here. Other data sheets and application notes provide sufficient information to fill in those details.

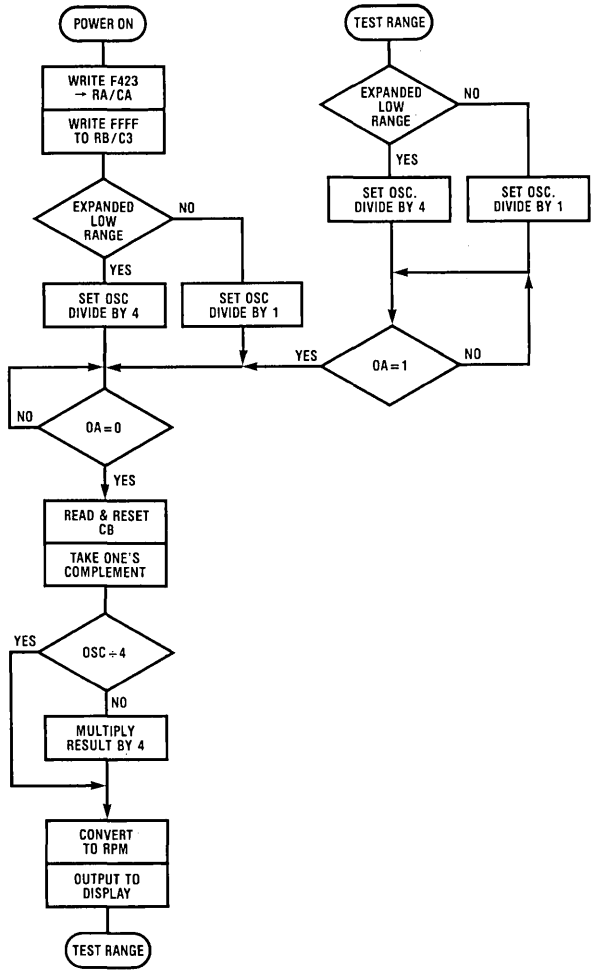
The hardware interface indicated in *Figure 18* and the code below, are completely general and valid of any COPS microcontroller. In specific applications both the hardware and software may be optimized to a greater extent than that shown here.

Applications Information (Continued)



TL/DD/6155–47

FIGURE 18. COP452 in Wide Range Tachometer Application



TL/DD/6155–48

FIGURE 19. Flowchart for Tachometer Application

Applications Information (Continued)

```
                .PAGE   0
GSTATE          =       0,15
POWRON:         CLRA
                XAS                 ; TURN OFF THE SK CLOCK—C = 0 AT POWER UP
                LBI     GSTATE
                OBD                 ; DRIVE D LINES HIGH TO DESELECT DISPLAY
                STII    15
                LBI     GSTATE
                OMG                 ; DESELECT THE COP452L
                LD
                CAMQ                ; SET THE Q REGISTER TO ALL 1'S FOR INPUT
                LBI     0,0
                STII    3           ; NOW SET UP TO WRITE RA OF COP452L
                STII    2
                STII    4
                STII    15          ; WRITE RA WITH F423 HEX
                STII    1
                STII    2           ; REMEMBER COP452L IS RESET AT POWER UP
                JSRP    WRDATA
                LBI     0,0
                STII    5           ; TRANSFER RA TO CA
                STII    2
                JSRP    WRCMND
                JSR     RSTRB       ; RESET RB AND COUNTER B WITH FFFF
                JSR     RANGE       ; TEST RANGE AND SET OSCILLATOR DIVIDER
                LEI     4           ; ENABLE Q TO L—DRIVE L LINES HIGH
                LBI     0,0         ; LOOK FOR OA = 0
TSTOA0:         INL
                SKMBZ   3
                JP      TSTOA0
                LBI     0,0         ; OA IS 0, READ COUNTER
                STII    6           ; FIRST TRANSFER CB TO RB
                STII    2
                JSRP    WRCMND
                LBI     0,0         ; THEN READ RB
                STII    2
                STII    2
                JSRP    READ
                LBI     0,0         ; NOW TAKE THE 1'S COMPLEMENT
ONECMP:         COMP
                XIS
                COMP
                XIS
                COMP
                XIS
                COMP
                X
                LBI     0,0         ; NOW SAVE VALUE IN R1
XFER1:          LD      1
                XIS     1
                JP      XFER1
                JSR     RSTRB       ; RESET RB AND CB WITH FFFF FOR NEXT TIME
;
; AT THIS POINT INSERT THE APPROPRIATE CODE FOR ANY NECESSARY
; ARITHMETIC, BINARY/BCD CONVERSION, DISPLAY OUTPUT, AND ANY OTHER
; SYSTEM REQUIREMENTS. AFTER THESE ARE COMPLETE, JUMP TO LABEL
; TSTRNG WHICH HAS BEEN ARBITRARILY PLACED IN PAGE 4.
. PAGE          2
WRDATA:

WRCMND:                         ; SEE SOFTWARE INTERFACE SECTION FOR THESE
                                ; THREE ROUTINES
READ:
                .PAGE   4
TSTRNG:         JSR     RANGE       ; CHECK THE RANGE
                LEI     4           ; BE SURE Q IS ENABLED TO L
                LBI     0,0         ; LOOK FOR OA = 1
TSTOA1:         INL
                SKMBZ   3
                JMP     TSTOA0
                JP      TSTOA1
;
; THE SUBROUTINES RANGE AND RSTRB ARE INSERTED HERE
;
RANGE:          LEI     4           ; MAKE SURE L ENABLED
                LBI     3,15        ; WILL SAVE RANGE STATUS IN 3, 15
                INL
                X
                CLRA                ; NOW PREPARE TO SET OSCILLATOR DIVIDER
```

TL/DD/6155–49

TL/DD/6155–50

3

Applications Information (Continued)

```
                AISC    8           ; AN 8 MEANS DIVIDE BY 1
                SKMBZ   3
                JP      HILOW
LOW:            AISC    1           ; IF DIVIDE BY 4, WANT A 9 IN A
HILOW:          LBI     0, 0
                XIS
                STII    2
                JMP     WRCMND
;
; THE FOLLOWING SUBROUTINE USES A SUBROUTINE LEVEL. IT RESETS BOTH
; REGISTER B AND COUNTER B OF THE COP452L TO FFFF
;
RSTRB:          LBI     0, 0
                STII    15
                STII    15
                STII    15
                STII    15
                STII    0
                STII    2
                JSRP    WRDATA      ; WRITE FFFF TO RB
                LBI     0, 0
                STII    4           ; TRANSFER RB TO CB
                STII    2
                JMP     WRCMND
```

TL/DD/6155–51

APPLICATION #4

The triggered pulse mode of the COP452L provides the capability of generating the appropriate signals for triac control. *Figure 20* is a general diagram of such an application.

Assume the requirement is to switch on the triac 45 degrees into the waveform. With a 60 Hz sine wave signal, the 45 degree delay is 2.0833 ms from the zero crossing. Assume also that the triac requires a gate pulse width of 150 μs. As the diagram indicates, a 2.097 MHz crystal provides the oscillator input to the COP452L. With the above information the two values that must be loaded in the COP452L can be determined. With CKI at 2.097 MHz and the oscillator divider at divide by 4, the period of the internal frequency is 1.9075 μs. From the description of the triggered pulse mode, the pulse width is given by:

$$T = Bt$$

where: T = desired pulse width

B = contents of register B

t = period of internal clock

Solving for B is trivial and gives:

$$B = T/t$$
$$= 150\ \mu s/1.9075\ \mu s$$
$$= 78.64$$

Since the register and counter can be loaded with whole numbers only, register B and counter B must be initialized with 79 (002F hex) to give a pulse width of 150 μs.

The delay from the zero cross trip point is given by:

$$T = (A + 1.5)t$$

where: T = delay from zero cross trip point

A = contents of register A

t = period of internal clock

Solving for A we have:

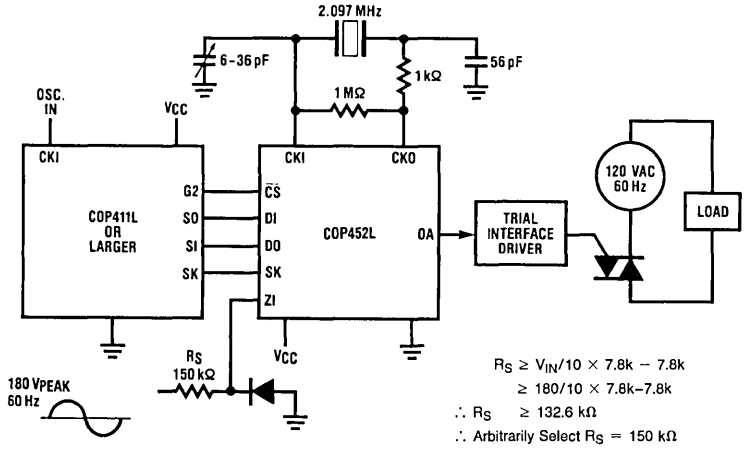
$$A = (T/t) - 1.5$$
$$= (2.0833\ ms/1.9075\ \mu s) - 1.5$$
$$A = 1090.66 \text{ rounded up to } 1091$$

Therefore register A and counter A must be initialized with 1091 (0443 hex) to delay 2.0833 ms (45 degrees at 60 Hz) from zero cross.

Once the data has been given to the COP452L and the device placed in the triggered pulse mode, no further attention is required. The COP452L will generate the pulses with the appropriate delay as long as the power is applied and the input sine wave is available. It is a trivial matter to change any of the information. Merely write the appropriate register/counter pair. Thus very easy control is available over the firing angle of triacs.

Sample code to accomplish this function is given below. The code is general purpose and is written to work in any COPS microcontroller.

Applications Information (Continued)



TL/DD/6155-52

FIGURE 20. COP452L as Triac Controller

```
          . PAGE    0
GSTATE    =         0, 15
POWRON:   CLRA
          XAS                   ; TURN OFF THE SK CLOCK
          LBI       GSTATE
          STII      15
          LBI       GSTATE
          OMG                   ; DESELECT THE COP452L — G2 HIGH
          LBI       0, 0        ; NOW WRITE RB/CB WITH 002F HEX TO GIVE
          STII      15          ; 150 μs PULSE WIDTH
          STII      2
          STII      0
          STII      0
          STII      0
          STII      2
          JSRP      WRDATA
          LBI       0, 0
          STII      4           ; TRANSFER RB TO CB
          STII      2
          JSRP      WRCMND
          LBI       0, 0        ; NOW WRITE RA/CA WITH 0443 HEX FOR THE DELAY
          STII      3
          STII      4
          STII      4
          STII      0
          STII      1
          STII      2
          JSRP      WRDATA
          LBI       0, 0
          STII      5
          STII      2
          JSRP      WRCMND      ; TRANSFER RA TO CA
          LBI       0, 0
          STII      9
```

TL/DD/6155-53

```
          STII      2           ; SET OSCILLATOR DIVIDER TO DIVIDE BY 4
          JSRP      WRCMND
          LBI       0, 0
          STII      1           ; SET TRIGGERED PULSE MODE
          STII      3
          JSRP      WRCMND
; ALL COMPLETE AT THIS POINT. ROUTINES WRCMND AND WRDATA ASSUMED
; IN PAGE 2 AND ARE THE SAME AS GIVEN IN SOFTWARE INTERFACE SECTION.
; THE COP452L WILL NOW GENERATE THE 150 μs PULSE DELAYED BY 2.0833 ms
; FROM EVERY ZERO CROSSING. THE USER CAN NOW IGNORE THE TRIAC CONTROL
; AND DO WHATEVER ELSE IS REQUIRED IN THE SYSTEM. FURTHER ATTENTION
; IS REQUIRED ONLY WHEN THE DATA IN THE COP452 MUST BE CHANGED.
```

TL/DD/6155-54

Applications Information (Continued)

Let us now compute the minimum and maximum delays from the true zero crossing in this application. As indicated earlier, the period of the internal frequency here is 1.9075 μs. Counter A contains 0443 hex (decimal 1091). R_S is 150k and the peak input voltage is 180V. A 60 Hz sine wave is assumed. As given earlier, the minimum time is:

$$T_{MIN} = (A+1.5)t - \frac{1}{2\pi f}\arcsin\left(0.15\frac{R_S+2.6k}{V_{IN}\times 2.6k}\right)+0.3\ \mu s$$

Substituting we have:

$$T_{MIN} = 1092.5t - \frac{1}{120\pi}\arcsin\left(0.15\frac{152.6k}{180\times 2.6k}\right)+0.3\ \mu s$$

$$= 2093.9\ \mu s - 129.7\ \mu s + 0.3\ \mu s$$

$$T_{MIN} = 1954.5\ \mu s$$

Similarly, the maximum time is given as:

$$T_{MAX} = (A + 1.5)t + \frac{1}{2\pi f}\arcsin\left(0.15\frac{R_S + 2.6k}{V_{IN}\times 2.6k}\right) + 0.6\ \mu s + \frac{t}{2}$$

Substituting, we have:

$$T_{MAX} = 1092.5t + \frac{1}{120\pi}\arcsin\left(0.15\frac{152.6k}{180\times 2.6k}\right) + 0.6\ \mu s + \frac{1.9075\ \mu s}{2}$$

$$= 2083.9\ \mu s + 129.7\ \mu s + 0.6\ \mu s + 0.9538\ \mu s$$

$$T_{MAX} = 2215.15\ \mu s$$

As is obvious from the preceding analysis, the parameter previously defined as X_1 is the most significant of the additional factors that define the time delay from true zero. This factor can be minimized by using as small a series resistance as possible. The frequency and input voltage will be governed by the application. The user must also remember that the minimum and maximum times calculated in this manner are absolute worst case values derived using the worst case condition.

National Semiconductor Corporation

COP470/COP370 V.F. Display Driver

General Description

The COP470 is a peripheral member of National's COPS™ Microcontroller family. It is designed to directly drive a multiplexed Vacuum Fluorescent display. Data is loaded serially and held in internal latches. The COP470 has an on-chip oscillator to multiplex four digits of eight segment display and may be cascaded and/or stacked to drive more digits, more segments, or both.

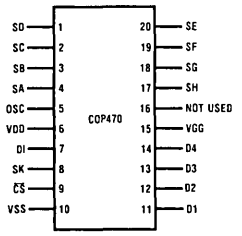
With the addition of external drivers, the COP470 also provides a convenient means of interfacing to a large-digit LED display. The COP370 is the extended temperature range version of the COP470.

Features

- Directly interfaces to multiplexed 4 digit by 8 segment Vacuum Fluorescent displays
- Expandable to drive 8 digits and/or 16 segments
- Compatible with all COP400 processors
- Needs no refresh from processor
- Internal or external oscillator
- No "glitches" on outputs when loading data
- Drives large and small displays
- Programmable display brightness
- Small (20-pin) dual-in-line package
- Operates from 4.5V to 9.5V
- Outputs switch 35V and require no external resistors
- Static latches
- MICROWIRE™ compatible serial I/O
- Extended temperature device COP370 (−40°C to +85°C)

Connection and Block Diagrams

Dual-In-Line Package

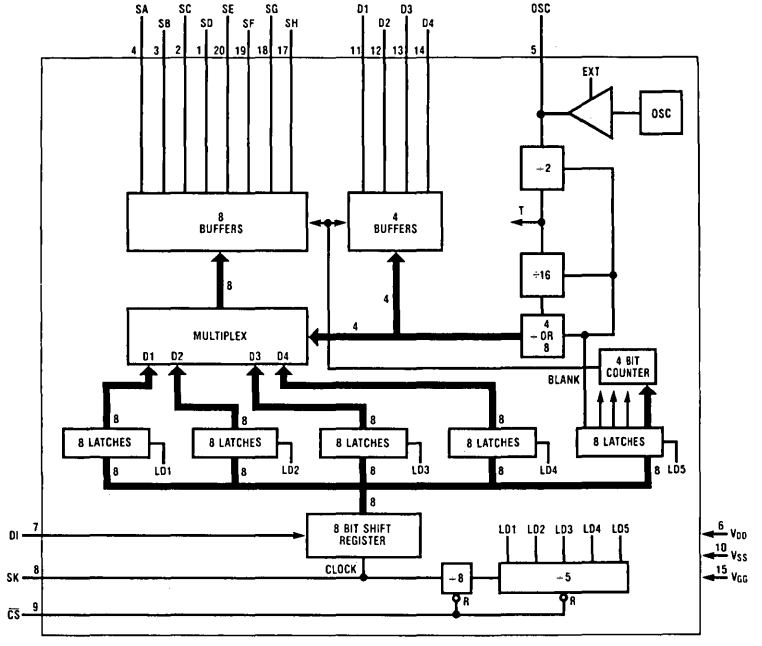


TL/DD/6154-1

Top View

FIGURE 1. COP470

Order Number COP470D, COP370D, COP470N or COP370N
See NS Package Number D20A or N20A



TL/DD/6154-2

FIGURE 2. COP470

Absolute Maximum Ratings (V_{SS} = 0)

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Display Outputs	+0.3V to −35V
Voltage at All Other Pins	+0.3V to −20V
Operating Temperature	
COP470	0°C to +70°C
COP370	−40°C to +85°C
Storage Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 sec.)	300°C
Package Power Dissipation	400 mW at 25°C 200 mW at 70°C 125 mW at 85°C

DC Electrical Characteristics

V_{SS} = 0, V_{DD} = −4.5V to −9.5V, V_{GG} = −30V to −35V, T_A = 0°C to +70°C for COP470 and T_A = 40°C to 85°C for COP370 unless otherwise specified.

Parameter	Min	Max	Units
Power Supply Voltage			
V_{DD}	−9.5	−4.5	V
V_{GG} (COP470)	−35	V_{DD}	V
V_{GG} (COP370)	−32	V_{DD}	V
Power Supply Current			
I_{DD}		5	mA
I_{GG} (Displayed Blanked)		1	mA
Input Levels			
V_{IH}	−1.5	+0.3	V
V_{IL}	−10.0	−4.0	V
Output Drive Digits and Segments			
I_{OH} @ V_{OH} = V_{SS} − 3V	10		mA
I_{OH} @ V_{OH} = V_{SS} − 2V	7		mA
I_{OL} @ V_{OL} = V_{GG} + 2V(1)	10		μA
Output Drive @ V_{GG} = V_{DD} = V_{SS} −5V			
I_{OH} @ V_{OH} = V_{SS} −2V	1		mA
Allowable Source Current			
Per Pin		20	mA
Total for Segments		60	mA
Input Capacitance		7	pF
Input Leakage		1	μA

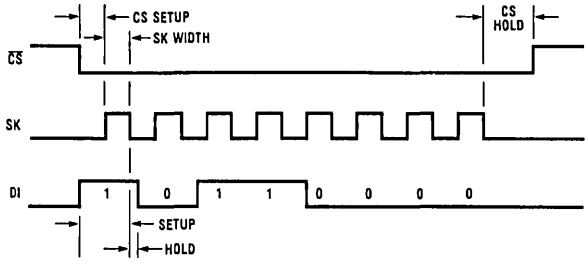
AC Electrical Characteristics

V_{SS} = 0, V_{DD} = −4.5V to −9.5V, V_{GG} = −30V to −35V, T_A = 0°C to +70°C for COP470 and T_A = 40°C to 85°C for COP370 unless otherwise specified.

Parameter	Min	Max	Units
OSC Period (internal or external)	4	20	μs
OSC Pulse Width	1.5		μs
Clock Period T (twice Osc. period)	8	40	μs
Display Frequency			
4 digits = 1/64T	390	2000	Hz
8 digits = 1/128T	190	1000	Hz
SK Clock Frequency	0	250	kHz
SK Clock Width	1.5		μs
Data Set-up and Hold Time			
t_{SETUP}	1.0		μs
t_{HOLD}	50		ns
CS Set-up and Hold Time			
t_{SETUP}	1.0		μs
t_{HOLD}	1.0		μs
Duty Cycle			
4 digits	1/64	15/64	
8 digits	1/128	15/128	

Note 1: I_{OL} current is to V_{GG} with the chip running. Current is measured just after the output makes a high-to-low transition.

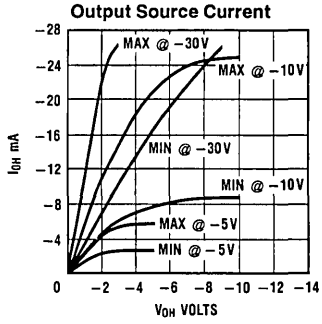
Timing Diagram



TL/DD/6154-3

FIGURE 3. Serial Load Timing Diagram

Performance Characteristic



TL/DD/6154-4

Functional Description

SEGMENT DATA BITS

Data is loaded in serially in sets. Each set of segment data is in the following format:

SA	SB	SC	SD	SE	SF	SG	SH

Data is shifted into an eight bit shift register. The first bit of the data is for segment H, digit 1. The eighth bit is segment A, digit 1.

A set of eight bits is shifted in and then loaded into the digit one latches. The second set of 8 bits is loaded into digit two latches. The third set into digit three latches and the fourth set is loaded into digit four latches.

DISPLAY ON TIME AND CONTROL BITS

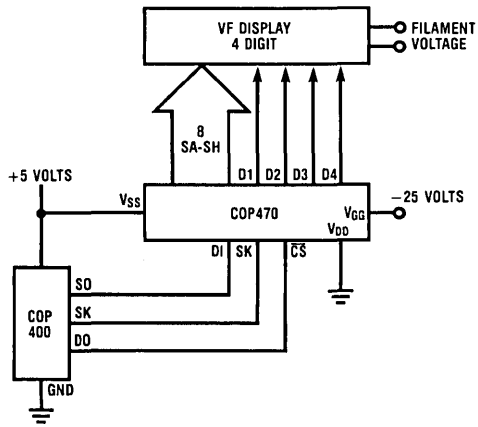
The fifth set of 8 data bits contains blank time data and control data in the following format:

Display Digits

Sync	Ext.	Right	Left	←	On Time	→	
	Osc.	4 of 8	4 of 8	LSB			MSB

The first four bits shifted in contain the on time. This is used to control display brightness. The brightness is a function of the on time of each segment divided by the total time (duty cycle). The on time is programmable from 0 to 15 and the total time is 64. For example, if the on time is 15, the duty cycle is 15/64 which is maximum brightness. If on time is 8, the duty cycle is 8/64, about 1/2 brightness. There are 16 levels of brightness from 15/64 to 0/64 (off).

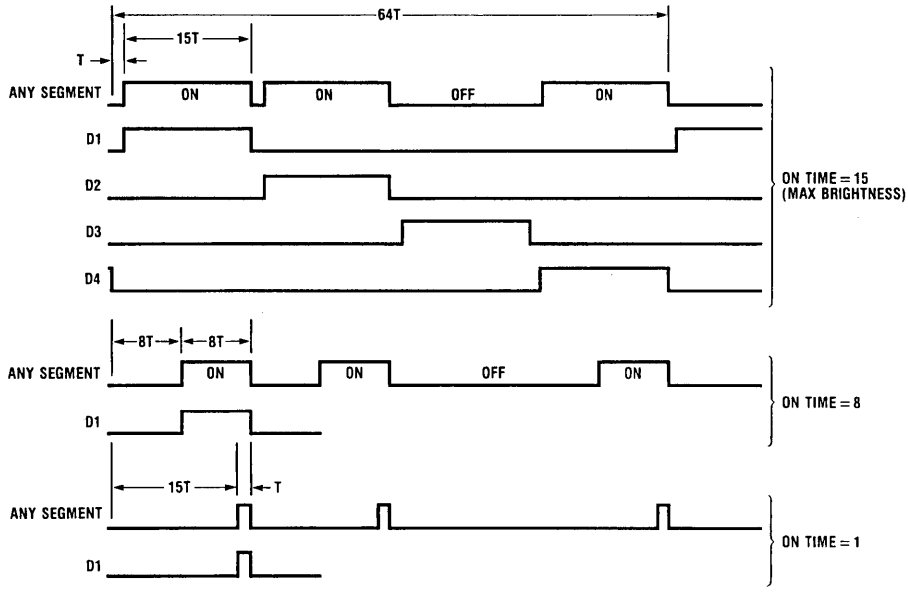
The fifth and sixth bits control the multiplex digits. To enable the COP470 to drive a 4 digit multiplex display, set both bits to one. If two COP470s are used to drive an 8 digit display, bit five is set on the left COP470 and bit six is set on the right COP470 (see *Figure 6*). In the eight digit mode, the display duty cycle is on time/128.



TL/DD/6154-5

FIGURE 4. System Diagram—4 Digit Display

Functional Description (Continued)



TL/DD/6154–6

FIGURE 5. Segment and Digit Output Timing Diagram

The seventh bit selects internal or external oscillator. The OSC pin of the COP470 is either an output of the internal oscillator (bit 7 = 0) or is an input allowing the COP470 to run from an external oscillator (bit 7 = 1).

The eighth bit is set to synchronize two COP470s. For example, to set the COP470 to internal osc, 4 digits, and maximum brightness, send out six ones and two zeros.

LOADING SEQUENCE

Step

1. Turn $\overline{CS}$ Low.
2. Clock in 8 bits of data for digit 1.
3. Clock in 8 bits of data for digit 2.
4. Clock in 8 bits of data for digit 3.
5. Clock in 8 bits of data for digit 4.
6. Clock in 8 bits of data for on time and control bits.
7. Turn $\overline{CS}$ high.

Note: $\overline{CS}$ may be turned high after any step. For example, to load only 2 digits of data do steps 1, 2, 3, and 7. $\overline{CS}$ must make a high to low transition before loading data in order to reset internal counters.

8 DIGIT Displays

Two COP470s may be tied together in order to drive an eight digit multiplexed display. This is shown in *Figure 6.* The following is the loading sequence to drive an eight digit display using two COP470s.

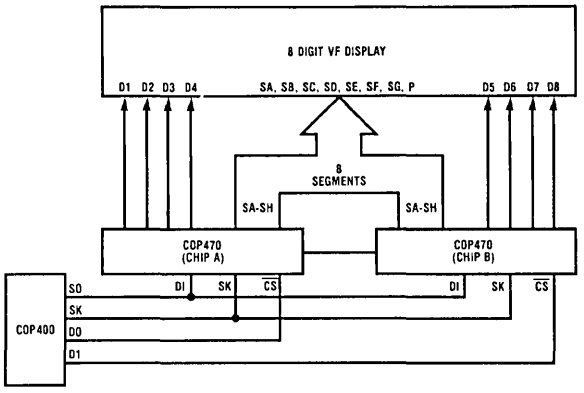
1. Turn $\overline{CS}$ lwo on both COP470s.
2. Shift in 32 bits of data for the right 4 digits.
3. Shift in 4 bits of on time, a zero and three ones. This synchronizes both chips, sets to external oscillator, and to right four of eight digits. Thus both chips are synchronized and the oscillator is stopped.
4. Turn $\overline{CS}$ high to both chips.
5. Turn $\overline{CS}$ low to the left COP470.
6. Shift in 32 bits of data for the left 4 digits.
7. Shift in 4 bits of on time, a one and three zeros. This sets this COP470 to internal oscillator and to left four of eight digits. Now both chips start and run off the same oscillator.
8. Turn $\overline{CS}$ high.

The chips are now synchronized and driving eight digits of display. To load new data simply load each chip separately in the normal manner.

16 SEGMENT DISPLAY

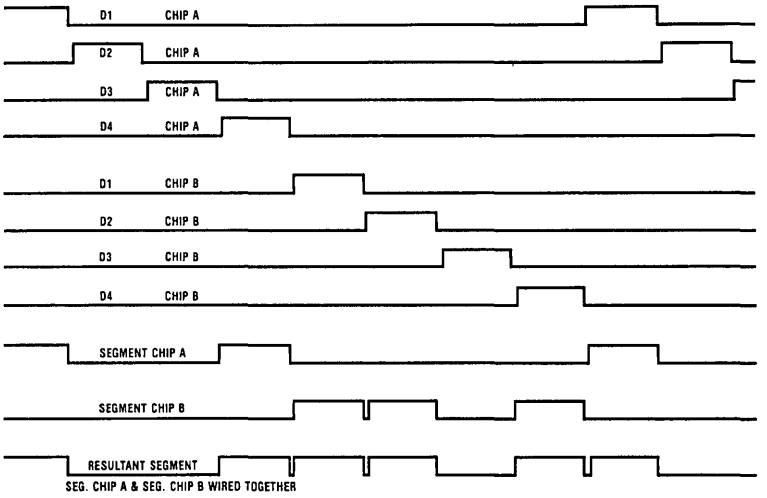
Two COP470s may be tied together in order to drive a sixteen segment display. This is shown in *Figure 8.* To do this, both chips must be synchronized, one must run off external oscillator while the other runs off its internal oscillator outputting to the other. Similarly, four COP470s could be tied together to drive eight digits of sixteen segments.

Functional Description (Continued)



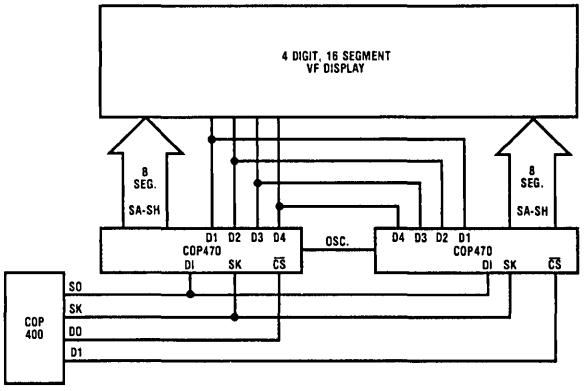
TL/DD/6154-7

FIGURE 6. System Diagram 8 Digit Display



TL/DD/6154-8

FIGURE 7. Segment and Digit Output Timing Diagram for 8 Digits



TL/DD/6154-9

FIGURE 8. System Diagram for 16 Segment Display

Functional Description (Continued)

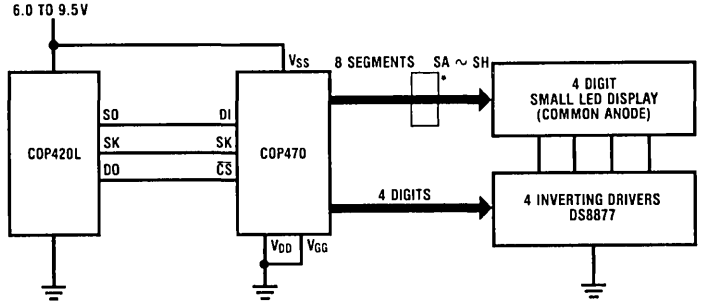
LED DISPLAY

The COP470 may be used to drive LED displays. The COP470 can drive the segments directly on small, low current LED displays as shown in *Figure 9.* By adding display drivers, large, high current LED displays can be driven as shown in *Figure 10.*

Example:
COP420 Code to Load COP470
(Display Data is in Memory 0, 12–0, 15)

```
        LBI 0,12        ; Point to first display data
        OBD             ; Turn CS low (DO)
LOOP:   CLRA
        LQID            ; Look up segment data
        CQMA            ; Copy data from Q to M & A
        SC              ; Set C to turn on SK
        XAS             ; Output lower 4 bits of data
        NOP             ; Delay
        NOP             ; Delay
        LD              ; Load A with upper 4 bits
        XAS             ; Output 4 bits of data
        NOP             ; Delay
        NOP             ; Delay
        RC              ; Reset C
        XAS             ; Turn off SK clock
        XIS             ; Increment B for next data
        JP LOOP         ; Skip this jump after last digit
        SC              ; Set C
        CLRA            ;
        AISC 15         ; 15 to A
        XAS             ; Output on time (max brightness)
        NOP             ;
        CLRA            ;
        AISC 12         ; 12 to A
        XAS             ; Output control bits
        NOP             ;
        LBI 0,15        ; 15 to B
        RC              ; Reset C
        XAS             ; Turn off SK
        OBD             ; Turn CS high (DO)
```

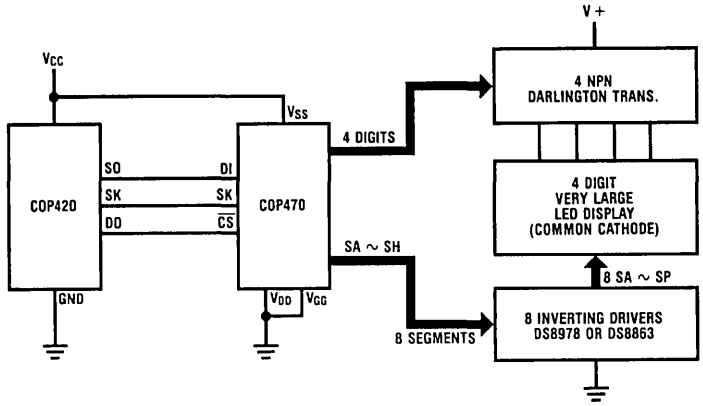
Functional Description (Continued)



TL/DD/6154–10

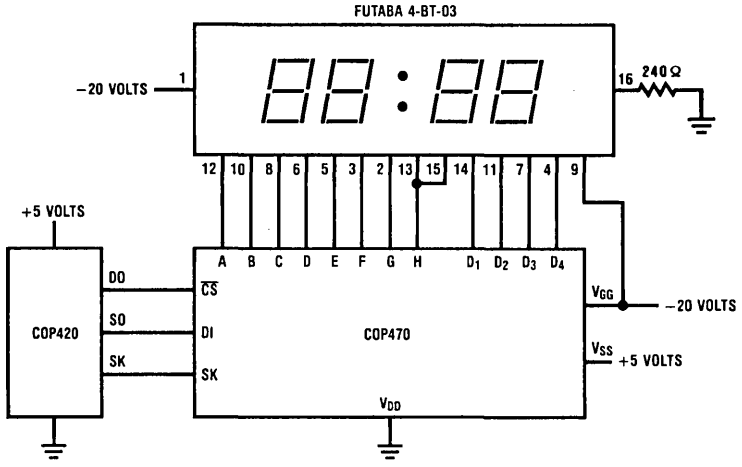
*Segment buffer may be added for larger display.

FIGURE 9. LED Display



TL/DD/6154–11

FIGURE 10. Large LED Display



TL/DD/6154–12

FIGURE 11. Sample V.F. System

National Semiconductor Corporation

COP472-3 Liquid Crystal Display Controller

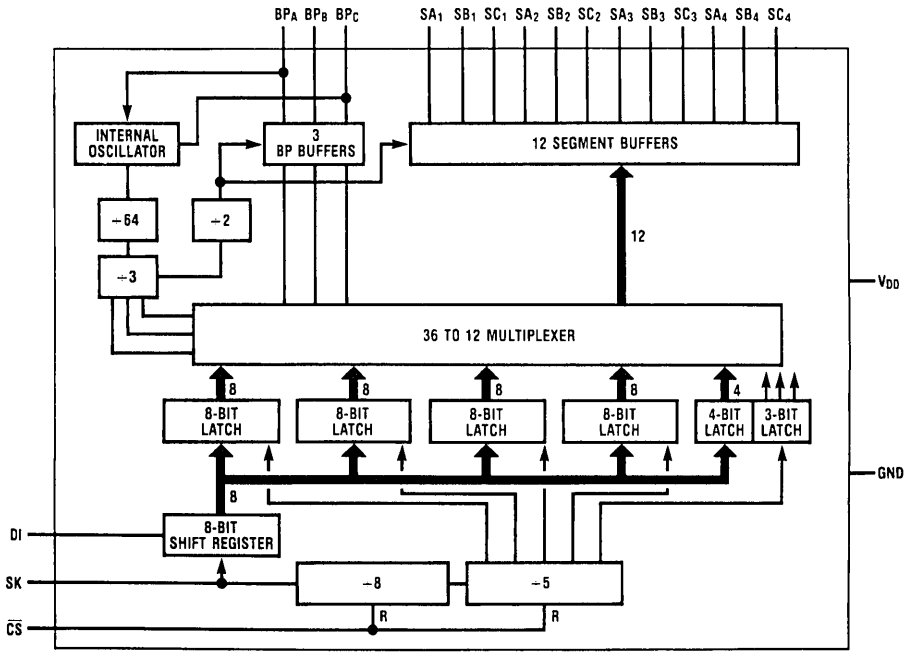
General Description

The COP472-3 Liquid Crystal Display (LCD) Controller is a peripheral member of the COPS™ family, fabricated using CMOS technology. The COP472-3 drives a multiplexed liquid crystal display directly. Data is loaded serially and is held in internal latches. The COP472-3 contains an on-chip oscillator and generates all the multi-level waveforms for backplanes and segment outputs on a triplex display. One COP472-3 can drive 36 segments multiplexed as 3 x 12 (4½ digit display). Two COP472-3 devices can be used together to drive 72 segments (3 x 24) which could be an 8½ digit display.

Features

- Direct interface to TRIPLEX LCD
- Low power dissipation (100 μW typ.)
- Low cost
- Compatible with all COP400 processors
- Needs no refresh from processor
- On-chip oscillator and latches
- Expandable to longer displays
- Software compatible with COP470 V.F. Display Driver chip
- Operates from display voltage
- MICROWIRE™ compatible serial I/O
- 20-pin Dual-In-Line package

Block Diagram



TL/DD/6932-1

Absolute Maximum Ratings

Voltage at CS, DI, SK pins	−0.3V to +9.5V	Storage Temperature	−65°C to +150°C
Voltage at all other Pins	−0.3V to V_{DD}+0.3V	Lead Temp. (Soldering, 10 Seconds)	300°C
Operating Temperature Range	0°C to 70°C		

DC Electrical Characteristics

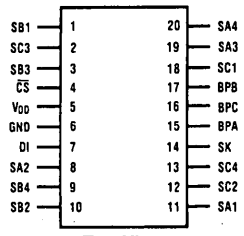
GND = 0V, V_{DD} = 3.0V to 5.5V, T_A = 0°C to 70°C (depends on display characteristics)

Parameter	Conditions	Min	Max	Units
Power Supply Voltage, V_{DD}		3.0	5.5	Volts
Power Supply Current, I_{DD} (Note 1)	V_{DD}=5.5V		250	μA
	V_{DD}=3V		100	μA
Input Levels				
DI, SK, CS				
V_{IL}			0.8	Volts
V_{IH}		0.7 V_{DD}	9.5	Volts
BPA (as Osc. in)				
V_{IL}			0.6	Volts
V_{IH}		V_{DD}−0.6	V_{DD}	Volts
Output Levels, BPC (as Osc. Out)				
V_{OL}			0.4	Volts
V_{OH}		V_{DD}−0.4	V_{DD}	Volts
Backplane Outputs (BPA, BPB, BPC)				
$V_{BPA, BPB, BPC}$ ON	During	$V_{DD}-\Delta V$	V_{DD}	Volts
$V_{BPA, BPB, BPC}$ OFF	BP+ Time	$\frac{1}{3}V_{DD}-\Delta V$	$\frac{1}{3}V_{DD}+\Delta V$	Volts
$V_{BPA, BPB, BPC}$ ON	During	0	ΔV	Volts
$V_{BPA, BPB, BPC}$ OFF	BP− Time	$\frac{2}{3}V_{DD}-\Delta V$	$\frac{2}{3}V_{DD}+\Delta V$	Volts
Segment Outputs (SA_1 ~ SA_4)				
V_{SEG} ON	During	0	ΔV	Volts
V_{SEG} OFF	BP+ Time	$\frac{2}{3}V_{DD}-\Delta V$	$\frac{2}{3}V_{DD}+\Delta V$	Volts
V_{SEG} ON	During	$V_{DD}-\Delta V$	V_{DD}	Volts
V_{SEG} OFF	BP− Time	$\frac{1}{3}V_{DD}-\Delta V$	$\frac{1}{3}V_{DD}+\Delta V$	Volts
Internal Oscillator Frequency		15	80	kHz
Frame Time (Int. Osc. ÷ 192)		2.4	12.8	ms
Scan Frequency ($1/T_{SCAN}$)		39	208	Hz
SK Clock Frequency		4	250	kHz
SK Width		1.7		μs
DI				
Data Setup, t_{SETUP}		1.0		μs
Data Hold, t_{HOLD}		100		ns
$\overline{CS}$				
t_{SETUP}		1.0		μs
t_{HOLD}		1.0		μs
Output Loading Capacitance			100	pF

Note 1: Power supply current is measured in stand-alone mode with all outputs open and all inputs at V_{DD}.

Note 2: $\Delta V = 0.05V_{DD}$.

Dual-In-Line Package



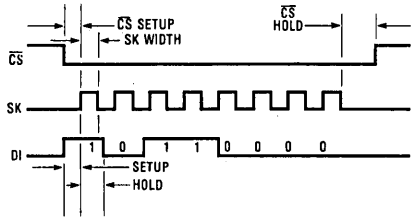
TL/DD/6932-2

Top View

Order Number COP472N-3
See NS Package N20A

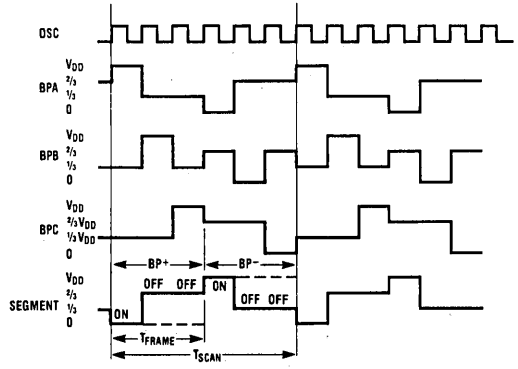
Pin	Description
$\overline{CS}$	Chip select
V_{DD}	Power supply (display voltage)
GND	Ground
DI	Serial data input
SK	Serial clock input
BP_A	Display backplane A (or oscillator in)
BP_B	Display backplane B
BP_C	Display backplane C (or oscillator out)
SA1 ~ SC4	12 multiplexed outputs

FIGURE 2. Connection Diagram



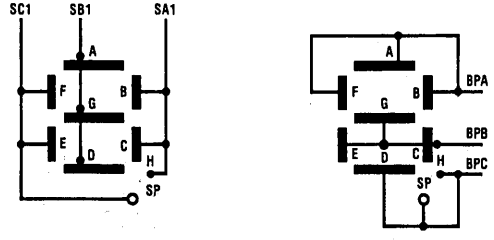
TL/DD/6932-3

FIGURE 3. Serial Load Timing Diagram



TL/DD/6932-4

FIGURE 4. Backplane and Segment Waveforms



TL/DD/6932-5

FIGURE 5. Typical Display Internal Connections
Epson LD-370

Functional Description

The COP472-3 drives 36 bits of display information organized as twelve segments and three backplanes. The COP472-3 requires 40 information bits: 36 data and 4 control. The function of each control bit is described below. Display information format is a function of the LCD interconnections. A typical segment/backplane configuration is illustrated in *Figure 5*, with this configuration the COP472-3 will drive 4 digits of 9 segments.

To adapt the COP472-3 to any LCD display configuration, the segment/backplane multiplex scheme is illustrated in Table I.

Two or more COP472-3 chips can be cascaded to drive additional segments. There is no limit to the number of COP472-3's that can be used as long as the output loading capacitance does not exceed specification.

TABLE I. COP472-3 Segment/Backplane Multiplex Scheme

Bit Number	Segment, Backplane	Data to Numeric Display	
1	SA1, BPC	SH	
2	SB1, BPB	SG	
3	SC1, BPA	SF	
4	SC1, BPB	SE	Digit 1
5	SB1, BPC	SD	
6	SA1, BPB	SC	
7	SA1, BPA	SB	
8	SB1, BPA	SA	
9	SA2, BPC	SH	
10	SB2, BPB	SG	
11	SC2, BPA	SF	
12	SC2, BPB	SE	Digit 2
13	SB2, BPC	SD	
14	SA2, BPB	SC	
15	SA2, BPA	SB	
16	SB2, BPA	SA	
17	SA3, BPC	SH	
18	SB3, BPB	SG	
19	SC3, BPA	SF	
20	SC3, BPB	SE	Digit 3
21	SB3, BPC	SD	
22	SA3, BPB	SC	
23	SA3, BPA	SB	
24	SB3, BPA	SA	
25	SA4, BPC	SH	
26	SB4, BPB	SG	
27	SC4, BPA	SF	
28	SC4, BPB	SE	Digit 4
29	SB4, BPC	SD	
30	SA4, BPB	SC	
31	SA4, BPA	SB	
32	SB4, BPA	SA	
33	SC1, BPC	SPA	Digit 1
34	SC2, BPC	SP2	Digit 2
35	SC3, BPC	SP3	Digit 3
36	SC4, BPC	SP4	Digit 4
37	not used		
38	Q6		
39	Q7		
40	SYNC		

SEGMENT DATA BITS

Data is loaded in serially, in sets of eight bits. Each set of segment data is in the following format:

SA	SB	SC	SD	SE	SF	SG	SH

Data is shifted into an eight bit shift register. The first bit of the data is for segment H, digit 1. The eighth bit is segment A, digit 1. A set of eight bits is shifted in and then loaded into the digit one latches. The second set of 8 bits is loaded into digit two latches. The third set into digit three latches, and the fourth set is loaded into digit four latches.

CONTROL BITS

The fifth set of 8 data bits contains special segment data and control data in the following format:

SYNC	Q7	Q6	X	SP4	SP3	SP2	SP1

The first four bits shifted in contain the special character segment data. The fifth bit is not used. The sixth and seventh bits program the COP472-3 as a stand alone LCD driver or as a master or slave for cascading COP472-3's. BPC of the master is connected to BPA of each slave. The following table summarizes the function of bits six and seven:

Q7	Q6	Function	BPC Output	BPA Output
1	1	Slave	Backplane Output	Oscillator Input
0	1	Stand Alone	Backplane Output	Backplane Output
1	0	Not Used	Internal Osc. Output	Oscillator Input
0	0	Master	Internal Osc. Output	Backplane Output

The eighth bit is used to synchronize two COP472-3's to drive an 8½-digit display.

LOADING SEQUENCE TO DRIVE A 4½-DIGIT DISPLAY

Steps:

1. Turn $\overline{CE}$ low.
2. Clock in 8 bits of data for digit 1.
3. Clock in 8 bits of data for digit 2.
4. Clock in 8 bits of data for digit 3.
5. Clock in 8 bits of data for digit 4.
6. Clock in 8 bits of data for special segment and control function of BPC and BPA.

0	0	1	1	SP4	SP3	SP2	SP1

7. Turn $\overline{CS}$ high.

Note: $\overline{CS}$ may be turned high after any step. For example to load only 2 digits of data, do steps 1, 2, 3, and 7.

$\overline{CS}$ must make a high to low transition before loading data in order to reset internal counters.

LOADING SEQUENCE TO DRIVE AN 8½-DIGIT DISPLAY

Two or more COP472-3's may be connected together to drive additional segments. An eight digit multiplexed display is shown in *Figure 7*. The following is the loading sequence to drive an eight digit display using two COP472-3's. The right chip is the master and the left the slave.

Steps:

1. Turn $\overline{CS}$ low on both COP472-3's.
2. Shift in 32 bits of data for the slave's four digits.
3. Shift in 4 bits of special segment data: a zero and three ones.

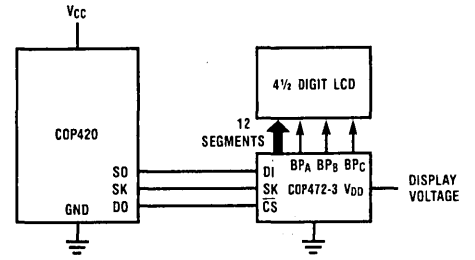
1	1	1	0	SP4	SP3	SP2	SP1

   This synchronizes both the chips and BPA is oscillator input. Both chips are now stopped.
4. Turn CS high to both chips.
5. Turn CS low to master COP472-3.
6. Shift in 32 bits of data for the master's 4 digits.
7. Shift in four bits of special segment data, a one and three zeros.

0	0	0	1	SP4	SP3	SP2	SP1

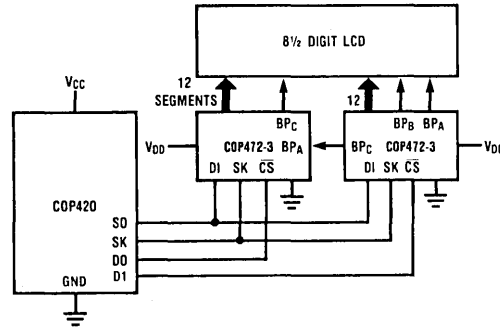
   This sets the master COP472-3 to BPA as a normal backplane output and BPC as oscillator output. Now both the chips start and run off the same oscillator.
8. Turn $\overline{CS}$ high.

The chips are now synchronized and driving 8 digits of display. To load new data simply load each chip separately in the normal manner, keeping the correct status bits to each COP472-3 (0110 or 0001).



TL/DD/6932-6

FIGURE 6. System Diagram – 4½ Digit Display



TL/DD/6932-7

FIGURE 7. System Diagram – 8½ Digit Display

Example Software

Example 1

COP420 Code to load a COP472-3 [Display data is in M(0, 12)-M(0, 15), special segment data is in M(0, 0)]

```
                LBI 0, 12          ; POINT TO FIRST DISPLAY DATA
                OBD                ; TURN CS LOW (DO)
LOOP:           CLRA
                LQID               ; LOOK UP SEGMENT DATA
                CQMA               ; COPY DATA FROM Q TO M & A
                SC                 ; SET C TO TURN ON SK
                XAS                ; OUTPUT LOWER 4 BITS OF DATA
                NOP                ; DELAY
                NOP                ; DELAY
                LD                 ; LOAD A WITH UPPER 4 BITS
                XAS                ; OUTPUT 4 BITS OF DATA
                NOP                ; DELAY
                NOP                ; DELAY
                RC                 ; RESET C
                XAS                ; TURN OFF SK CLOCK
                XIS                ; INCREMENT B FOR NEXT DATA
                JP LOOP            ; SKIP THIS JUMP AFTER LAST DIGIT
                SC                 ; SET C
                LBI 0, 0           ; ADDRESS SPECIAL SEGMENTS
                LD                 ; LOAD INTO A
                XAS                ; OUTPUT SPECIAL SEGMENTS
                NOP                ;
                CLRA               ;
                AISC 12            ; 12 to A
                XAS                ; OUTPUT CONTROL BITS
                NOP                ;
                LBI 0, 15          ; 15 to B
                RC                 ; RESET C
                XAS                ; TURN OFF SK
                OBD                ; TURN CS HIGH (DO)
```

Example Software (Continued)

Example 2

COP420 Code to load two COP472-3 parts [Display data is in M(0, 12)-M(0, 15) and M(1, 12)-M(1, 15), special segment data is in M(0, 0) and M(1, 0)]

```
INIT:           LBI     0, 15
                OBD                     ; TURN BOTH CS'S HIGH
                LEI     8               ; ENABLE SO OUT OF S. R.
                RC
                XAS                     ; TURN OFF SK CLOCK
                LBI     3, 15           ; USE M(3, 15) FOR CONTROL BITS
                STII    7               ; STORE 7 TO SYNC BOTH CHIPS
                LBI     0, 12           ; SET B TO TURN BOTH CS'S LOW
                JSR     OUT             ; CALL OUTPUT SUBROUTINE

MAIN DISPLAY SEQUENCE
DISPLAY         LBI     3, 15
                STII    8               ; SET CONTROL BITS FOR SLAVE
                LBI     0, 13           ; SET B TO TURN SLAVE CS LOW
                JSR     OUT             ; OUTPUT DATA FROM REG. 0
                LBI     3, 15
                STII    6               ; SET CONTROL BITS FOR MASTER
                LBI     1, 14           ; SET B TO TURN MASTER CS LOW
                JSR     OUT             ; OUTPUT DATA FROM REG. 1

OUTPUT SUBROUTINE
OUT:            OBD                     ; OUTPUT B TO CS'S
                CLRA
                AISC    12              ; 12 TO A
                CAB                     ; POINT TO DISPLAY DIGIT (BD = 12)
LOOP            CLRA
                LQID                    ; LOOK UP SEGMENT DATA
                CQMA                    : COPY DATA FROM Q TO M & A
                SC
                XAS                     ; OUTPUT LOWER 4 BITS OF DATA
                NOP                     ; DELAY
                NOP                     ; DELAY
                LD                      ; LOAD A WITH UPPER 4 BITS
                XAS                     ; OUTPUT 4 BITS OF DATA
                NOP                     ; DELAY
                NOP                     ; DELAY
                RC                      ; RESET C
                XAS                     ; TURN OFF SK
                XIS                     ; INCREMENT B FOR NEXT DISPLAY DIGIT
                JP      LOOP            ; SKIP THIS JUMP AFTER LAST DIGIT
                SC                      ; SET C
                NOP
                LD                      ; LOAD SPECIAL SEGS. TO A (BD = 0)
                XAS                     ; OUTPUT SPECIAL SEGMENTS
                NOP
                LBI     3, 15
                LD                      ; LOAD A
                XAS                     ; OUTPUT CONTROL BITS
                NOP
                NOP
                RC
                XAS                     ; TURN OFF SK
                OBD                     ; TURN CS'S HIGH (BD = 15)
                RET
```

National Semiconductor Corporation

NMC9306/COP494 256-Bit Serial Electrically Erasable Programmable Memory

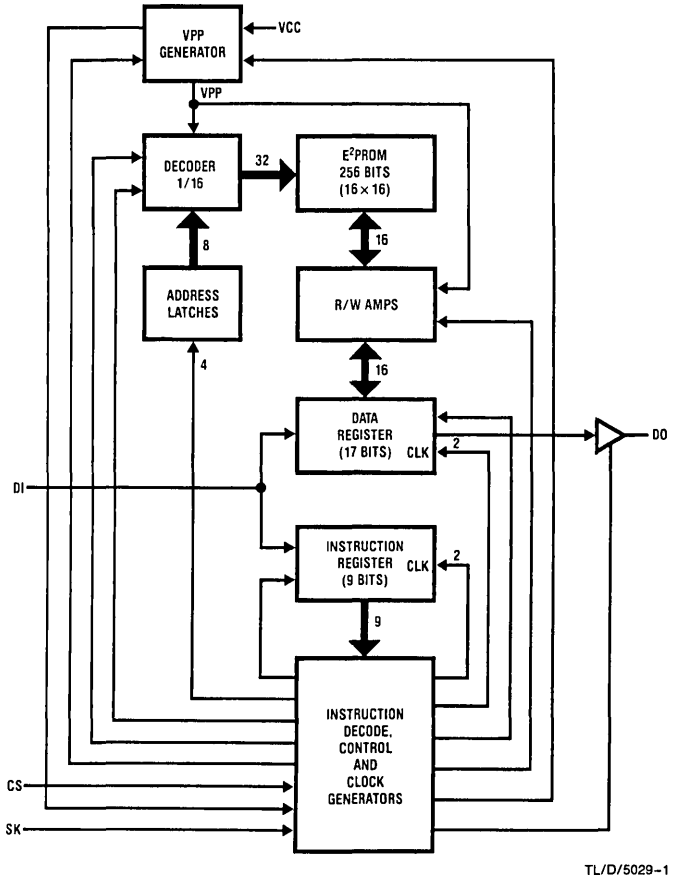
General Description

The NMC9306/COP494 is a 256-bit non-volatile sequential access memory fabricated using advanced floating gate N-channel E^2PROM technology. It is a peripheral memory designed for data storage and/or timing and is accessed via the simple MICROWIRE™ serial interface. The device contains 256 bits of read/write memory divided into 16 registers of 16 bits each. Each register can be serially read or written by a COP400 series controller. Written information is stored in a floating gate cell with at least 10 years data retention and can be updated by an erase-write cycle. The NMC9306/COP494 has been designed to meet applications requiring up to 1×10^4 erase/write cycles per register. A power down mode reduces power consumption by 70 percent.

Features

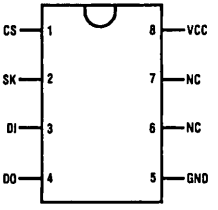
- Low cost
- Single supply operation (5V ± 10%)
- TTL compatible
- 16×16 serial read/write memory
- MICROWIRE compatible serial I/O
- Compatible with COP400 processors
- Low standby power
- Non-volatile erase and write
- Reliable floating gate technology

Block and Connection Diagrams



TL/D/5029-1

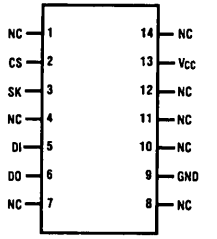
Dual-In-Line Package



TL/D/5029-10

Top View

SO Package



TL/D/5029-2

Top View

Order Number NMC9306N, NMC9306 or COP494N
See NS Package N08E or M14B

Pin Names

CS	Chip Select
SK	Serial Data Clock
DI	Serial Data Input
DO	Serial Data Output
V_{CC}	Power Supply
GND	Ground

Absolute Maximum Ratings

Voltage Relative to GND	+6V to −0.3V
Ambient Operating Temperature	
NMC9306/COP494	0°C to +70°C
Ambient Storage Temperature	
with Data Retention	−65°C to +125°C
Lead Temperature (Soldering, 10 seconds)	300°C

Note: *Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

Electrical Characteristics 0°C ≤ TA ≤ 70°C, V_{CC} = 5V ± 10% unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
Operating Voltage (V_{CC})		4.5		5.5	V
Operating Current (I_{CC1})	V_{CC} = 5.5V, CS = 1			10	mA
Standby Current (I_{CC2})	V_{CC} = 5.5V, CS = 0			3	mA
Input Voltage Levels					
V_{IL}		−0.1		0.8	V
V_{IH}		2.0		V_{CC} + 1	V
Output Voltage Levels					
V_{OL}	I_{OL} = 2.1 mA			0.4	V
V_{OH}	I_{OH} = −400 μA	2.4			V
Input Leakage Current	V_{IN} = 5.5V			10	μA
Output Leakage Current	V_{OUT} = 5.5V, CS = 0			10	μA
SK Frequency		0		250	kHz
SK HIGH TIME t_{SKH} (Note 2)		1			μs
SK LOW TIME t_{SKL} (Note 2)		1			μs
Input Set-Up and Hold Times					
CS t_{CSS}		0.2			μs
t_{CSH}		0			μs
DI t_{DIS}		0.4			μs
t_{DIH}		0.4			μs
Output Delay	CL = 100 pF				
DO t_{PD1}	V_{OL} = 0.8V, V_{OH} = 2.0V			2	μs
t_{PD0}	V_{IL} = 0.45V, V_{IH} = 2.40V			2	μs
Erase/Write Pulse Width ($t_{E/W}$) (Note 1)		10		30	ms
CS Low Time (t_{CS}) (Note 3)		1			μs

Note 1: $t_{E/W}$ measured to rising edge of SK or CS, whichever occurs last.

Note 2: The SK frequency spec. specifies a minimum SK clock period of 4 μs, therefore in an SK clock cycle, t_{SKH} + t_{SKL} must be greater than or equal to 4 μs. e.g. if t_{SKL} = 1μs then the minimum t_{SKH} = 3 μs in order to meet the SK frequency specification.

Note 3: CS must be brought low for a minimum of 1 μs (t_{CS}) between consecutive instruction cycles.

Instruction Set

Instruction	SB	Op Code	Address	Data	Comments
READ	1	10xx	A3A2A1A0		Read register A3A2A1A0
WRITE	1	01xx	A3A2A1A0	D15−D0	Write register A3A2A1A0
ERASE	1	11xx	A3A2A1A0		Erase register A3A2A1A0
EWEN	1	0011	xxxx		Erase/write enable
EWDS	1	0000	xxxx		Erase/write disable
ERAL	1	0010	xxxx		Erase all registers
WRAL	1	0001	xxxx	D15−D0	Write all registers

NMC9306/COP494 has 7 instructions as shown. Note that MSB of any given instruction is a "1" and is viewed as a start bit in the interface sequence. The next 8 bits carry the op code and the 4-bit address for 1 of 16, 16-bit registers.

X is a don't care state.

Functional Description

The NMC9306/COP494 is a small peripheral memory intended for use with COPS™ controllers and other non-volatile memory applications. Its organization is sixteen registers and each register is sixteen bits wide. The input and output pins are controlled by separate serial formats. Seven 9-bit instructions can be executed. The instruction format has a logical '1' as a start bit, four bits as an op code, and four bits of address. SK clock cycle is necessary after CS equals logical "1" before the instruction can be loaded. The on-chip programming-voltage generator allows the user to use a single power supply (V_{CC}). Only during the read mode is the serial output (DO) pin valid. During all other modes the DO pin is in TRI-STATE®, eliminating bus contention.

READ

The read instruction is the only instruction which outputs serial data on the DO pin. after a READ instruction is received, the instruction and address are decoded, followed by data transfer from the memory register into a 16-bit serial-out shift register. A dummy bit (logical '0') precedes the 16-bit data output string. Output data changes are initiated by a low to high transition of the SK clock.

ERASE/WRITE ENABLE AND DISABLE

Programming must be preceded once by a programming enable (EWEN) instruction. Programming remains enabled until a programming disable (EWDS) instruction is executed. The programming disable instruction is provided to protect against accidental data disturb. Execution of a READ instruction is independent of both EWEN and EWDS instructions.

ERASE (Note 4)

Like most E²PROMS, the register must first be erased (all bits set to 1s) before the register can be written (certain bits set to 0s). After an ERASE instruction is input, CS is dropped low. This falling edge of CS determines the start of programming. The register at the address specified in the instruction is then set entirely to 1s. When the erase/write programming time ($t_{E/W}$) constraint has been satisfied, CS is brought up for at least one SK period. A new instruction may then be input, or a low-power standby state may be achieved by dropping CS low.

WRITE (Note 4)

The WRITE instruction is followed by 16 bits of data which are written into the specified address. This register must have been previously erased. Like any programming mode, erase/write time is determined by the low state of CS following the instruction. The on-chip high voltage section only generates high voltage during these programming modes, which prevents spurious programming during other modes. When CS rises to V_{IH}, the programming cycle ends. All programming modes should be ended with CS high for one SK period, or followed by another instruction.

CHIP ERASE (Note 4)

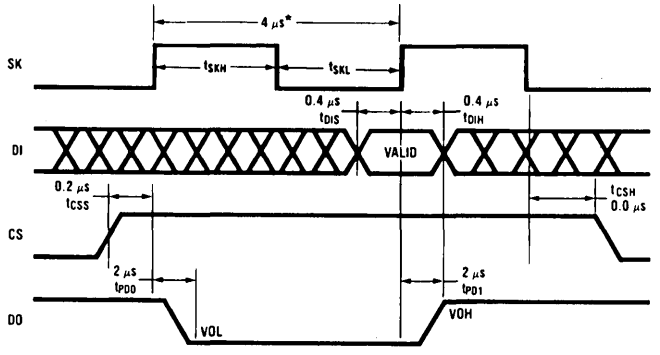
Entire chip erasing is provided for ease of programming. Erasing the chip means that all registers in the memory array have each bit set to a 1. Each register is then ready for a WRITE instruction.

CHIP WRITE (Note 4)

All registers must be erased before a chip write operation. The chip write cycle is identical to the write cycle, except for the different op code. All registers are simultaneously written with the data pattern specified in the instruction.

Note 4: During a programming mode (write, erase, chip erase, chip write), SK clock is only needed while the actual instruction, i.e., start bit, op code, address and data, is being input. It can remain deactivated during the Erase/Write pulse width ($t_{E/W}$).

Timing Diagrams

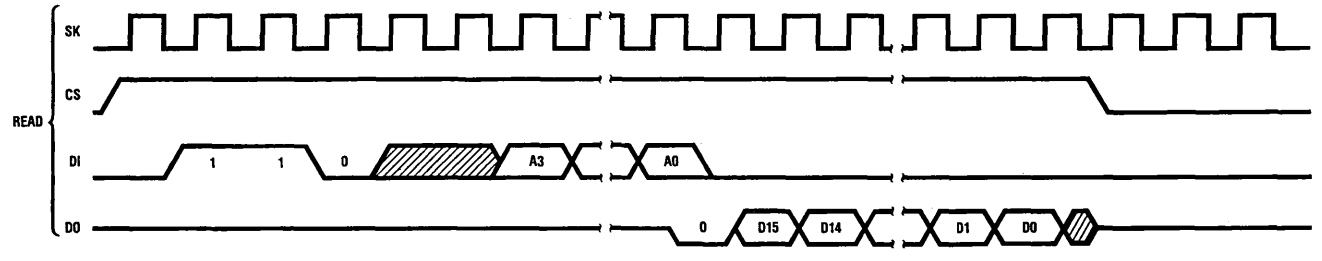


TL/D/5029-3

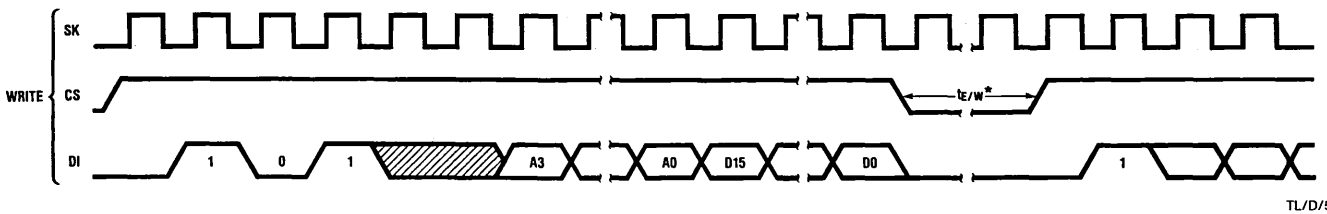
*This is the minimum SK period

Synchronous Data Timing

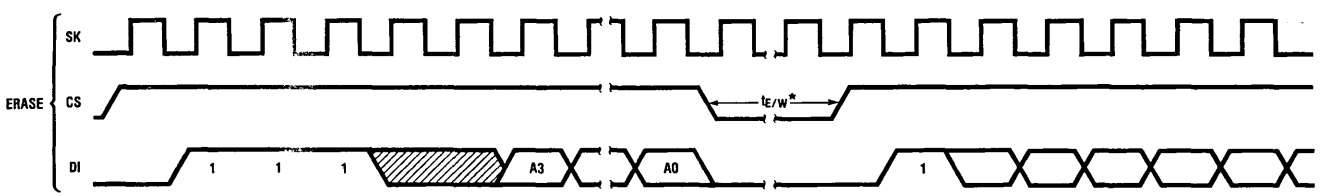
Timing Diagrams (Continued)



TL/D/5029–4



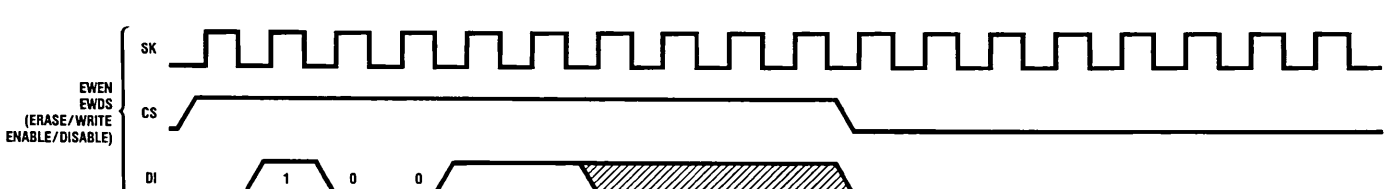
TL/D/5029–5



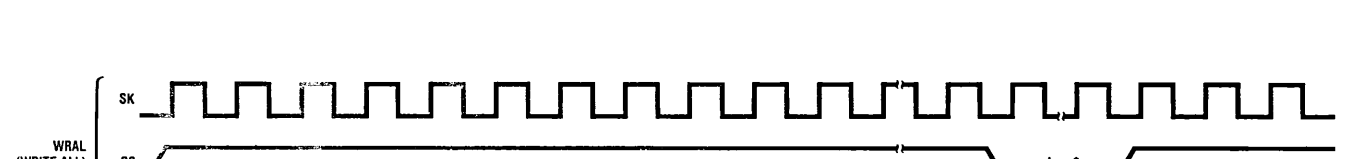
TL/D/5029–6

*$t_{E/W}$ measured to rising edge of SK or CS, whichever occurs last.

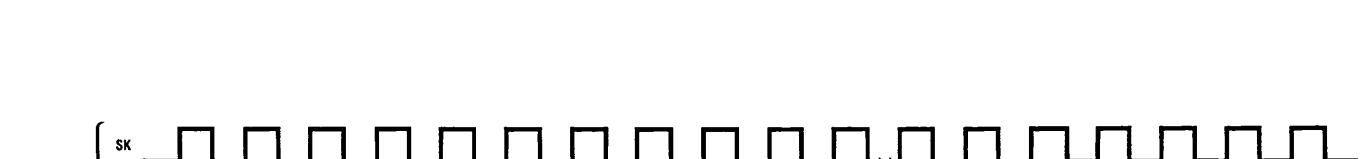
Instruction Timing



TL/D/5029–7



TL/D/5029–8



TL/D/5029–9

*$t_{E/W}$ measured to rising edge of SK or CS, whichever occurs last.

Instruction Timing (Continued)

National Semiconductor Corporation

NMC9345/COP495 1024-Bit Serial Electrically Erasable Programmable Memory (5V Only)

General Description

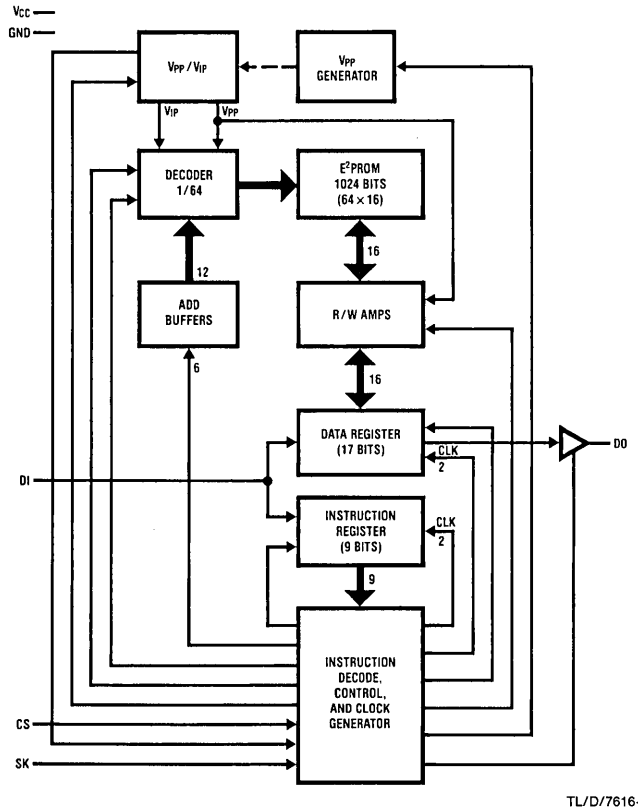
The NMC9345/COP495 is a 1024-bit non-volatile, sequential E²PROM, fabricated using advanced N-channel E²PROM technology. It is an external memory with the 1024 bits of read/write memory divided into 64 registers of 16 bits each. Each register can be serially read or written by a COP400 controller, or a standard microprocessor. Written information is stored in a floating gate cell until updated by an erase and write cycle. The NMC9345 has been designed for applications requiring up to 10^4 erase/write cycles per register. A power-down mode is provided by CS to reduce power consumption by 75 percent.

Features

- Low cost
- Single supply read/write/erase operations (5V ± 10%)
- TTL compatible
- 64×16 serial read/write memory
- MICROWIRE™ compatible serial I/O
- Simple interfacing
- Low standby power
- Non-volatile erase and write
- Reliable floating gate technology
- Self-timed programming cycle
- Device status signal during programming

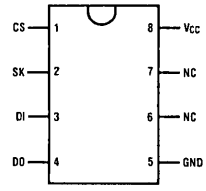
Block and Connection Diagrams

Serial E²PROM



TL/D/7616-1

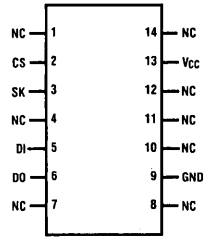
Dual-In-Line Package



TL/D/7616-2

Top View

SO Package



TL/D/7616-10

Top View

Order Number NMC9345N, NMC9345 or COP495N
See NS Package Number N08E or M14B

Pin Names

CS	Chip Select
SK	Serial Data Clock
DI	Serial Data Input
DO	Serial Data Output
V_{CC}	Power Supply
GND	Ground
NC	Not Connected

Absolute Maximum Ratings (Note 1)

Voltage Relative to GND	+6V to −0.3V	Ambient Storage Temperature	−65°C to +125°C
Ambient Operating Temperature	0°C to +70°C	Lead Temp. (Soldering, 10 seconds)	300°C

DC and AC Electrical Characteristics NMC9345: $0°C \le T_A \le 70°C$, $V_{CC} = 5V \pm 10\%$ unless specified

Symbol	Parameter	Conditions	Min	Max	Units
V_{CC}	Operating Voltage		4.5	5.5	V
I_{CC1}	Operating Current Erase/Write Operating Current	$V_{CC} = 5.5V$, CS = 1, SK = 1 $V_{CC} = 5.5V$		12 12	mA mA
I_{CC2}	Standby Current	$V_{CC} = 5.5V$, CS = 0		3	mA
	Input Voltage Levels				
V_{IL}			−0.1	0.8	V
V_{IH}			2.0	$V_{CC}+1$	V
	Output Voltage Levels				
V_{OL}		$I_{OL} = 2.1$ mA		0.4	V
V_{OH}		$I_{OH} = -400\ \mu A$	2.4		V
I_{LI}	Input Leakage Current	$V_{IN} = 5.5V$		10	μA
I_{LO}	Output Leakage Current	$V_{OUT} = 5.5V$, CS = 0		10	μA
	SK Frequency		0	250	kHz
t_{SKH}	SK High Time		2		μs
t_{SKL}	SK Low Time		1		μs
	Inputs				
t_{CSS}	CS		0.2		μS
t_{CSH}			0		μS
t_{DIS}	DI		0.4		μS
t_{DIH}			0.4		μS
	Output	$C_L = 100$ pF			
$t_{pd}1$	DO	$V_{OL} = 0.8V$, $V_{OH} = 2.0V$		2	μS
$t_{pd}0$		$V_{IL} = 0.45V$, $V_{IH} = 2.40V$		2	μS
$t_{E/W}$	Self-Timed Program Cycle			10	ms
t_{CS}	Min CS Low Time (Note 3)		1		μS
t_{SV}	Rising Edge of CS to Status Valid	$C_L = 100$ pF		1	μS
t_{0H}, t_{1H}	Falling Edge of CS to DO TRI-STATE®			0.4	μS

Note 1: Stress above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of the specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Note 2: The SK frequency spec. specifies a minimum SK clock period of 4 μs, therefore in an SK clock cycle $t_{SKH} + t_{SKL}$ must be greater than or equal to 4 μs. e.g. if $t_{SKL} = 1$ μs then the minimum $t_{SKH} = 3$ μs in order to meet the SK frequency specification.

Note 3: CS must be brought low for a minimum of 1μS (t_{CS}) between consecutive instruction cycles.

Functional Description

The NMC9345/COP495 is a small peripheral memory intended for use with COPS™ controllers and other non-volatile memory applications. Its organization is sixty-four registers and each register is sixteen bits wide. The input and output pins are controlled by separate serial formats. Seven 9-bit instructions can be executed. The instruction format has a logical '1' as a start bit, two bits as an op code, and six bits of address. The programming cycle is self-timed, with the data out (DO) pin indicating the read/busy status of the chip.

The on-chip programming voltage generator allows the user to use a single power supply (V_{CC}). It only generates high voltage during the programming modes (write, erase, chip erase, chip write) to prevent spurious programming during other modes. The DO pin is valid as data out during the read mode, and if initiated, as a ready/busy status indicator during a programming cycle. During all other modes the DO pin is in TRI-STATE, eliminating bus contention.

Functional Description (Continued)

READ

The read instruction is the only instruction which outputs serial data on the DO pin. After a read instruction is received, the instruction and address are decoded, followed by data transfer from the memory register into a 16-bit serial-out shift register. A dummy bit (logical '0') precedes the 16-bit data output string. Output data changes are initiated by a low to high transition of the SK clock.

ERASE/WRITE ENABLE AND DISABLE

When V_{CC} is applied to the part it powers up in the programming disable (EWDS) state, programming must be preceded by a programming enable (EWEN) instruction. Programming remains enabled until a programming disable (EWDS) instruction is executed or V_{CC} is removed from the part. The programming disable instruction is provided to protect against accidental data disturb. Execution of a read instruction is independent of both EWEN and EWDS instructions.

ERASE (Note 4)

Like most E²PROMs, the register must first be erased (all bits set to logical '1') before the register can be written (certain bits set to logical '0'). After an erase instruction is input, CS is dropped low. This falling edge of CS determines the start of the self-timed programming cycle. If CS is brought high subsequently (after observing the t_{CS} specification), the DO pin will indicate the ready/busy status of the chip. The DO pin will go low if the chip is still programming. The DO pin will go high when all bits of the register at the address specified in the instruction have been set to a logical '1'. The part is now ready for the next instruction sequence.

WRITE (Note 4)

The write instruction is followed by 16 bits of data to be written into the specified address. After the last bit of data (D0) is put on the data in (DI) pin CS must be brought low before the next rising edge of the SK clock. This falling edge of CS initiates the self-timed programming cycle. Like all programming modes, DO indicates the ready/busy status of the chip if CS is brought high after a minimum of 1 μS (t_{CS}). DO = logical '0' indicates that programming is still in progress. DO = logical '1' indicates that the register at the address specified in the instruction has been written with the data pattern specified in the instruction and the part is ready for another instruction. The register to be written into must have been previously erased.

CHIP ERASE (Note 4)

Entire chip erasing is provided for ease of programming. Erasing the chip means that all registers in the memory array have each bit set to a logical '1'. Each register is then ready for a write instruction. The chip erase cycle is identical to the erase cycle except for the different op code.

CHIP WRITE (Note 4)

All registers must be erased before a chip write operation. The chip write cycle is identical to the write cycle except for the different op code. All registers are simultaneously written with the data pattern specified in the instruction.

Note 4: During a programming mode (write, erase, chip erase, chip write), SK clock is only needed while the actual instruction, i.e., start bit, op code, address and data, is being input. It can remain deactivated during the self-timed programming cycle and status check.

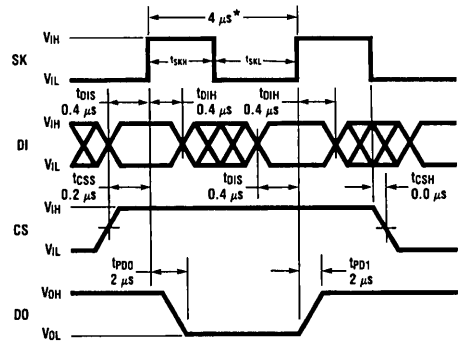
IC INSTRUCTION SET FOR NMC9345/COP495

Instruction	SB	Opcode	Address	Data	Comments
READ	1	10	A5A4A3A2A1A0		Read register A5A4A3A2A1A0
WRITE	1	01	A5A4A3A2A1A0	D15-D0	Write register A5A4A3A2A1A0
ERASE	1	11	A5A4A3A2A1A0		Erase register A5A4A3A2A1A0
EWEN	1	00	11xxxx		Erase/Write enable
EWDS	1	00	00xxxx		Erase/Write disable
ERAL	1	00	10xxxx		Erase all registers
WRAL	1	00	01xxxx	D15-D0	Write all registers

NMC9345/COP495 has 7 instructions as shown. Note that the MSB of any given instruction is a "1" and is viewed as a start bit in the interface sequence. The next 8 bits carry the op code and the 6-bit address for 1 of 64, 16-bit registers.

Timing Diagrams

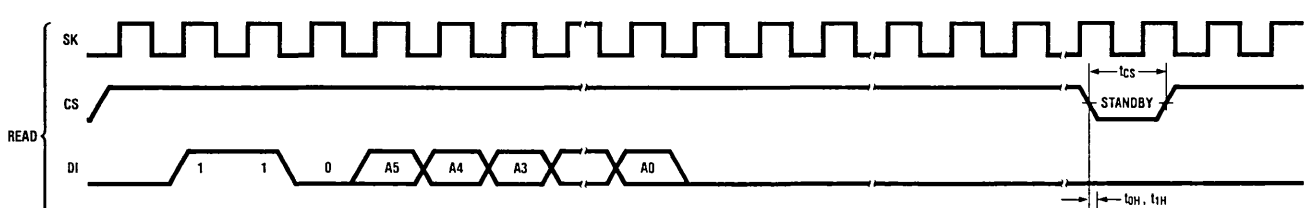
Synchronous Data Timing



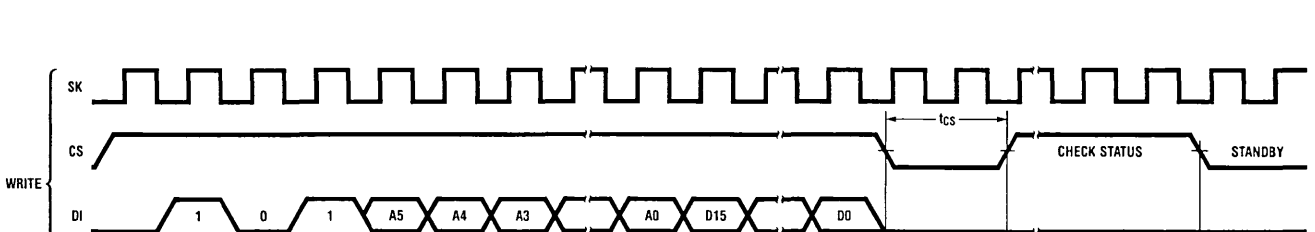
*This is the minimum SK period.

TL/D/7616–3

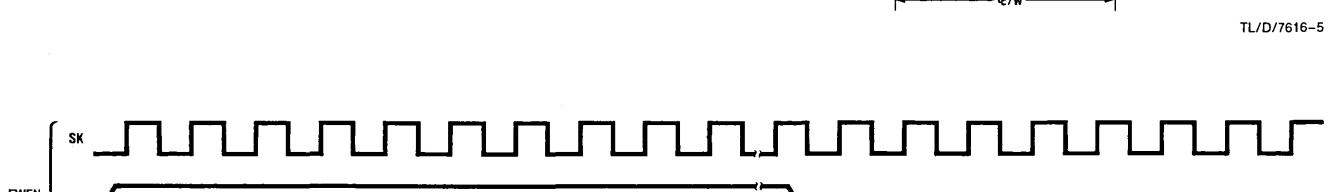
Instruction Timing



TL/D/7616–4



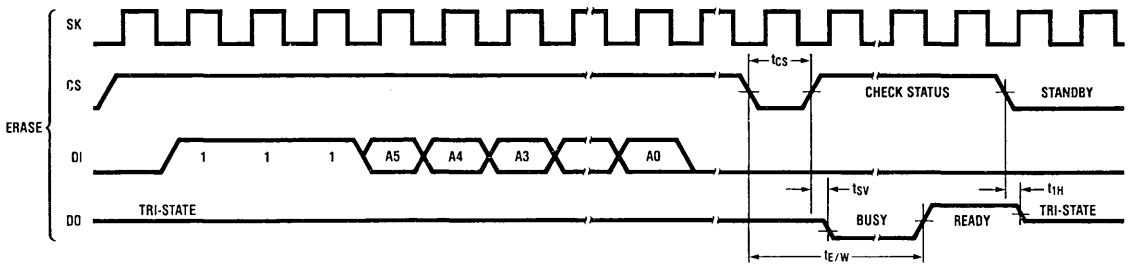
TL/D/7616–5



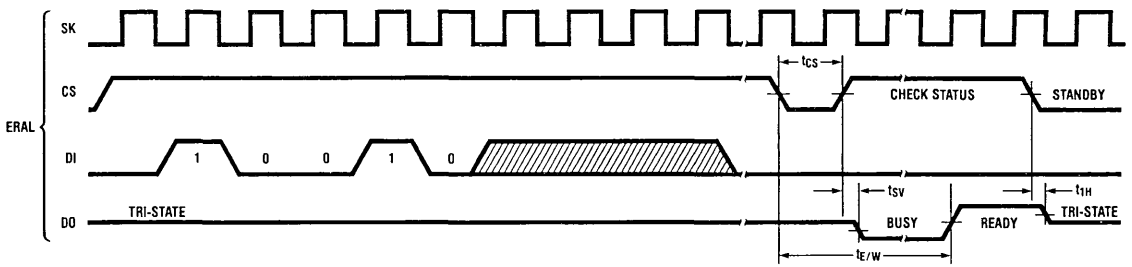
TL/D/7616–6

Timing Diagrams (Continued)

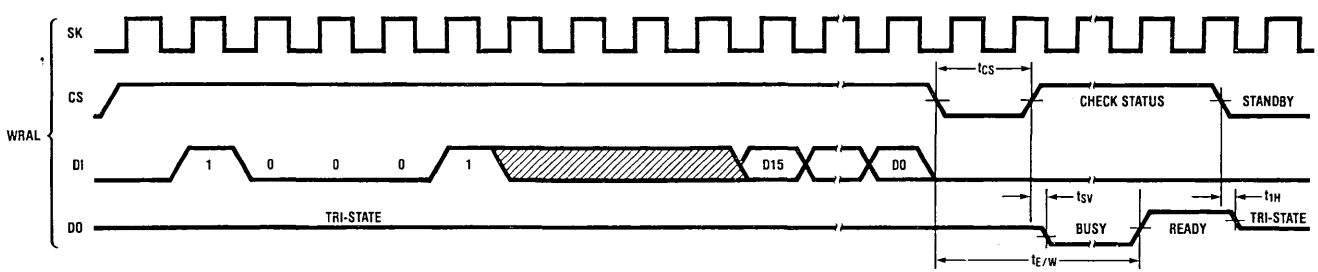
Instruction Timing



TL/D/7616–7



TL/D/7616–8



TL/D/7616–9

National Semiconductor Corporation

COP498/COP398 Low Power CMOS RAM and Timer (RAT™) COP499/COP399 Low Power CMOS Memory

General Description

The COP498/398 Low Power CMOS RAM and Timer (RAT) and the COP499/399 Memory are peripheral members of the COPS™ family, fabricated using low power CMOS technology. These devices provide external data storage and/or timing, and are accessed via the simple MICROWIRE™ serial interface. Each device contains 256 bits of read/write memory organized into 4 registers of 64 bits each; each register can be serially loaded or read by a COPS controller.

The COP498/398 also contain a crystal-based timer for timekeeping purposes, and can provide a "wake-up" signal to turn on a COPS controller. Hence, these devices are ideal for applications requiring very low power drain in a standby mode, while maintaining a real-time clock (e.g., electronically-tuned automobile radio). Power is minimized by cycling controller power off for periods of time when no processing is required.

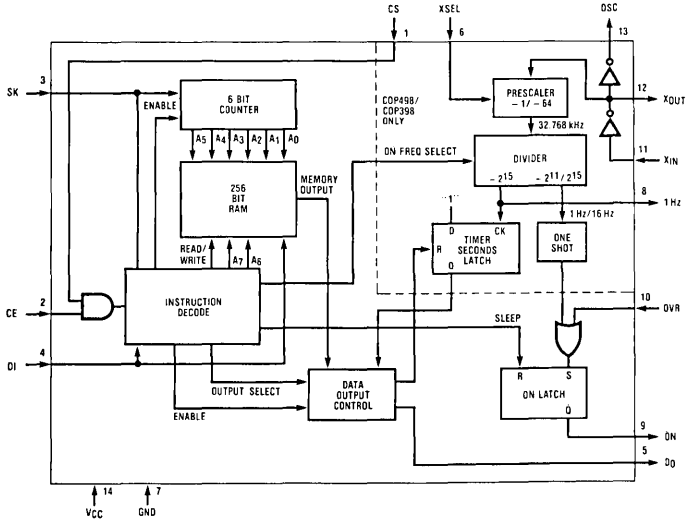
The COP499/399 contain circuitry that enables the user to turn a controller on and off while maintaining the integrity of the memory.

A COP400 series N-channel microcontroller coupled with a COP498 (or 499) RAM/Timer offers a user the low-power advantages of an all CMOS system and the low-cost advantage of an NMOS system. This type of system is ideally suited to a wide variety of automotive and instrumentation applications.

Features

- Low power dissipation
- Quiescent current = 40 nA typical (25°C, V_{CC} = 3.0V)
- Low cost
- Single supply operation (2.4V–5.5V)
- CMOS-compatible I/O
- 4 x 64 serial read/write memory
- Crystal-based selectable timer—2.097152 MHz or 32.768 kHz (COP498/398)
- Software selectable 1 Hz or 16 Hz "wake-up" signal for COPS controller (COP498/398)
- External override to "wake-up" controller
- Compatible with all COP400 processors (processor $V_{CC} \leq 9.5V$)
- MICROWIRE-compatible serial I/O
- Memory protection with write enable and write disable instructions
- 14-pin Dual-In-Line package (COP498/398) or 8-pin Dual-In-Line package (COP499/399)

Block Diagram



TL/DD/6684–1

FIGURE 1

Absolute Maximum Ratings

Voltage relative to GND	
At XSEL, 1 Hz, X_{IN}, X_{OUT}, DO	−0.3V to V_{CC} +0.3V
At all other pins	−0.3V to 10V
Maximum V_{CC} Voltage	6.5V
Total Sink Current Allowed	15 mA
Total Source Current Allowed	10 mA
Ambient Operating Temperature	
COP398/COP399	−40°C to +85°C
COP498/COP499	0°C to +70°C
Ambient Storage Temperature	−65°C to +150°C
Lead Temp. (Soldering, 10 seconds)	300°C
Power Dissipation	50 mW

Note: *"Absolute maximum ratings" indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not insured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics

COP398/COP399: −40°C ≤ T_A ≤ +85°C unless otherwise specified.
COP498/COP499: 0°C ≤ T_A ≤ +70°C unless other wise specified.

Parameter	Conditions	Min	Max	Units
Operating Voltage	COP498/COP499	2.4	5.5	V
	COP398/COP399	3.0	5.5	V
Quiescent Current	All inputs at GND			
	T_A = 25°C, V_{CC} = 3.0V		1.0	μA
	T_A = 25°C, V_{CC} = 5.0V		3.0	μA
	T_A = 25°C, V_{CC} = 5.5V		6.0	μA
	T_A = 70°C, V_{CC} = 3.0V		4.0	μA
	T_A = 70°C, V_{CC} = 5.0V		10	μA
	T_A = 70°C, V_{CC} = 5.5V		20	μA
(COP398/COP399 only)	T_A = 85°C, V_{CC} = 3.0V		8.0	μA
	T_A = 85°C, V_{CC} = 5.0V		16	μA
	T_A = 85°C, V_{CC} = 5.5V		30	μA
COP498/COP398				
Standby Current (sleep mode)	V_{CC} = Min., Osc. = 2.097 MHz		200	μA
(running with crystal	V_{CC} = Max., Osc. = 2.097 MHz		700	μA
	V_{CC} = Min., Osc. = 32.768 kHz		20	μA
	V_{CC} = Max., Osc. = 32.768 kHz		100	μA
Operating Current	SK = 250 kHz square wave			
	V_{CC} = Min., Osc. = 2.097 MHz		300	μA
	V_{CC} = Max., Osc. = 2.097 MHz		920	μA
	V_{CC} = Min., Osc. = 32.768 kHz		120	μA
	V_{CC} = Max., Osc. = 32.768 kHz		320	μA
COP499/COP399 Operating Current	SK = 250 kHz square wave			
	V_{CC} = 2.4V for COP498/COP499		100	μA
	V_{CC} = 3.0V for COP398/COP399		140	μA
	V_{CC} = Max.		250	μA
Input Voltage Levels				
CE Input	(Schmitt Trigger Input)			
Logic High (V_{IH})		0.8V_{CC}		V
Logic Low (V_{IL})			0.4V_{CC}	V
OVR Input	(Schmitt Trigger Input)			
Logic High (V_{IH})		0.8V_{CC}		V
Logic Low (V_{IL})			0.2V_{CC}	V
All Other Inputs				
Logic High (V_{IH})		0.7V_{CC}		V
Logic Low (V_{IL})			0.3V_{CC}	V
Output Voltage Levels—DO, 1 Hz				
CMOS Operation				
Logic High (V_{OH})	I_{OH} = −10 μA	V_{CC}−0.1		V
Logic Low (V_{OL})	I_{OL} = 10 μA		0.1	V

DC Electrical Characteristics (Continued)

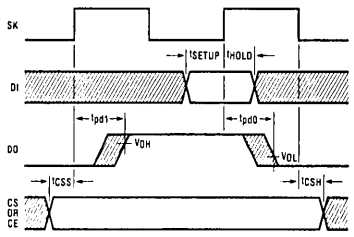
Parameter	Conditions	Min	Max	Units
Input Leakage Current	COP498/COP499, $V_{IH} = V_{CC}$, $V_{IL} = 0V$	−1.0	+1.0	μA
	COP398/COP399, $V_{IH} = V_{CC}$, $V_{IL} = 0V$	−2.0	+2.0	μA
TRI-STATE®, Open Drain	COP498/COP499, $V_H = V_{CC}$, $V_L = 0V$	−2.5	+2.5	μA
Leakage Current	COP398/COP399, $V_H = V_{CC}$, $V_L = 0V$	−5.0	+5.0	μA
Output Current Levels	$V_{CC} = 4.5V$			
Sink Current				
OSC	$V_{OL} = 0.4V$	0.5		mA
$\overline{ON}$	$V_{OL} = 1.5V$	1.5	7.5	mA
X_{OUT}	XSEL = 1, $X_{IN} = 4.5V$, $V_{OL} = 1.0V$	0.25		mA
X_{OUT}	XSEL = 0, $X_{IN} = 4.5V$, $V_{OL} = 2.0V$	8.0		μA
1 Hz, DO	$V_{OL} = 0.8V$	0.8		mA
Source Current				
$\overline{ON}$	$V_{OH} = 1.0V$	60		μA
X_{OUT}	XSEL = 1, $X_{IN} = 0V$, $V_{OH} = 3.0V$	0.27		mA
X_{OUT}	XSEL = 0, $X_{IN} = 0V$, $V_{OH} = 3.0V$	10		μA
1 Hz, DO	$V_{OH} = 2.0V$	0.4		mA

AC Electrical Characteristics

COP398/COP399: $-40°C \leq T_A \leq +85°C$ unless otherwise specified.

COP498/COP499: $0°C \leq T_A \leq +70°C$ unless otherwise specified.

Parameter	Conditions	Min	Max	Units
COP Interface				
SK Frequency	CS = 1, CE = 1 COP498/COP499	4.096	250	kHz
	CS = 1, CE = 1 COP398/COP399	8.192	250	kHz
SK Duty Cycle	SK frequency ≥ 25 kHz	25	75	%
	SK frequency = 4.096 kHz	48	52	%
Inputs				
CS				
t_{CSS}		0.2		μs
t_{CSH}		0		μs
DI				
t_{SETUP}		0.4		μs
t_{HOLD}		0.4		μs
Output				
DO	$C_L = 100$ pF, $4.5V \leq V_{CC} \leq 5.5V$,			
t_{pd1}, t_{pd0}	$V_{OUT} = 1.5V$		2.0	μs
t_{pd1}, t_{pd0}	$C_L = 50$ pF, V_{CC} = Min.,			
	$V_{OUT} = 1.5V$		2.4	μs
Crystal Osc. Frequency	XSEL = 1		2.1	MHz
	XSEL = 0		65	kHz

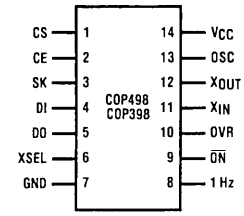


TL/DD/6684-2

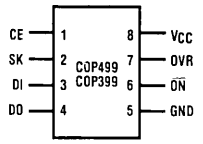
FIGURE 2. Synchronous Data Timing

Connection Diagrams

Dual-In-Line Package



Top View



TL/DD/6684–3

FIGURE 3

Order Number COP398N, COP498N, COP399N, or COP499N
See NS Package Number N08E or N14A

Pin Descriptions

Pin	Description
CS	Chip Select
CE	Chip Enable
SK	Serial Data Clock
DI	Serial Data Input
DO	Serial Data Output
XSEL	Crystal Option Select
X_{IN}	Crystal Oscillator Input
X_{OUT}	Crystal Oscillator Output

Pin	Description
1 Hz	1 Hz Square Wave Output
$\overline{ON}$	Active Low Wake-Up Signal to COPS Controller
OVR	External Override Wake-Up for COPS Controller
OSC	Open Drain Oscillator Output
V_{CC}	Power Supply
GND	Ground

COP398 and COP399 are extended temperature devices (−40°C to +85°C) of COP498 and COP499 (0°C to 70°C) respectively, with all other functional and electrical characteristics being the same. Therefore, no further attempt will be made to distinguish between COP498 and COP398 or between COP499 and COP399. Unless otherwise specified, the following descriptions will apply to both COP498 and COP499, and they will be known as the device.

INSTRUCTION SET

COP498 has six instructions as indicated in *Figure 4.* Note that the MSB of any given instruction is a "1". This bit is properly viewed as a start bit in the interface sequence. The lower 4 bits of the instruction contain the command for the device. One of the instructions (TSEC) should not be used in COP499 as it serves no purpose.

Instruction	Opcode	Comments
	MSB	
WRITE	1 s 1 r_1 r_0	s = $\overline{ON}$ (wake up signal) frequency select 1 = 16 Hz, 0 = 1 Hz (s selection for COP498 only) (s = 0 for COP499)
READ	1 1 0 r_1 r_0	r_1, r_0 = register number (00, 01, 10, 11)
WREN	1 0 0 1 1	Write enable
WRDS	1 0 0 0 0	Write disable
TSEC	1 0 0 1 0	Test timer seconds latch (COP498 only)
SLEEP	1 0 0 0 1	Put COPS controller to sleep ($\overline{ON}$ high)

FIGURE 4. Instruction Set

Functional Description

A block diagram of COP498 and COP499 is given in *Figure 1.* Positive logic is used. When a bit is set to the higher voltage it is a logic "1"; when a bit is reset to the lower voltage it is a logic "0". The COP498 can execute six instructions: READ (from any one of 4 registers in memory); WRITE (to any one of 4 registers in memory); WREN (write enable); WRDS (write disable); TSEC (test and reset timer seconds latch); and SLEEP (drive $\overline{ON}$ signal high to turn off COPS controller). The COP499 can execute all the above instructions except TSEC. All communications with the device are via the serial MICROWIRE interface. Both CS and CE (CE only in COP499) must be high to enable the device. The device must be deselected between instructions — either CS and/or CE must go low to insure proper operation. The deselecting of the device resets the counters and serial input register.

READ/WRITE MEMORY

The device has 256 bits of read/write memory. The memory is organized as 4 registers of 64 bits each. The data is accessed serially through the Data input (DI) and Data Output (DO) pins. SK is the clock signal for data and instructions.

The memory address register can be conceived of as two registers: one two bits long and loaded directly from the instruction; the other six bits long and incremented by 1 with each SK pulse as long as the chip is selected. The two bit register does not change during the execution of a given instruction. The six bit register is reset to zero while the device is deselected. When counting, the six bit register wraps around from its maximum value back to zero. Thus memory locations are addressed relative to the number of SK pulses after the chip is selected.

Functional Description (Continued)

The READ instruction will select one of the 4 registers (the register being identified in the instruction opcode as indicated in *Figure 4*) and output the contents of that register to the DO pin until the device is deselected. Note that data output from the device, as a result of a READ instruction, continues as long as the device is selected and clocks are provided. Reading more than 64 bits will cause rereading of some bits as the memory address register wraps around from the maximum value back to zero.

The WRITE instruction selects one of the 4 registers (the register being identified in the instruction opcode as indicated in *Figure 4*) and takes the data from the DI pin and stores that data into the memory register until the device is deselected. The write Operation continues as long as the device is selected and clocks are provided. Thus writing more than 64 bits will cause a portion of the data to be overwritten.

TIMER (COP498 ONLY)

With the XSEL pin tied high (V_{CC}), the timer is a 21 stage counter which can divide a 2.097152 MHz signal down to 1 Hz. This creates the 1 Hz signal output. With XSEL tied low (ground), the timer is a 15 stage counter which divides a 32.768 kHz signal down to create the 1 Hz signal output. The rising edge of the 1 Hz signal is used internally to set the timer seconds latch. A wake-up signal is generated at the $\overline{ON}$ output. This signal can be used to turn a COPS controller on. The wake-up rate is software selectable and may be either 1 Hz or 16 Hz. A bit in the WRITE instruction controls this wake-up rate (see *Figure 4*). By means of the SLEEP instruction a COPS controller may cause the $\overline{ON}$ signal to go high thereby providing a means for the controller to safely turn itself off.

An override capability is present whereby the $\overline{ON}$ pin may be prevented from going high. A "1" level at the OVR pin will force $\overline{ON}$ to go low (or stay low) thereby causing the controller to turn on or remain on. $\overline{ON}$ will remain low, and the controller on, as long as the OVR pin is high. To preserve timekeeping when using the override feature, a timer seconds latch is provided. This latch is set by the rising edge of the 1 Hz signal and is read and reset by the TSEC instruction. The timer seconds latch is primarily intended for use when the override feature is implemented. However, it does provide a convenient one second timer which is software testable over a common serial port.

SYSTEM CONSIDERATIONS

When the COPS processor is being turned on and off, during the power supply transition between ground and operating voltage, some pulses may occur at the output pins of the processor. By using the WRDS and WREN instructions, together with the higher "1" level of the CE pin, accidental writing into the memory may be prevented. This is done by disabling the write operation before going to sleep and enabling the write operation when the COPS processor starts execution. A WRDS instruction is automatically executed if the SLEEP instruction causes $\overline{ON}$ to go high turning off the COPS processor. Furthermore, WREN instruction is disabled as long as $\overline{ON}$ remains high.

The XSEL pin, which identifies the timer counter length, should be tied to either V_{CC} or ground depending on the crystal input. For proper operation, the state of XSEL should not be changed while the device is in operation. If the oscillator and timer features are not used, the X_{IN} pin should be connected to the GND pin and XSEL tied to V_{CC}. If the override feature is not used the OVR pin should be connected to the GND pin.

The device is in a static mode when either the CS or CE pin is low. However, the device is in a dynamic mode when both CS and CE are high and at least one high level has been detected at SK while both pins are high. Because of this, a minimum frequency is specified for the SK clock. This minimum frequency really translates to maximum on and off times for the SK clock. As the SK clock slows down, the duty cycle must get closer to 50%. For best operation, the user should regard the maximum on and off times for the SK clock as about 122 μs (61 μs for COP398/COP399).

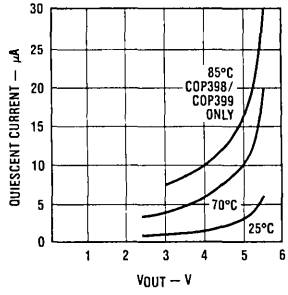
COPS CONTROLLER TO COP498/COP499 HARDWARE INTERFACE

If the COPS controller is operating with a 4 μs instruction cycle time, a 47k resistor should be connected between SK and V_{CC} to speed up the rise time of the SK clock. If the override feature is used in COP498, the override signal should be connected to the OVR pin of the COP498 and an input of the COPS controller. This is simply to provide a means for the controller to know if it was turned on by override or normal timeout. The override signal should be free of noise. In systems where the COPS controller is operating with V_{CC} greater than 6 volts, SI and the override input on the controller should have high impedance, standard TTL level input options selected. To minimize current drain in the controller, the override input to the controller should always use the high impedance option.

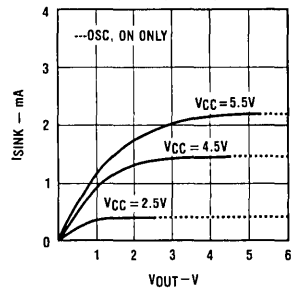
Figure 6a illustrates the COP498 interface in a system with supply voltage less than 6 volts. The COPS controller can either be turned on by the timer or an external signal. A PNP transistor, controlled by the $\overline{ON}$ signal of the COP498, is used to gate the power to the COPS controller. A 0.05 μF capacitor is connected across the supply pins of the controller to reduce voltage variations due to current spikes. It is not recommended to use large capacitance values here as problems can be introduced if the power supply fall time is too long. The switched supply fall time should be kept to about ten instruction cycles of the COPS processor. Resistor R2, between the $\overline{ON}$ pin of the COP498 and the base of the transistor, is used to limit current. Resistor R1, between the base and emitter of the transistor, is used to turn the transistor off when $\overline{ON}$ is high. The CE pin of the COP498 is tied to the V_{CC} pin of the controller. This guarantees that the controller is at its full operating voltage before the COP498 can be accessed. When turned on, the PNP transistor should be saturated in order to minimize the voltage drop across it. The system power supply, which here is V_{CC} to the COP498, must be high enough to insure that the controller V_{CC} — which is the system supply less the voltage drop across the PNP transistor — is high enough to be recognized as a logic "1" at the CE input of the COP498. It is also desirable to have all input signals to the COP498 as close as possible to the COP498 supply levels to eliminate any static power drain which could significantly increase standby and operating current.

Typical Performance Curves

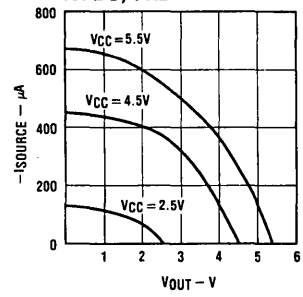
Maximum Quiescent Current COP498/499/398/399



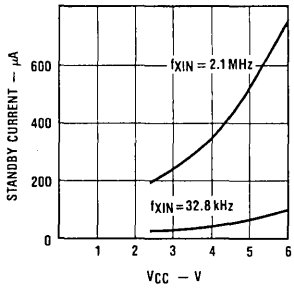
Minimum Sink Current for DO, 1 Hz, OSC, $\overline{ON}$



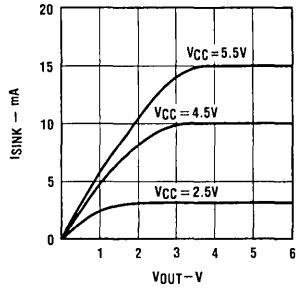
Minimum Source Current for DO, 1 Hz



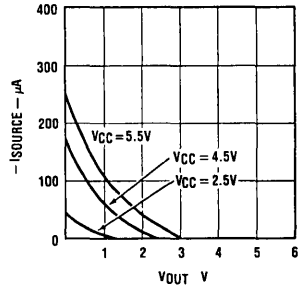
Maximum Standby Current for COP498/398



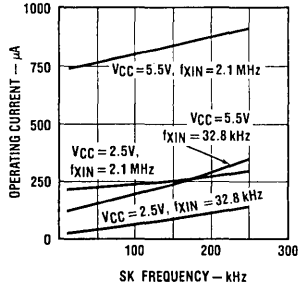
Maximum Sink Current for $\overline{ON}$



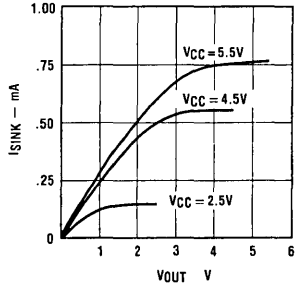
Minimum Source Current for $\overline{ON}$



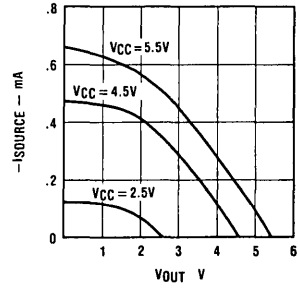
Maximum COP498/398 Operating Current



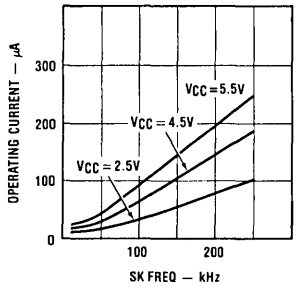
X_{OUT} Minimum Sink Current with XSEL = 1



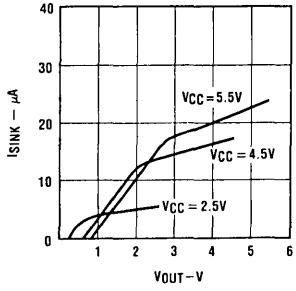
X_{OUT} Minimum Source Current with XSEL = 1



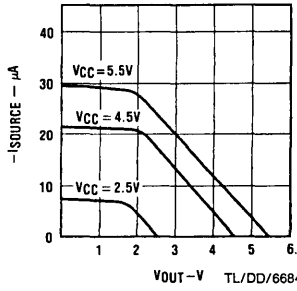
Maximum COP499/399 Operating Current



X_{OUT} Minimum Sink Current with XSEL = 0

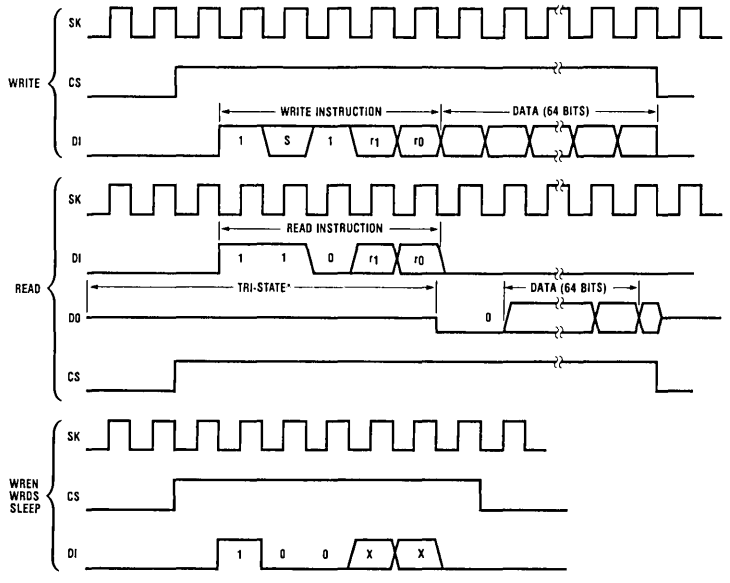


X_{OUT} Minimum Source Current with XSEL = 0



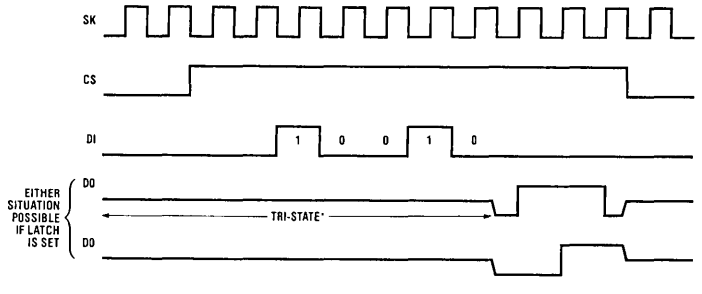
TL/DD/6684-4

Functional Description (Continued)



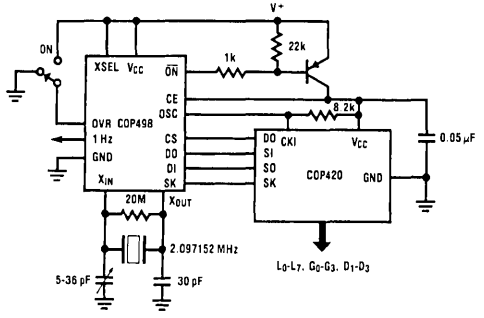
TL/DD/6684-5

FIGURE 5a. Instruction Timing



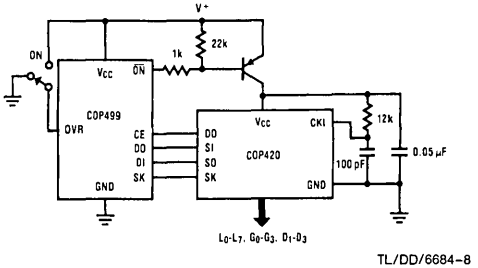
TL/DD/6684-6

FIGURE 5b. TSEC Instruction Timing



TL/DD/6684-7

FIGURE 6a. COP498-COP420 Interface



TL/DD/6684-8

FIGURE 6b. COP499-COP420 Interface

3

Functional Description (Continued)

Figure 6b illustrates the COP499 interface in a system with a supply voltage less than 6 volts. The COPS processor is being turned on by a switch (or an external signal) connected to the OVR pin.

Figure 7 illustrates a COP498 interface in a system with a supply voltage greater than 6 volts. In such a system, the COP498 cannot be connected directly across the system supply. The power to the COP498 is derived from the system supply by means of a standard zener diode arrangement. A zener diode with a breakdown of about 5 volts is recommended. A capacitor is connected across the COP498 supply pins to reduce voltage variations due to current spikes and to supply extra current when the COP498 is in active operation. Here it is assumed that the COP498 is in standby mode, i.e., deselected, most of the time and is active, selected, for a short period (less than 100 SK periods).

The zener diode series resistor R3 should be selected to meet the current requirements of the zener diode and the standby current of the device. The primary purpose of the zener diode is to place an upper limit on the value of V_{CC} to the device. This insures that V_{CC} to the device will not exceed the specified maximum value. Since the device will operate from 2.5V to 6.0V, the choice of zener diode and series resistor is not critical.

Note that the user may generate the two supply voltages in any manner compatible with system requirements.

Because the COPS controller and the device have different operating voltages, the high impedance standard TTL level input should be selected on the COPS controller for SI and any other input to the controller from the device.

SAMPLE SYSTEM CURRENT DRAIN CALCULATION

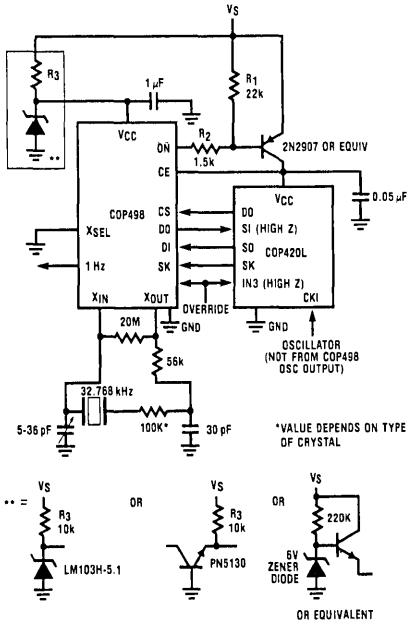
Suppose a 5V system consists of a COP420 and a COP498 with a 32.768 kHz crystal. The COP420 is being turned on once a second. Assume that the COP420 need 10 ms for internal reset and 10 ms to update all the necessary information, then the COP420 will be turned on for 20 ms every second, i.e., a duty cycle of 2%; and the COP498 will be in operating mode for at most 10 ms, i.e., a duty cycle of less than 1%. Because of the short duty cycle, it is further assumed that the COP498 current drain will be that of standby current, about 75 μA at 5V. The current drain through the base of the switching transistor that turns on the COP420 can be estimated by the voltage drop across the current limiting resistor and in this case is assumed to be 3.5 mA.

COP498 current drain = 75 μA
COP420 current drain = 0.02 × 25 mA = 500 μA
Switching transistor base current = 0.02 × 3.5 mA = 70 μA
Total system current drain = 500 + 70 + 75 μA = 645 μA

The result shows that it is possible to achieve the low cost of NMOS and low power dissipation of CMOS simultaneously with a system consisting of a COP498 and a COPS processor.

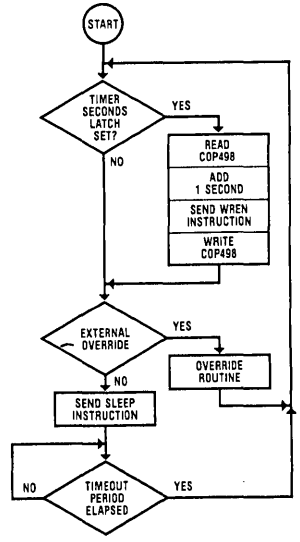
COPS CONTROLLER — COP498/398 SOFTWARE INTERFACE

Figure 8 shows a typical flow chart for a COP498 or COP499 interface to a COPS microcontroller system. This flow chart also illustrates the override feature. Since the override feature is being used, the first step is to inquire the device if it is necessary to increment the time. It is assumed that timekeeping is a necessary part of the application. This interrogation of the device is accomplished by means of the TSEC instruction which dumps the contents of the timer seconds latch to the serial output port and resets the latch. If the latch was set, the time must be incremented. This is accomplished by reading the appropriate memory register into the controller, incrementing the time and writing the register back out to the device. The next step is to check for the override signal. If it is present a special override routine may be performed. If no override is present, the controller



TL/DD/6684-9

FIGURE 7. COP498-COP420L Interface with V_S = 9V and 32.768 kHz Crystal



TL/DD/6684-10

FIGURE 8. Typical COP498 Interface Flowchart

Functional Description (Continued)

turns itself off by sending a SLEEP command to the device. After sending the SLEEP command, the controller goes into a loop to wait for power to go off. In the event the controller is turned back on by the override signal before the voltage has dropped, the loop has a time limit which, when exceeded, causes the controller to jump to the beginning of the program and start again. If the override feature is not used there is no need to test the timer seconds latch nor to test for the override signal. Without the override, the controller can only be turned on by the COP498 if the time out period has elapsed. Note also that the timer features continue to operate regardless of the state of the override signal. The override signal, when high, merely forces the $\overline{ON}$ pin to go low. The operation of the rest of the chip is in no way affected by the override signal.

GENERAL CODE FOR SOFTWARE INTERFACE

The code in *Figure 9a* is recommended for interfacing the device to any COPS controller other than COP410L/COP411L. The code in *Figure 9b* is the recommended interface code for COP410L/COP411L. The code is written as subroutines and the code uses one level of subroutine internally. It is apparent from the code that the software interface is somewhat different for the READ and WRITE instructions than for the rest of the instructions. The routine labelled SETUP is assumed to be in page 2 of the ROM. The rest of the code may be located anywhere in program memory subject to the usual programming rules of COPS microcontrollers. The lower four bits of the instruction opcode are assumed to be located in RAM location COMAND, which is chosen as location 3,15. Data I/O uses register 2. The controller-COP498/499 interface is assumed to be as in *Figure 6* or *Figure 7*. It is assumed that the SIO register in the COPS controller is enabled as serial I/O prior to entry to these routines.

```
WRITE:   JSRP   SETUP
RW:      LD
         XAS              ; READ/WRITE DATA
         XIS
         JP     RW
         OBD              ; DISABLE THE COP498/499 (B=0)
         JP     FINISH
READ:    JSRP   SETUP
         NOP              ; NEED A TOTAL OF 5 SK CLOCK DELAYS (5 NOP'S)
         NOP              ; UNTIL DATA OUT IS VALID AT SIO REGISTER
         NOP
         NOP
         NOP
         JP     RW
INSTRT:  JSRP   SETUP     ; ROUTINE FOR THE REST OF THE INSTRUCTIONS
         NOP
         NOP              ; DELAYS TO INSURE PROPER TIMING
FINISH:  CLRA
         RC
         OBD              ; DESELECT THE COP498/499 (B=0)
         XAS              ; TURN OFF THE CLOCK
         RET
         . PAGE 2
SETUP:   LBI    COMMAND   ; POINT TO LOCATION WHERE COMMAND STORED
         CLRA
         SC
         XAS              ; TURN ON SK CLOCK
         OBD              ; ENABLE THE COP498/499 (B=15)
         CLRA
         XAS              ; MAKE SURE NO INVALID DATA SENT
         CLRA
         AISC   1         ; SET UP START BIT
         SC
         XAS              ; SEND START BIT MSD OF INSTRUCTION
         LD               ; FETCH COMMAND TO A
         NOP
         NOP              ; MAINTAIN PROPER TIMING
         XAS              ; SEND COMMAND
         LBI    2,0       ; POINT TO READ/WRITE REGISTER
         RET              ; RETURN TO MAIN ROUTINE
```

FIGURE 9a. Software Interface to COP498/COP499 for COPS Controllers Other Than COP410L/COP411L

Functional Description (Continued)

```
WRITE:   JSRP   SETUP
RW1:     XAS                ; SEND COMMAND
RW2:     LD
         XDS                ; POSITION Bd PROPERLY
RW:      LD
         XAS
         XIS
         JP     RW
         OBD                ; DISABLE THE COP498/499 (B=0)
         JP     FINISH
READ:    JSRP   SETUP
         XAS                ; SEND READ COMMAND
         NOP                ; DELAY FOR DATA VALID
         NOP
         NOP
         NOP
         NOP
         JP     RW2
INSTRT:  JSRP   SETUP       ; ROUTINE FOR REST OF INSTRUCTIONS
         XAS                ; SEND INSTRUCTION
         NOP
         NOP
         NOP                ; DELAY FOR INSTRUCTION ACCEPT
         NOP
FINISH:  CLRA
         RC
         OBD                ; DESELECT THE COP498/499
         XAS                ; TURN OFF THE CLOCK
         RET
         . PAGE 2
SETUP:   LBI    COMMAND
         CLRA
         SC
         XAS                ; TURN ON SK CLOCK
         OBD                 ENABLE THE COP498/COP499 (B=15)
         CLRA
         XAS                ; MAKE SURE NO INVALID DATA SENT
         CLRA
         AISC   1
         SC
         XAS                ; SEND START BIT-MSD OF INSTRUCTION
         LD                 ; FETCH INSTRUCTION
         LBI    2,9
         RET
```

FIGURE 9b. COP410L/COP411L Software Interface to COP498/COP499

The code in *Figure 9a* will read or write 64 bits at a time. Note that in the COP410L/411L the code in *Figure 9b* will read or write 32 bits at a time. The code of *Figure 10* is recommended if the user wishes to work in blocks of 64 bits with the COP410L/411L. Only the code which is different from that shown in *Figure 9b* is shown in *Figure 10.*

The routine in *Figure 10* will read/write into registers 2 and 1 in the COP410L/411L. *Figure 10* illustrates the preferred method of achieving full utilization of the device memory when the COP410L/411L is the contoller. Remember that all the other routines are as shown in *Figure 9B. Figure 10* illustrates only that code that must be changed to achieve full usage of the device memory when using the COP410L/411L.

GENERAL NOTES

1. For complete safety in all cases it is recommended that the SK clock be turned off after the device has been deselected since the device is dynamic when it is enabled, If the clock is turned off while the device is selected, special care must be given to the SK timing characteristics. In no case should the clock be turned off while the device is selected if the SK period is greater than about 50 μs.

Functional Description (Continued)

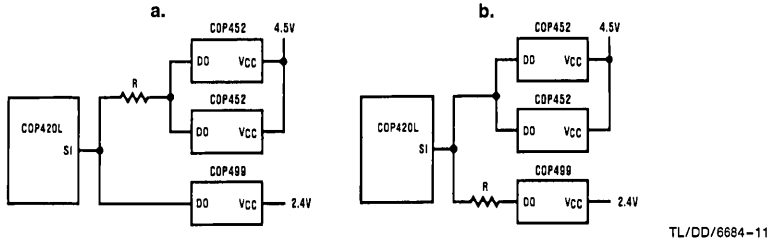
```
WRITE: JSRP   SETUP      ; INITIALIZE, SEE FIGURE 9B
RW1:   XAS               ; SEND COMMAND
RW2:   LD                ; POSITION Bd
       XDS
RW:    LD
       XAS
       X      3          ; USE REGISTERS 2 AND 1
       LD
       NOP
       XAS
       XIS    3
       JP     RW
       OBD               ; DESELECT THE COP498/499
       JP     FINISH
```

FIGURE 10. COP410L/411L-COP498/499 Special Routine

2. The device does not become dynamic until both CS and CE are high and at least one high level is seen at the SK input. Thus the device may be safely enabled prior to turning on the clock as long as SK is low when the device is enabled.
3. The device must be deselected between instructions. Failure to do so will yield improper operation. The device relies on the select lines changing state in order to clear internal registers. Only one of the select lines on the COP498 needs to go low between instructions.
4. The user must insure that a WREN (write enable) instruction has been performed in order to write to the device memory. The WREN command need be given only once unless the SLEEP feature is used. If $\overline{ON}$ goes high as a result of a SLEEP command, a write disable is automatically performed in order to provide maximum protection to the device memory while the COPS controller is powering up and powering down. As long as $\overline{ON}$ remains high, WRITE and WREN instructions are disabled. Thus when the COPS controller wakes up after previously issuing a SLEEP command, a WREN instruction is required before data can be written to the device.
5. The six bit section of the RAM address register will increment whenever there are clock pulses present when the CS and CE pins are high. Thus the user can position the RAM address register if he wishes by selecting the device, holding the DI pin low and supplying the appropriate number of clocks. Then, without deselecting the device, the user would send the instruction and read or write data. Although possible, this technique is not recommended as it is fairly involved.
6. When using the TSEC command in COP498 with the code as given in *Figure 9,* the master program should test for the accumulator greater than 1 to determine if the timer seconds latch was set. Note again, test for greater than 1; do not test for greater than zero.

NOTE ON MICROWIRE INTERFACE

If the device is connected to a MICROWIRE interface containing other circuits whose DO (data output) pins may produce a signal swing higher than V_{CC} of the device, some protection is needed on the DO pin of the device. This happens when the DO pins of several peripherals powered by different voltages are connected together; e.g., a COP452 at 4.5V with a COP499 at 2.4V. When the DO pin of COP498/499 is externally driven above its power supply voltage, a current will flow into it and this current must be limited to 1 mA. As an example we have two COP452s with a COP420L operating at 4.5V and a COP499 operating at 2.4V. When enabled, the DO pin of a COP452 may swing higher than 2.4V, the power supply voltage of the COP499. One way to limit the current is to use a current limiting resistor of 2 kΩ between the DO pins of the COP452 and the COP499. NOTE: the SI pin of the COPS processor MUST BE A Hi-Z INPUT. Two configurations are possible as shown in *Figure 11.* Note that the resistor between DO and SI will give extra RC delay to the signal going from the DO pin to the SI pin of the COPS processor. Connection B is preferred because the DO signal from COP499 has nearly a whole SK cycle to become valid at SI input before the signal is read by the processor. When a ROMless COPS processor (COP401L/COP402/COP404L) is used for emulation, the circuit shown in *Figure 12* may be used to simulate a Hi-Z input for the SI pin.



TL/DD/6684-11

FIGURE 11. High Voltage Protection on DO pin

Functional Description (Continued)

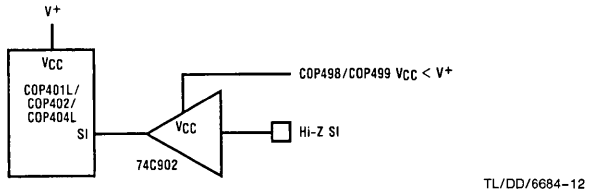


FIGURE 12. Simulating Hi-Z SI Input on ROMless Processors

National Semiconductor Corporation

DS8906 AM/FM Digital Phase-Locked Loop Synthesizer

General Description

The DS8906 is a PLL synthesizer designed specifically for use in AM/FM radios. It contains the reference oscillator, a phase comparator, a charge pump, a 120 MHz ECL/I^2L dual modulus programmable divider, and a 20-bit shift register/latch for serial data entry. The device is designed to operate with a serial data controller generating the necessary division codes for each frequency, and logic state information for radio function inputs/outputs.

The Colpitts reference oscillator for the PLL operates at 4 MHz. A chain of dividers is used to generate a 500 kHz clock signal for the external controller. Additional dividers generate a 12.5 kHz reference signal for FM and a 500 Hz reference signal for AM/SW. One of these reference signals is selected by the data from the controller for use by the phase comparator. Additional dividers are used to generate a 50 Hz timing signal used by the controller for "time-of-day".

Data is transferred between the frequency synthesizer and the controller via a 3 wire bus system. This consists of a data input line, an enable line and a clock line. When the enable line is low, data can be shifted from the controller into the frequency synthesizer. When the enable line is transitioned from low to high, data entry is disabled and data present in the shift register is latched.

From the controller 22-bit data stream, the first 2 bits address the device permitting other devices to share the same bus. Of the remaining 20-bit data word, the next 14-bits are used for the PLL divide code. The remaining 6 bits are connected via latches to output pins. These 6 bits can be used to drive radio functions such as gain, mute, FM, AM, LW and SW only. These outputs are open collector. Bit 18 is used internally to select the AM or FM local oscillator input and to select between the 500 Hz and 12.5 kHz reference. A high level at bit 18 indicates FM and a low level indicates AM.

The PLL consists of a 14-bit programmable I^2L divider, an ECL phase comparator, an ECL dual modulus (p/p + 1) prescaler, and a high speed charge pump. The programmable divider divides by (N+1), N being the number loaded into the shift register (bits 1–14 after address). It is clocked by the AM input via an ECL ÷ 7/8 prescaler, or through a ÷ 63/64 prescaler from the FM input. The AM input will work at frequencies up to 8 MHz, while the FM input works up to 120 MHz. The AM band is tuned with a frequency resolution of 500 Hz and the FM band is tuned with a resolution of 12.5 kHz. The buffered AM and FM inputs are self-biased and can be driven directly by the VCO thru a capacitor. The ECL phase comparator produces very accurate resolution of the phase difference between the input signal and the reference oscillator.

The high speed charge pump consists of a switchable constant current source (−0.3 mA) and a switchable constant current sink (+0.3 mA). If the VCO frequency is low, the charge pump will source current, and sink current if the VCO frequency is high.

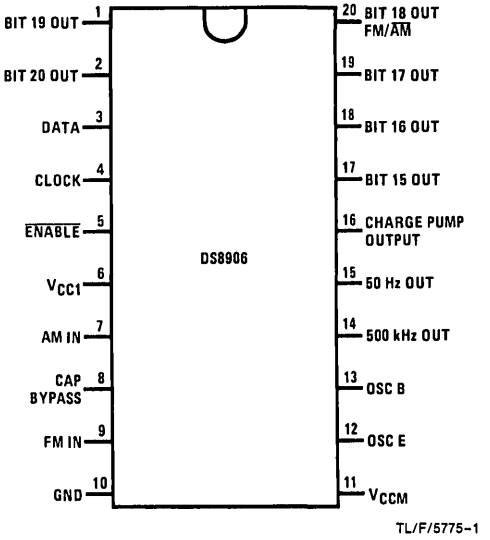
A separate V_{CCM} pin (typically drawing 1.5 mA) powers the oscillator and reference chain to provide controller clocking frequencies when the balance of the PLL is powered down.

Features

- Uses inexpensive 4 MHz reference crystal
- F_{IN} capability greater than 120 MHz allows direct synthesis at FM frequencies
- FM resolution of 12.5 kHz allows usage of 10.7 MHz ceramic filter distribution
- Serial data entry for simplified control
- 50 Hz output for "time-of-day" reference with separate low power supply (V_{CCM})
- 6-open collector buffered outputs for band switching and other radio functions
- Separate AM and FM inputs. AM input has 15 mV (typical) hysteresis

Connection Diagram

Dual-In-Line Package



TL/F/5775–1

Top View

Order Number DS8906N
See NS Package Number N20A

Absolute Maximum Ratings (Note 1)

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage (V_{CC1})	7V
(V_{CCM})	7V
Input Voltage	7V
Output Voltage	7V
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 4 seconds)	260°C

Operating Conditions

	Min	Max	Units
Supply Voltage, V_{CC}			
V_{CC1}	4.75	5.25	V
V_{CCM}	4.5	6.0	V
Temperature, T_A	0	70	°C

DC Electrical Characteristics (Notes 2 and 3)

Symbol	Parameter	Conditions		Min	Typ	Max	Units
V_{IH}	Logical "1" Input Voltage			2.1			V
I_{IH}	Logical "1" Input Current	$V_{IN} = V_{CC1}$			0	10	μA
V_{IL}	Logical "0" Input Voltage					0.7	V
I_{IL}	Logical "0" Input Current	Data, Clock and $\overline{\text{ENABLE}}$ INPUTS, $V_{IN} = 0V$			−5	−25	μA
I_{OH}	Logical "1" Output Current All Bit Outputs, 50 Hz Output	$V_{OH} = 5.25V$				50	μA
	500 kHz Output	$V_{OH} = 2.4V$, $V_{CCM} = 4.5V$				−250	μA
V_{OL}	Logical "0" Output Voltage All Bit Outputs	$I_{OL} = 5$ mA				0.5	V
	50 Hz Output, 500 kHz Output	$I_{OL} = 250$ μA				−0.5	V
I_{CC1}	Supply Current (V_{CC1})	All Bit Outputs High			90	160	mA
$I_{CCM(STANDBY)}$	V_{CCM} Supply Current	$V_{CCM} = 6.0V$, All Other Pins Open			1.5	4.0	mA
I_{OUT}	Charge Pump Output Current	$1.2V \le V_{OUT} \le V_{CCM} - 1.2V$, $V_{CCM} \le 6.0V$	Pump Up	−0.10	−0.30	−0.6	mA
			Pump Down	0.10	0.30	0.6	mA
			TRI-STATE®		0	±100	nA
$I_{CCM(OPERATE)}$	V_{CCM} Supply Current	$V_{CCM} = 6.0V$, $V_{CC1} = 5.25V$, All Other Pins Open			2.5	6.0	mA

AC Electrical Characteristics $V_{CC} = 5V$, $T_A = 25°C$, $t_r \le 10$ ns, $t_f \le 10$ ns

Symbol	Parameter	Conditions		Min	Typ	Max	Units
$V_{IN(MIN)(F)}$	F_{IN} Minimum Signal Input	AM and FM Inputs, $0°C \le T_A \le 70°C$			20	100	mV (rms)
$V_{IN(MAX)(F)}$	F_{IN} Maximum Signal Input	AM and FM Inputs, $0°C \le T_A \le 70°C$		1000	1500		mV (rms)
$F_{OPERATE}$	Operating Frequency Range (Sine Wave Input)	V_{IN} = 100 mV rms, $0°C \le T_A \le 70°C$	AM	0.4		8	MHz
			FM	60		120	MHz
R_{IN} (FM)	AC Input Resistance, FM	120 MHz, V_{IN} = 100 mV rms		300			Ω
R_{IN} (AM)	AC Input Resistance, AM	2 MHz, V_{IN} = 100 mV rms		1000			Ω
C_{IN}	Input Capacitance, FM and AM	V_{IN} = 120 MHz		3	6	10	pF
$t_{\overline{EN}1}$	Minimum $\overline{\text{ENABLE}}$ High Pulse Width				625	1250	ns
$t_{\overline{EN}0}$	Minimum $\overline{\text{ENABLE}}$ Low Pulse Width				375	750	ns
$t_{CLK\overline{EN}0}$	Minimum Time before ENABLE Goes Low that CLOCK must be Low				−50	0	ns
$t_{\overline{EN}0CLK}$	Minimum Time after ENABLE Goes Low that CLOCK must Remain Low				275	550	ns
$t_{CLK\overline{EN}1}$	Minimum Time before $\overline{\text{ENABLE}}$ Goes High that Last Positive CLOCK Edge May Occur				300	600	ns

AC Electrical Characteristics V_{CC} = 5V, T_A = 25°C, $t_r \leq 10$ ns, $t_f \leq 10$ ns (Continued)

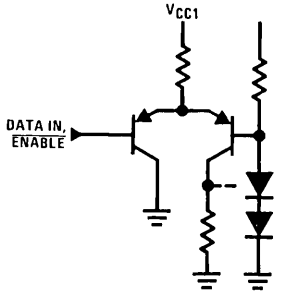
Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{EN1CLK}	Minimum Time After ENABLE Goes High Before an Unused Positive CLOCK Edge May Occur			175	350	ns
t_{CLKH}	Minimum CLOCK High Pulse Width			275	550	ns
t_{CLKL}	Minimum CLOCK Low Pulse Width			400	800	ns
t_{DS}	Minimum DATA Setup Time, Minimum Time Before CLOCK that DATA Must be Valid			150	300	ns
t_{DH}	Minimum DATA Hold Time, Minimum Time After CLOCK that DATA Must Remain Valid			400	800	ns

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range" they are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

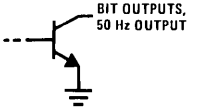
Note 2: Unless otherwise specified min/max limits apply across the 0°C to +70°C temperature range for the DS8906.

Note 3: All currents into device pins shown as positive, out of device pins as negative, all voltages referenced to ground unless otherwise noted. All values shown as max or min on absolute value basis.

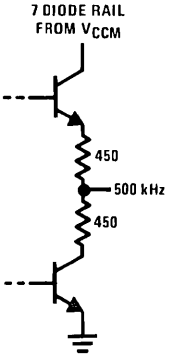
Schematic Diagrams (DS8906 AM/FM PLL Typical Input/Output Schematics)



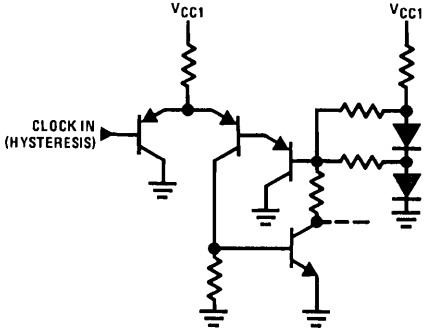
TL/F/5775-2



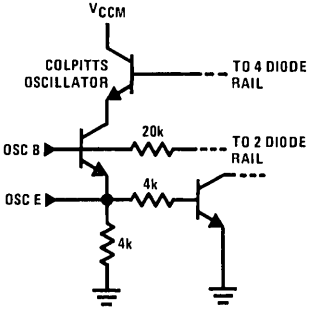
TL/F/5775-3



TL/F/5775-4

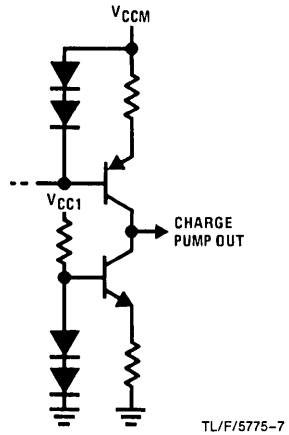


TL/F/5775-5

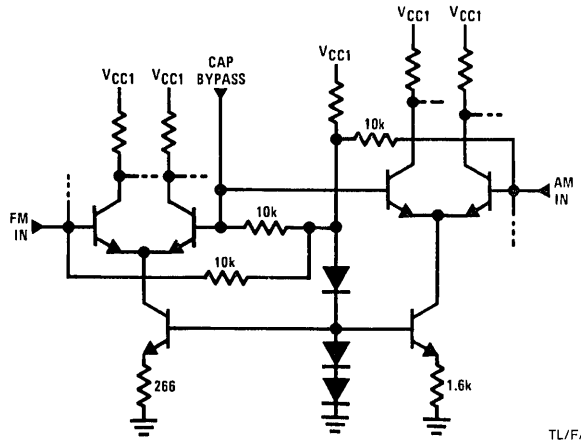


TL/F/5775-6

Schematic Diagrams (DS8906 AM/FM PLL Typical Input/Output Schematics) (Continued)



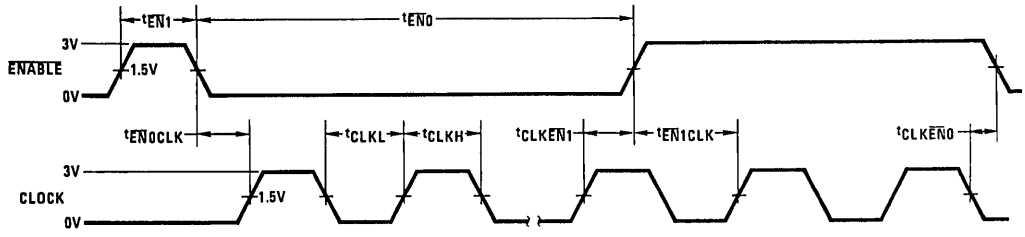
TL/F/5775–7



TL/F/5775–8

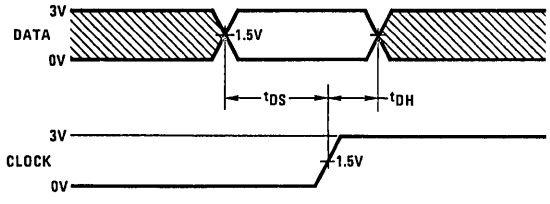
Timing Diagrams*

ENABLE vs CLOCK



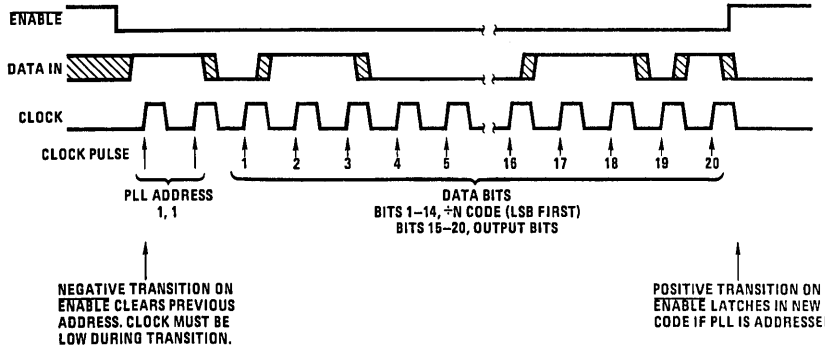
TL/F/5775–9

CLOCK vs DATA



TL/F/5775–10

AM/FM Frequency Synthesizer (Scan Mode)



*Timing diagrams are not drawn to scale. Scale within any one drawing may not be consistent, and intervals are defined positive as drawn.

TL/F/5775–11

Applications Information

SERIAL DATA ENTRY INTO THE DS8906

Serial information entry into the DS8906 is enabled by a low level on the $\overline{\text{ENABLE}}$ input. One binary bit is then accepted from the DATA input with each positive transition of the CLOCK input. The CLOCK input must be low for the specified time preceding and following the negative transition of the $\overline{\text{ENABLE}}$ input.

The first 2 bits accepted following the negative transition of the $\overline{\text{ENABLE}}$ input are interpreted as address. If these address bits are *not* 1,1, *no* further information will be accepted from the DATA inputs, and the internal data latches *will not* be changed when $\overline{\text{ENABLE}}$ returns high.

If these first 2 bits *are* 1,1, then all succeeding bits are *accepted* as data, and are shifted successively into the internal shift register as long as $\overline{\text{ENABLE}}$ remains low.

Any *data* bits preceding the 20th to last bit will be shifted out, and are thus irrelevant. Data bits are counted as any bits following 2 valid (1,1) address bits with the $\overline{\text{ENABLE}}$ low.

When the $\overline{\text{ENABLE}}$ input returns high, any further serial data input is inhibited. Upon this positive transition of the $\overline{\text{ENABLE}}$, the data in the internal shift register is transferred into the internal data latches.

Note that until this time, the states of the internal data latches have remained unchanged.

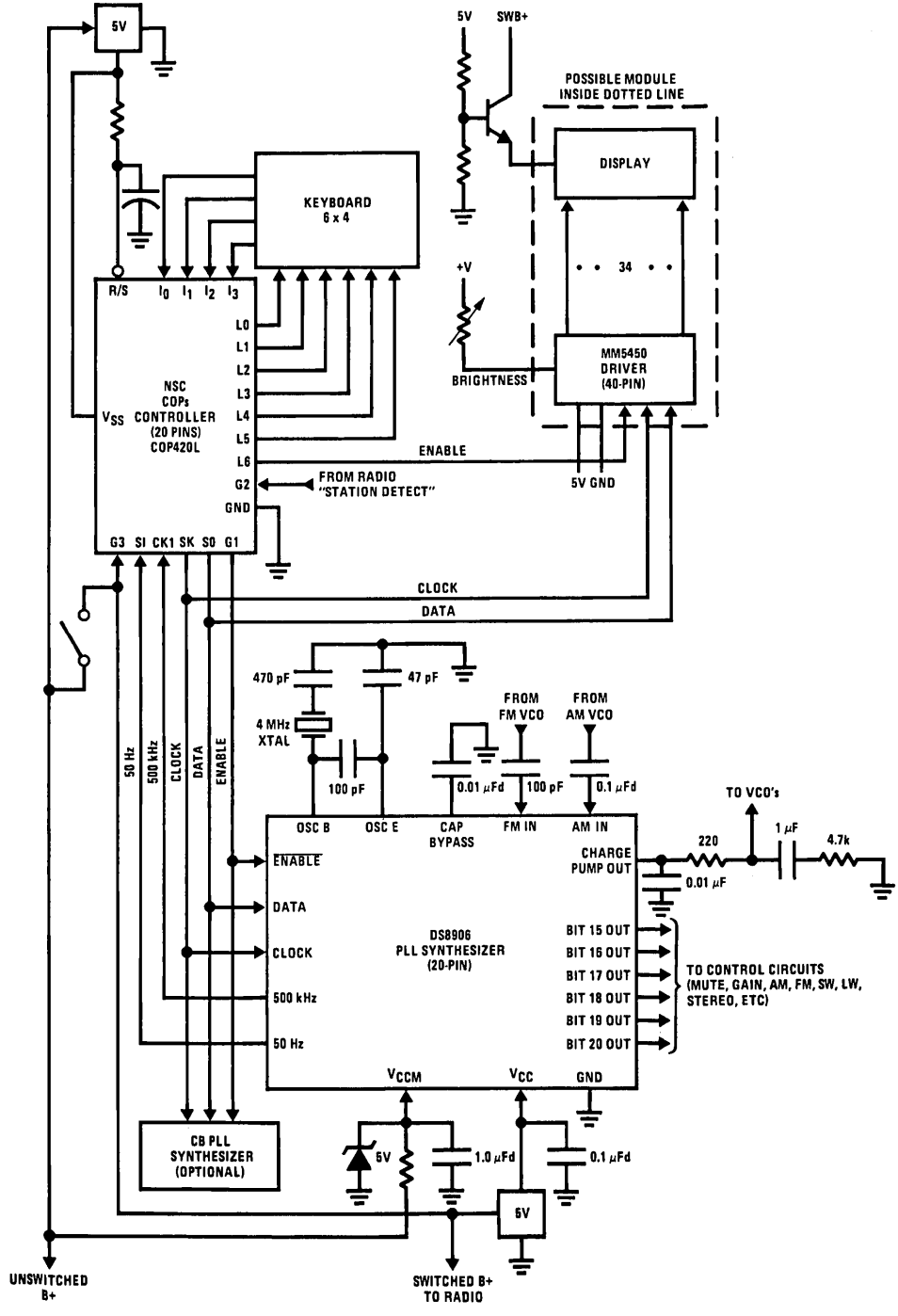
These data bits are interpreted as follows:

DATA BIT POSITION	DATA INTERPRETATION	
Last	Bit 20 Output (Pin 2)	
2nd to Last	Bit 19 Output (Pin 1)	
3rd to Last	Bit 18 Output (FM/$\overline{\text{AM}}$) (Pin 20)	
4th to Last	Bit 17 Output (Pin 19)	
5th to Last	Bit 16 Output (Pin 18)	
6th to Last	Bit 15 Output (Pin 17)	
7th to Last	MSB of N	(2^{13}) ÷ N
8th to Last		(2^{12})
9th to Last		(2^{11})
10th to Last		(2^{10})
11th to Last		(2^{9})
12th to Last		(2^{8})
13th to Last		(2^{7})
14th to Last		(2^{6})
15th to Last		(2^{5})
16th to Last		(2^{4})
17th to Last		(2^{3})
18th to Last		(2^{2})
19th to Last		(2^{1})
20th to Last	LSB of N	(2^{0})

Note. The actual divide code is N + 1, i.e., the number loaded plus 1.

Typical Application

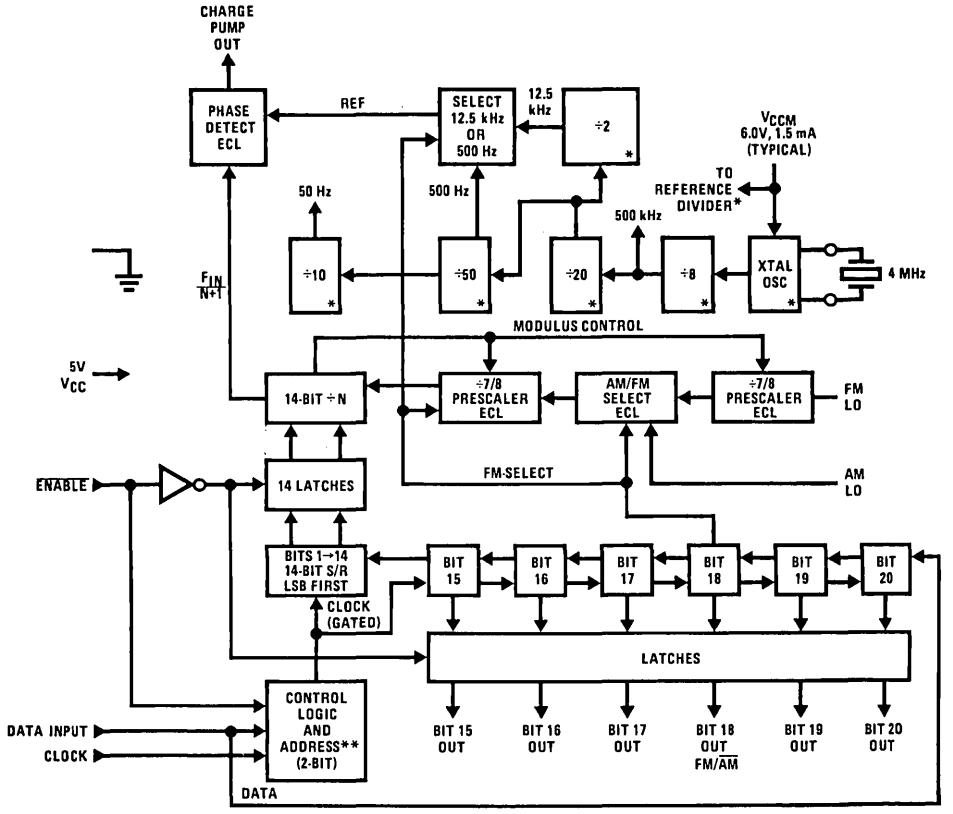
Electronically Tuned Radio Controller System; Direct Drive LED



TL/F/5775–12

Logic Diagram

AM/FM PLL Synthesizer



TL/F/5775–13

*Sections operating from V_{CCM} supply

**Address (1, 1)

National Semiconductor Corporation

DS8907 AM/FM Digital Phase-Locked Loop Frequency Synthesizer

General Description

The DS8907 is a PLL synthesizer designed specifically for use in AM/FM radios. It contains the reference oscillator, a phase comparator, a charge pump, a 120 MHz ECL/I²L dual modulus programmable divider, and an 18-bit shift register/latch for serial data entry. The device is designed to operate with a serial data controller generating the necessary division codes for each frequency, and logic state information for radio function inputs/outputs.

The Colpitts reference oscillator for the PLL operates at 4 MHz. A chain of dividers is used to generate a 500 kHz clock signal for the external controller. Additional dividers generate a 25 kHz reference signal for FM and a 10 kHz reference signal for AM. One of these reference signals is selected by the data from the controller for use by the phase comparator.

Data is transferred between the frequency synthesizer and the controller via a 3 wire bus system. This consists of a data input line, an enable line, and a clock line. When the enable line is low, data can be shifted from the controller into the frequency synthesizer. When the enable line is transitioned from low to high, data entry is disabled and data present in the shift register is latched.

From the controller 20-bit data stream, the first 2 bits address the device permitting other devices to share the same bus. Of the remaining 18-bit data word, the next 13 bits are used for the PLL divide code. The remaining 5 bits are connected via latches to output pins. These 5 bits can be used to drive radio functions such as gain, mute, FM, AM and stereo only. These outputs are open collector. Bit 16 is used internally to select the AM or FM local oscillator input and to select between the 10 kHz and 25 kHz reference. A high level at bit 16 indicates FM and a low level indicates AM.

The PLL consists of a 13-bit programmable I²L divider, an ECL phase comparator, an ECL dual modulus (p/p+1) prescaler, and a high speed charge pump. The programmable divider divides by (N+1), N being the number loaded into the shift register (bits 1–13 after address). It is clocked by the AM input via an ECL ÷7/8 prescaler, or through a ÷63/64 prescaler from the FM input. The AM input will work at frequencies up to 15 MHz, while the FM input works up to 120 MHz. The AM band is tuned with a frequency resolution of 10 kHz and the FM band is tuned with a resolution of 25 kHz. The buffered AM and FM inputs are self biased and can be driven directly by the VCO through a capacitor. The ECL phase comparator produces very accurate resolution of the phase difference between the input signal and the reference oscillator. The high speed charge pump consists of a switchable constant current source (−0.3 mA) and a switchable constant current sink (+0.3 mA). If the VCO frequency is low, the charge pump will source current, and sink current if the VCO frequency is high. When using an AFC the charge pump output may be forced into TRI-STATE® by applying a low level to the charge pump enable input.

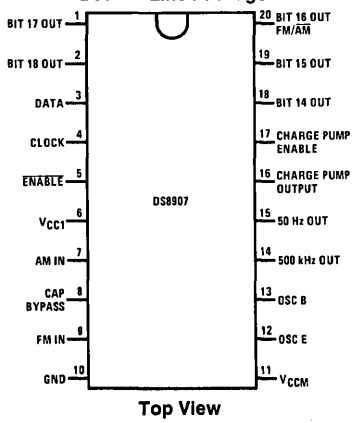
A separate V_{CCM} pin (typically drawing 1.5 mA) powers the oscillator and reference chain to provide controller clocking frequencies when the balance of the PLL is powered down.

Features

- Uses inexpensive 4 MHz reference crystal
- F_{IN} capability greater than 120 MHz allows direct synthesis at FM frequencies
- FM resolution of 25 kHz allows usage of 10.7 MHz ceramic filter distribution
- Serial data entry for simplified control
- 50 Hz output for "time-of-day" reference driven from separate low power V_{CCM}
- 5-open collector buffered outputs for controlling various radio functions
- Separate AM and FM inputs. AM input has 15 mV (typical) hysteresis

Connection Diagram

Dual-In-Line Package



Top View

TL/F/7511-1

Order Number DS8907N
See NS Package Number N20A

Absolute Maximum Ratings (Note 1)

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage	
(V_{CC1})	7V
(V_{CCM})	7V
Input Voltage	7V
Output Voltage	7V
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 4 sec.)	260°C

Operating Conditions

	Min	Max	Units
Supply Voltage, V_{CC}			
V_{CC1}	4.75	5.25	V
V_{CCM}	4.5	6.0	V
Temperature, T_A	0	70	°C

DC Electrical Characteristics (Notes 2 and 3)

Symbol	Parameter	Conditions		Min	Typ	Max	Units
V_{IH}	Logical "1" Input Voltage			2.1			V
I_{IH}	Logical "1" Input Current	V_{IN} = 2.7V			0	10	μA
V_{IL}	Logical "0" Input Voltage					0.7	V
I_{IL}	Logical "0" Input Current	Data, Clock, and $\overline{ENABLE}$ Inputs, V_{IN} = 0V			−5	−25	μA
I_{IL}	Logical "0" Input Current	Charge Pump Enable, V_{IN} = 0V			−250	−450	μA
I_{OH}	Logical "1" Output Current All Bit Outputs, 50 Hz Output	V_{OH} = 5.25V				50	μA
	500 kHz Output	V_{OH} = 2.4V, V_{CCM} = 4.5V				−250	μA
V_{OL}	Logical "0" Output Voltage All Bit Outputs	I_{OL} = 5 mA				0.5	V
	50 Hz Output, 500 Hz Output	I_{OL} = 250 μA				0.5	V
I_{CC1}	Supply Current (V_{CC1})	All Bits Outputs High			90	160	mA
$I_{CCM(STANDBY)}$	V_{CCM} Supply Current	V_{CCM} = 6.0V, All Other Pins Open			1.5	4.0	mA
I_{OUT}	Charge Pump Ougtput Current	1.2V ≤ V_{OUT} ≤ V_{CCM} −1.2V	Pump Up	−0.10	−0.30	−0.6	mA
		V_{CCM} ≤ 6.0V	Pump Down	0.10	0.30	0.6	mA
			TRI-STATE		0	±100	nA
$I_{CCM(OPERATE)}$	V_{CCM} Supply Current	V_{CCM} = 6.0V, V_{CC1} = 5.25V, All Other Pins Open			2.5	6.0	mA

AC Electrical Characteristics V_{CC} = 5V, T_A = 25°C, t_r ≤ 10 ns, t_f ≤ 10 ns

Symbol	Parameter	Conditions		Min	Typ	Max	Units
$V_{IN(MIN)(F)}$	F_{IN} Minimum Signal Input	AM and FM Inputs, 0°C ≤ T_A ≤ 70°C			20	100	mV (rms)
$V_{IN(MAX)(F)}$	F_{IN} Maximum Signal Input	AM and FM Inputs, 0°C ≤ T_A ≤ 70°C		1000	1500		mV (rms)
$F_{OPERATE}$	Operating Frequency Range (Sine Wave Input)	V_{IN} = 100 mV rms 0°C ≤ T_A ≤ 70°C	AM	0.4		8	MHz
			FM	60		120	MHz
$R_{IN}(FM)$	AC Input Resistance, FM	120 MHz, V_{IN} = 100 mV rms		300			Ω
$R_{IN}(AM)$	AC Input Resistance, AM	2 MHz, V_{IN} = 100 mV rms		1000			Ω
C_{IN}	Input Capacitance, FM and AM	V_{IN} = 120 MHz		3	6	10	pF
$t_{\overline{EN}1}$	Minimum $\overline{ENABLE}$ High Pulse Width				625	1250	ns
$t_{\overline{EN}0}$	Minimum $\overline{ENABLE}$ Low Pulse Width				375	750	ns
$t_{CLK\overline{EN}0}$	Minimum Time Before ENABLE Goes Low That CLOCK Must Be Low				−50	0	ns
$t_{\overline{EN}0CLK}$	Minimum Time After ENABLE Goes Low That CLOCK Must Remain Low				275	550	ns
$t_{CLK\overline{EN}1}$	Minimum Time Before $\overline{ENABLE}$ Goes High That Last Positive CLOCK Edge May Occur				300	600	ns

AC Electrical Characteristics V_{CC} = 5V, T_A = 25°C, $t_r \leq 10$ ns, $t_f \leq 10$ ns (Continued)

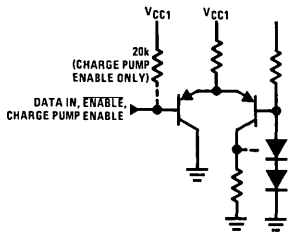
Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{EN1CLK}	Minimum Time After ENABLE Goes High Before an Unused Positive CLOCK Edge May Occur			175	350	ns
t_{CLKH}	Minimum CLOCK High Pulse Width			275	550	ns
t_{CLKL}	Minimum CLOCK Low Pulse Width			400	800	ns
t_{DS}	Minimum DATA Setup Time, Minimum Time before CLOCK That DATA Must Be Valid			150	300	ns
t_{DH}	Minimum DATA Hold Time, Minimum Time after CLOCK That DATA Must Remain Valid			400	800	ns

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range" they are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

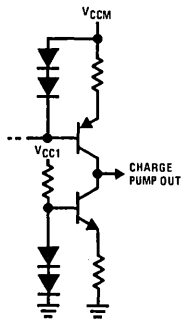
Note 2: Unless otherwise specified min/max limits apply across the −40°C to +85°C temperature range for the DS8907.

Note 3: All currents into device pins shown as positive, out of device pins as negative, all voltages referenced to ground unless otherwise noted. All values shown as max or min on absolute value basis.

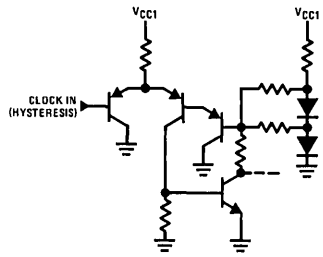
Schematic Diagrams (DS8907 AM/FM PLL typical Input/Output Schematics)



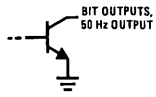
TL/F/7511–2



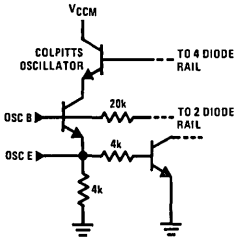
TL/F/7511–3



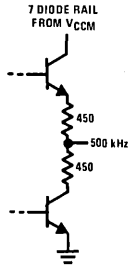
TL/F/7511–4



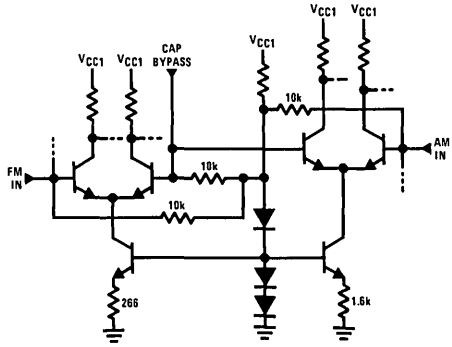
TL/F/7511–5



TL/F/7511–6



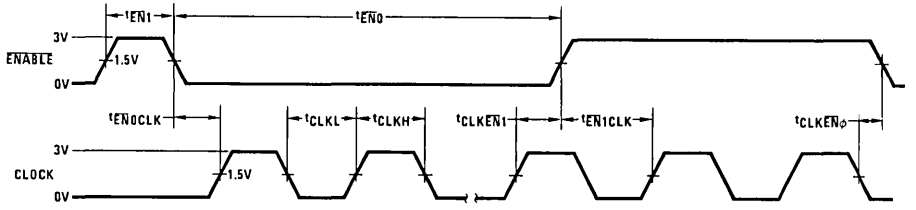
TL/F/7511–7



TL/F/7511–8

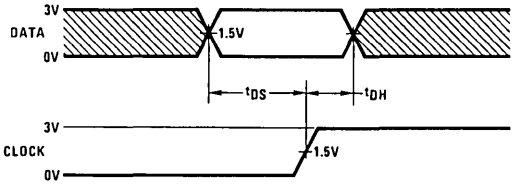
Timing Diagrams*

ENABLE vs CLOCK



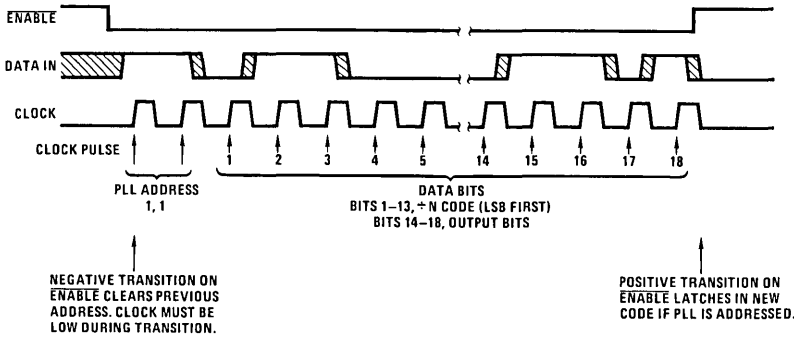
TL/F/7511–9

CLOCK vs DATA



TL/F/7511–10

AM/FM Frequency Synthesizer (Scan Mode)



TL/F/7511–11

*Timing diagrams are not drawn to scale. Scale within any one drawing may not be consistent, and intervals are defined positive as drawn.

SERIAL DATA ENTRY INTO THE DS8907

Serial information entry into the DS8907 is enabled by a low level on the $\overline{\text{ENABLE}}$ input. One binary bit is then accepted from the DATA input with each positive transition of the CLOCK input. The CLOCK input must be low for the specified time preceding and following the negative transition of the $\overline{\text{ENABLE}}$ input.

The first two bits accepted following the negative transition of the $\overline{\text{ENABLE}}$ input are interpreted as address. If these address bits are *not* 1,1 *no* further information will be accepted from the DATA inputs, and the internal data latches *will not* be changed when $\overline{\text{ENABLE}}$ returns high.

If these first two bits *are* 1,1, then all succeeding bits are *accepted* as data, and are shifted successively into the internal shift register as long as $\overline{\text{ENABLE}}$ remains low.

Any data bits preceding the 18th to last bit will be shifted out, and thus are irrelevant. Data bits are counted as any bits *following* two valid address bits (1,1) with the $\overline{\text{ENABLE}}$ low. When the $\overline{\text{ENABLE}}$ input returns high, any further serial data entry is inhibited. Upon this positive transition, the data in the internal shift register is transferred into the internal data latches. Note that until this time, the states of the internal data latches have remained unchanged.

These data bits are interpreted as follows:

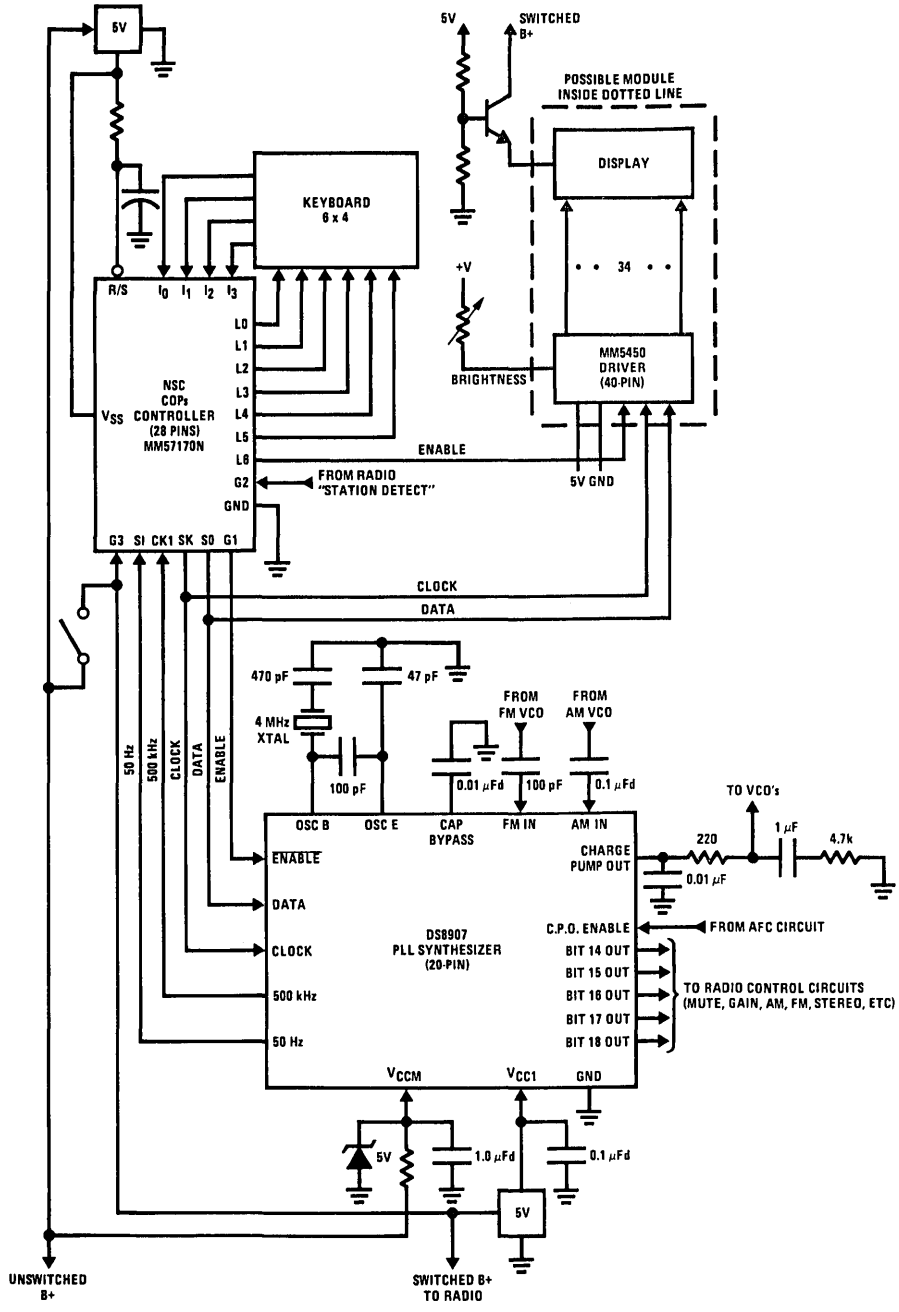
Data Bit Position	Data Interpretation
Last	Bit 18 Output (Pin 2)
2nd to Last	Bit 17 Output (Pin 1)
3rd to Last	Bit 16 Output (FM/$\overline{\text{AM}}$) (Pin 20)
4th to Last	Bit 15 Output (Pin 19)
5th to Last	Bit 14 Output (Pin 18)
6th to Last	MSB of ÷N (2^{12})
7th to Last	(2^{11})
8th to Last	(2^{10})
9th to Last	(2^{9})
10th to Last	(2^{8})
11th to Last	(2^{7})
12th to Last	(2^{6})
13th to Last	(2^{5})
14th to Last	(2^{4})
15th to Last	(2^{3})
16th to Last	(2^{2})
17th to Last	(2^{1})
18th to Last	LSB of ÷N (2^{0})

÷N

Note: The actual divide code is N+1, i.e., the number loaded plus 1.

Typical Application

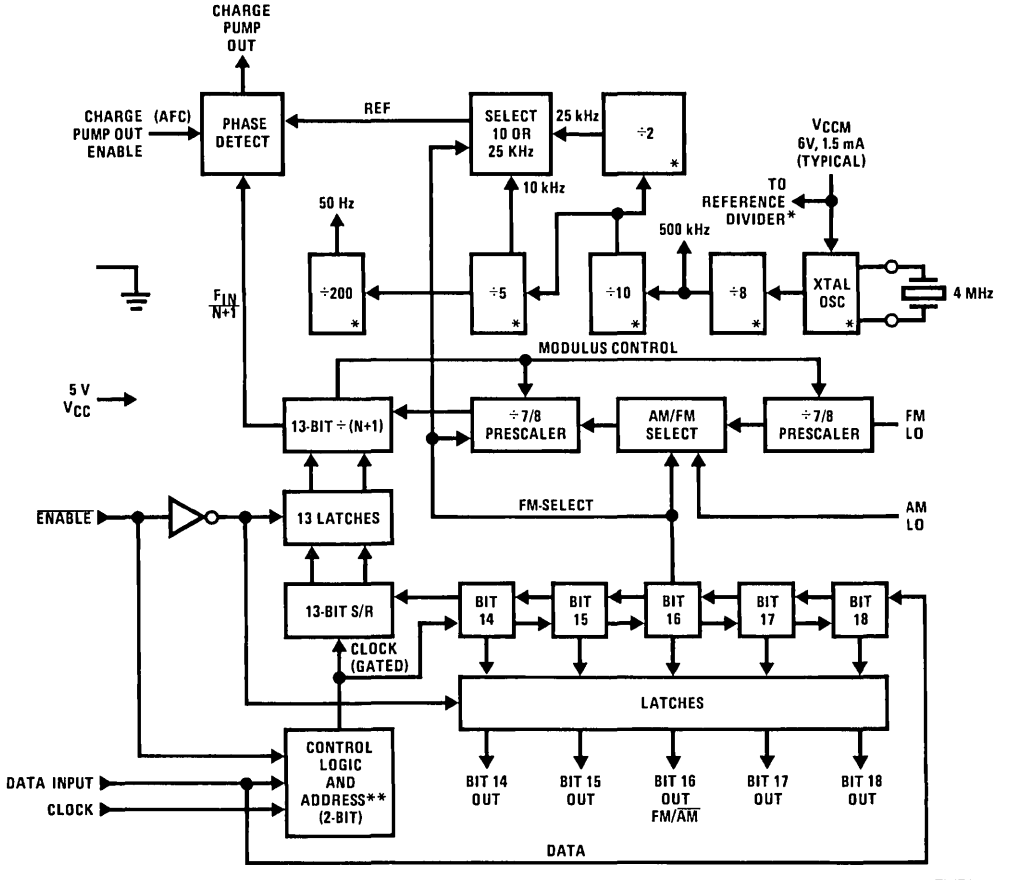
Electronically Tuned Radio Controller System; Direct Drive LED



TL/F/7511–12

Logic Diagram

AM/FM PLL/Synthesizer (Serial Data 20-Pin Package)



TL/F/7511–13

*Sections operating from V_{CCM} supply.

**Address (1, 1)

National Semiconductor Corporation

DS8908 AM/FM Digital Phase-Locked Loop Frequency Synthesizer

General Description

The DS8908 is a PLL synthesizer designed specifically for use in AM/FM radios. It contains the reference oscillator, a phase comparator, a charge pump, an operational amplifier, a 120 MHz ECL/I^2L dual modulus programmable divider, and a 19-bit shift register/latch for serial data entry. The device is designed to operate with a serial data controller generating the necesary division codes for each frequency, and logic state information for radio function inputs/outputs.

A 3.96 MHz pierce oscillator and divider chain generate a 1.98 MHz external controller clock, a 20 kHz, 10 kHz, 9 kHz, and a 1 kHz reference signals, and a 50 Hz time-of-day signal. The oscillator and divider chain are sourced by the V_{CCM} pin thus providing a low power controller clock drive and time-of-day indication when the balance of the PLL is powered down.

The 21-bit serial data steram is transferred between the frequency synthesizer and the controller via a 3-wire bus system comprised of a data line, a clock line, and an enable line.

The first 2 bits in the serial data stream address the synthesizer thus permitting other devices such as display drivers to share the same bus. The next 14 bits are used for the PLL(N+1) divide code. The 15th bit is used internally to select the AM or FM local oscillator input. A high level on this bit enables the FM input and a low level enables the AM input. The 16th and 17th bits are used to select one of the 4 reference frequencies. The 18th and 19th bits are connected via latches to open collector outputs. These outputs can be used to drive radio functions such as gain, mute, AM, FM, or charge pump current source levels.

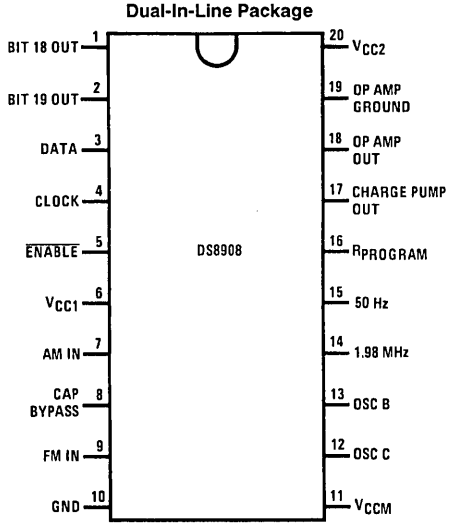
The PLL consists of a 14-bit programmable I^2L divider, an ECL phase comparator, an ECL dual modulus (p/p +1) prescaler, a high speed charge pump, and an operational amplifier. The programmable divider divides by (N+1), N being the number loaded into the shift register. The programmable divider is clocked through a $\div 7/8$ prescaler by the AM input or through a $\div 63/64$ prescaler by the FM input. The AM input will work at frequencies up to 15 MHz, while the FM input works up to 120 MHz. The VCO can be tuned with a frequency resolution of either 1 kHz, 9 kHz, 10 kHz, or 20 kHz. The buffered AM and FM inputs are self-biased and can be driven directly by the VCO through a capacitor. The ECL phase comparator produces very accurate resolution of the phase difference between the input signal and the reference oscillator. The high speed charge pump consists of a switchable constant current source and sink. The charge pump can be programmed to deliver from 75 μA to 750 μA of constant current by connection of an external resistor from pin $R_{PROGRAM}$ to ground or the open collector bit outputs. Connection of programming resistors to the bit outputs enables the controller to adjust the loop gain for the particular reference frequency selected. The charge pump will source current if the VCO frequency is high and sink current if the VCO frequency is low. The low noise operational amplifier provided has a high impedance JFET input and a large output voltage range. The op amp's negative input is common with the charge pump output and its positive input is internally biased.

Features

- Uses inexpensive 3.96 MHz reference crystal
- F_{IN} capability greater than 120 MHz allows direct synthesis at FM frequencies
- FM resolution of either 10 kHz or 20 kHz allows usage of 10.7 MHz ceramic filter distribution
- Serial data entry for simplified control
- 50 Hz output for time-of-day reference driven from separate low power V_{CCM}
- 2 open collector buffered outputs for controlling various radio functions or loop gain
- Separate AM and FM inputs; AM input has 15 mV (typical) hysteresis
- Programmable charge pump current sources enable adjustment of system loop gain
- Operational amplifier provides high impedance load to charge pump output and a wide voltage range for the VCO input

Connection Diagram

Dual-In-Line Package



TL/F/5111–1

Top View
Order Number DS8908N
See NS Package Number N20A

Absolute Maximum Ratings (Note 1)

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage	
(V_{CC1}) (V_{CCM})	7V
(V_{CC2})	17V
Input Voltage	7V
Output Voltage	7V
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 4 seconds)	260°C

Operating Conditions

	Min	Max	Units
V_{CC1}	4.5	5.5	V
V_{CC2}	V_{CC1} + 1.5	15.0	V
V_{CCM}	3.5	5.5	V
Temperature, T_A	−40	+85	°C

DC Electrical Characteristics (Notes 2 and 3)

Symbol	Parameter	Conditions		Min	Typ	Max	Units
V_{IH}	Logical "1" Input Voltage			2.0			V
I_{IH}	Logical "1" Input Current	V_{IN} = 2.7V			0	10	μA
V_{IL}	Logical "0" Input Voltage					0.8	V
I_{IL}	Logical "0" Input Current	Data, Clock, and $\overline{ENABLE}$ Inputs, V_{IN} = 0V			−5	−25	μA
I_{OH}	Logical "1" Output Current All Bit Outputs, 50 Hz Output	V_{OH} = 5.5V				50	μA
	1.98 MHz Output	V_{OH} = 2.4V, V_{CCM} = 4.5V				−250	μA
V_{OL}	Logical "0" Output Voltage All Bit Outputs	I_{OL} = 5 mA				0.5	V
	50 Hz Output, 1.98 MHz Output	I_{OL} = 250 μA				0.5	V
	1.98 MHz Output	I_{OL} = 20 μA, T_A > 70°C I_{OL} = 20 μA, T_A ≤ 70°C				0.3 0.4	V V
I_{CC1}	Supply Current (V_{CC1})	All Bit Outputs High				160	mA
I_{CCM}	V_{CCM} Supply Current	V_{CCM} = 5.5V, All Other Pins Open			2.5	4.0	mA
I_{OUT}	Charge Pump Ougtput Current	3.33k ≤ R_{PROG} ≤ 33.3k I_{OUT} Measured between Pin 17 and Pin 18 I_{PROG} = V_{CC1}/2 R_{PROG}	Pump Up	−20	I_{PROG}	+20	%
			Pump Down	−20	I_{PROG}	+20	%
			TRI-STATE®		0	11	nA
I_{CC2}	V_{CC2} Supply Current	V_{CCM} = 5V, V_{CC1} = 5.5V, V_{CC2} = 15V All Other Pins Open			6.7	11	mA
OP_{VOH}	Op Amp Minimum High Level	V_{CC1} = 4.5V, I_{OH} = −750 μA		V_{CC2} −0.4			V
OP_{VOL}	Op Amp Maximum Low Level	V_{CC1} = 5.5V, I_{OL} = 750 μA				0.6	V
CPO_{BIAS}	Charge Pump Bias Voltage Delta	CPO Shorted to Op Amp Output CPO = TRI-STATE Op Amp I_{OL}: 750 μA vs −750 μA				100	mV

AC Electrical Characteristics V_{CC} = 5V, T_A = 25°C, t_r ≤ 10 ns, t_f ≤ 10 ns

Symbol	Parameter	Conditions		Min	Typ	Max	Units
$V_{IN(MIN)(F)}$	F_{IN} Minimum Signal Input	AM and FM Inputs, −40°C ≤ T_A ≤ 85°C			20	100	mV(rms)
$V_{IN(MAX)(F)}$	F_{IN} Maximum Signal Input	AM and FM Inputs, −40°C ≤ T_A ≤ 85°C		1000	1500		mV(rms)
$F_{OPERATE}$	Operating Frequency Range (Sine Wave Input)	V_{IN} = 100 mV rms −40°C ≤ T_A ≤ 85°C	AM	0.5		15	MHz
			FM	80		120	MHz
R_{IN}(FM)	AC Input Resistance, FM	120 MHz, V_{IN} = 100 mV rms		600			Ω
R_{IN}(AM)	AC Input Resistance, AM	15 MHz, V_{IN} = 100 mV rms		1000			Ω
C_{IN}	Input Capacitance, FM and AM	V_{IN} = 120 MHz (FM), 15 MHz (AM)		3	6	10	pF
t_{EN1}	Minimum ENABLE High Pulse Width				625	1250	ns

AC Electrical Characteristics V_{CC} = 5V, T_A = 25°C, $t_r \leq 10$ ns, $t_f \leq 10$ ns (Continued)

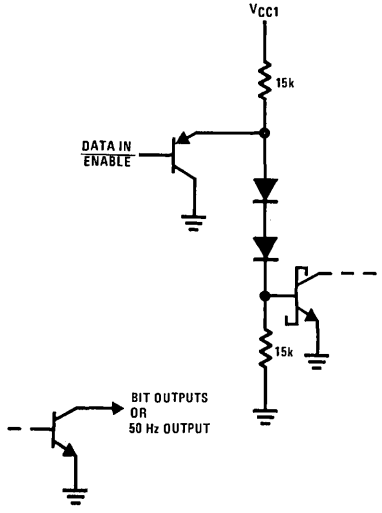
Symbol	Parameter	Conditions	Min	Typ	Max	Units
$t_{\overline{EN}0}$	Minimum $\overline{\text{ENABLE}}$ Low Pulse Width			375	750	ns
$t_{CLK\overline{EN}0}$	Minimum Time before $\overline{\text{ENABLE}}$ Goes Low That CLOCK Must Be Low			−50	0	ns
$t_{\overline{EN}0CLK}$	Minimum Time after $\overline{\text{ENABLE}}$ Goes Low That CLOCK Must Remain Low			275	550	ns
$t_{CLK\overline{EN}1}$	Minimum Time before $\overline{\text{ENABLE}}$ Goes High That Last Positive CLOCK Edge May Occur			300	600	ns
$t_{\overline{EN}1CLK}$	Minimum Time after $\overline{\text{ENABLE}}$ Goes High before an Unused Positive CLOCK Edge May Occur			175	350	ns
t_{CLKH}	Minimum CLOCK High Pulse Width			275	550	ns
t_{CLKL}	Minimum CLOCK Low Pulse Width			400	800	ns
t_{DS}	Minimum DATA Set-Up Time, Minimum Time before CLOCK That DATA Must Be Valid			150	300	ns
t_{DH}	Minimum DATA Hold Time, Minimum Time after CLOCK That DATA Must Remain Valid			400	800	ns

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range" they are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

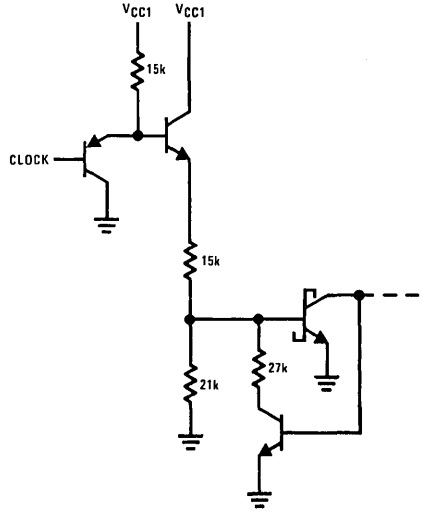
Note 2: Unless otherwise specified min/max limits apply across the −40°C to +85°C temperature range for the DS8908.

Note 3: All currents into device pins shown as positive, out of device pins as negative, all voltage referenced to ground unless otherwise noted. All values shown as max or min on absolute value basis.

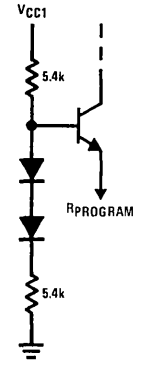
Schematic Diagrams (DS8908 AM/FM PLL Typical Input/Output Schematics)



TL/F/5111–2

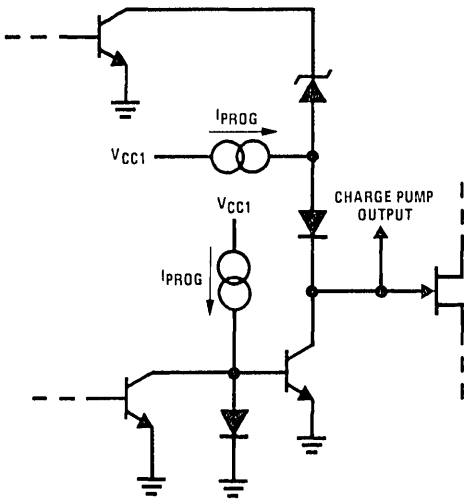


TL/F/5111–3

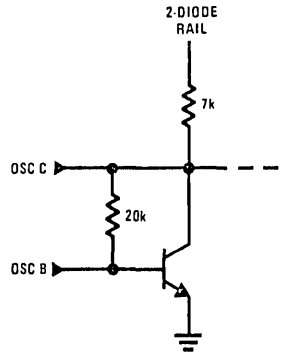


TL/F/5111–4

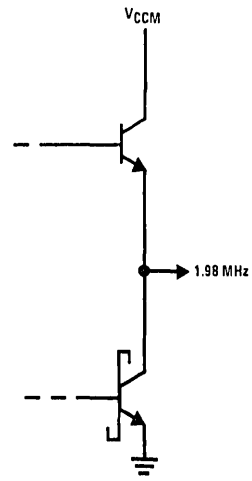
Schematic Diagrams (Continued)



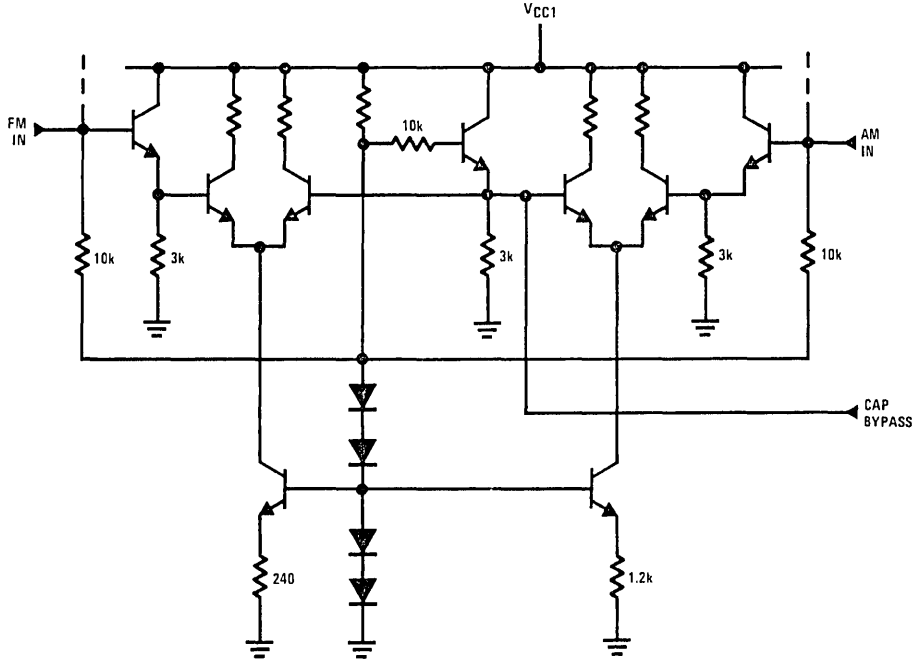
TL/F/5111–5



TL/F/5111–6

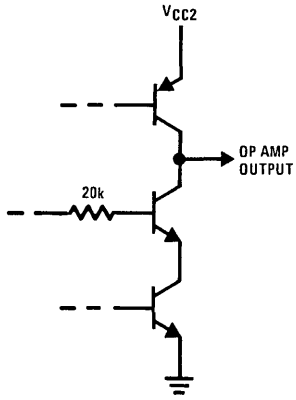


TL/F/5111–7



TL/F/5111–8

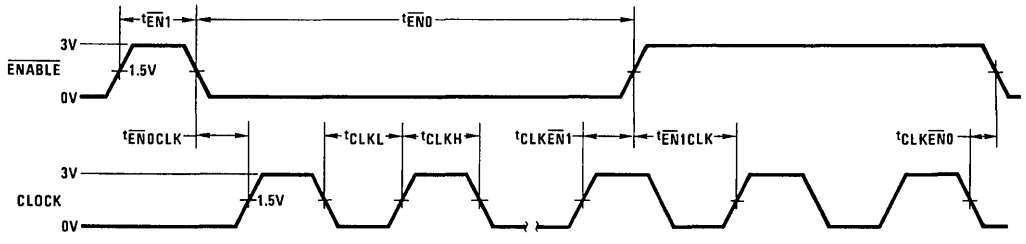
Schematic Diagrams (Continued)



TL/F/5111–9

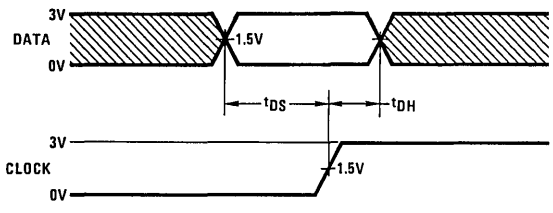
Timing Diagrams*

$\overline{\text{ENABLE}}$ vs CLOCK



TL/F/5111–10

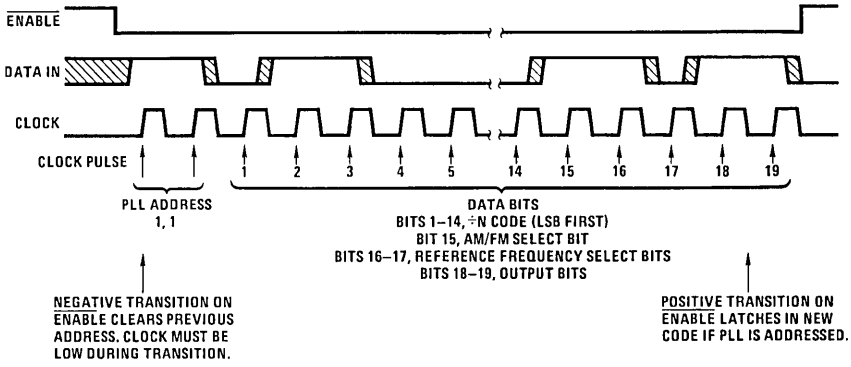
CLOCK vs DATA



TL/F/5111–11

Timing Diagrams*

AM/FM Frequency Synthesizer (Scan Mode)



TL/F/5111–12

*Timing diagrams are not drawn to scale. Scale within any one drawing may not be consistent, and intervals are defined positive as drawn.

SERIAL DATA ENTRY INTO THE DS8908

Serial information entry into the DS8908 is enabled by a low level on the $\overline{\text{ENABLE}}$ input. One binary bit is then accepted from the DATA input with each positive transition of the CLOCK input. The CLOCK input must be low for the specified time preceding and following the negative transition of the $\overline{\text{ENABLE}}$ input.

The first two bits accepted following the negative transition of the $\overline{\text{ENABLE}}$ input are interpreted as address. If these address bits are *not* 1,1 *no* further information will be accepted fromt he DATA inputs, and the internal data latches *will not* be changed when $\overline{\text{ENABLE}}$ returns high.

If these first two bits *are* 1,1, then all succeeding bits are *accepted* as data, and are shifted successively into the internal shift register as long as $\overline{\text{ENABLE}}$ remains low.

Any data bits preceding the 19th to last bit will be shifted out, and thus are irrelevant. Data bits are counted as any bits *following* two valid address bits (1,1) with the $\overline{\text{ENABLE}}$ low. When the $\overline{\text{ENABLE}}$ input returns high, any further serial data entry is inhibited. Upon this positive transition, the data in the internal shift register is transferred into the internal data latches. Note that until this time, the states of the internal data latches have remained unchanged.

These data bits are interpreted as follows:

Data Bit Position	Data Interpretation
Last	Bit 19 Output (Pin 2)
2nd to Last	Bit 18 Output (Pin 1)
3rd to Last	Ref. Freq. Select Bit(1) 17
4th to Last	Ref. Freq. Select Bit(1) 16
5th to Last	AM/FM Select Bit 15
6th to Last	(2^{13})
7th to Last	(2^{12})
8th to Last	(2^{11})
9th to Last	(2^{10})
10th to Last	(2^{9})
11th to Last	(2^{8})
12th to Last	(2^{7})
13th to Last	(2^{6})
14th to Last	(2^{5})
15th to Last	(2^{4})
16th to Last	(2^{3})
17th to Last	(2^{2})
18th to Last	(2^{1})
19th to Last	LSB of $\div N(2^{0})$

$\div N(2)$

Note 1: See Reference Frequency Select Truth Table.

Note 2: The actual divide code is N+1, ie., the number loaded plus 1.

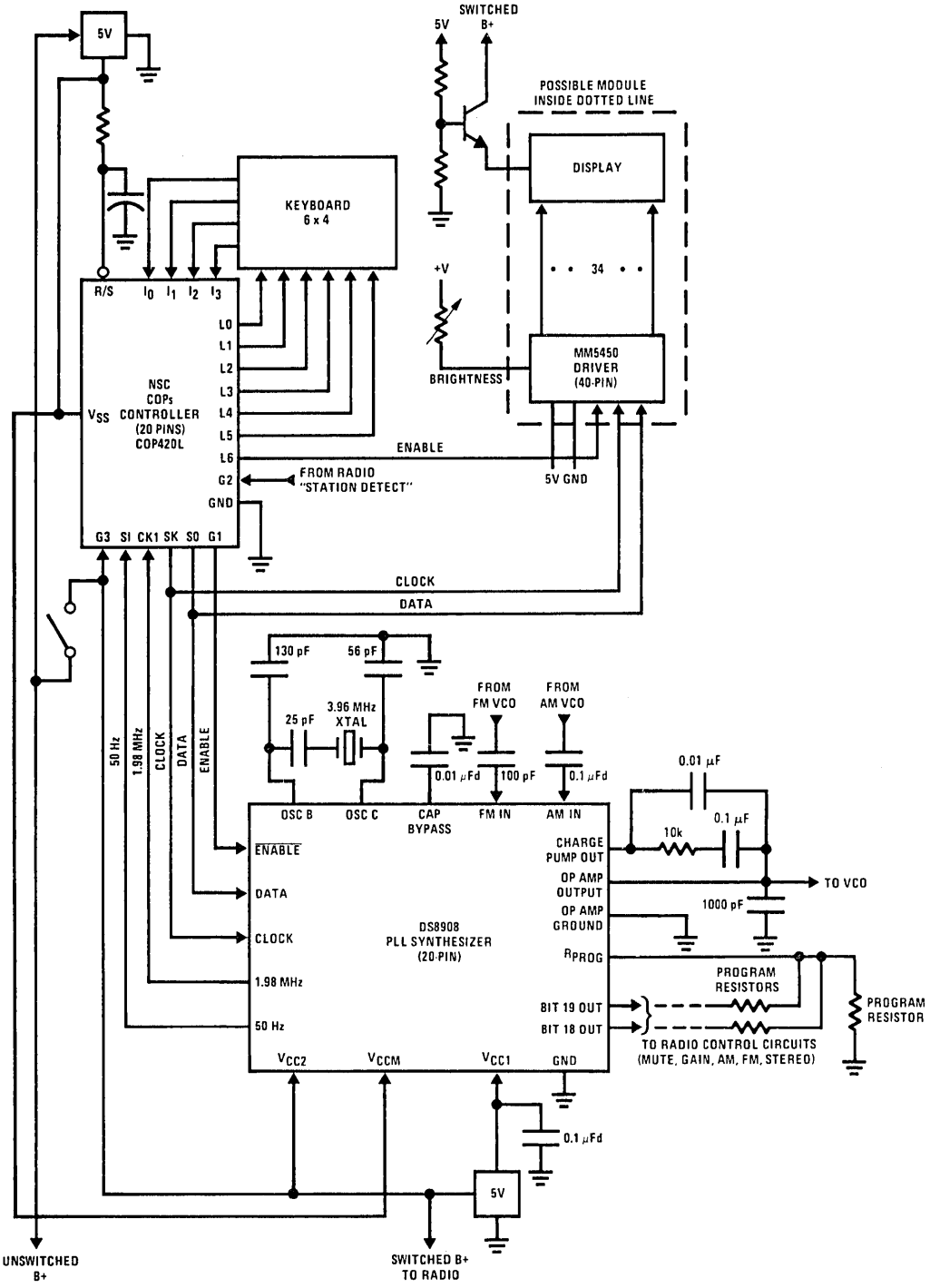
Truth Table

Reference Frequency Selection Truth Table

Serial Data		Reference Frequency
Bit 16	Bit 17	(kHz)
1	1	20
1	0	10
0	1	9
0	0	1

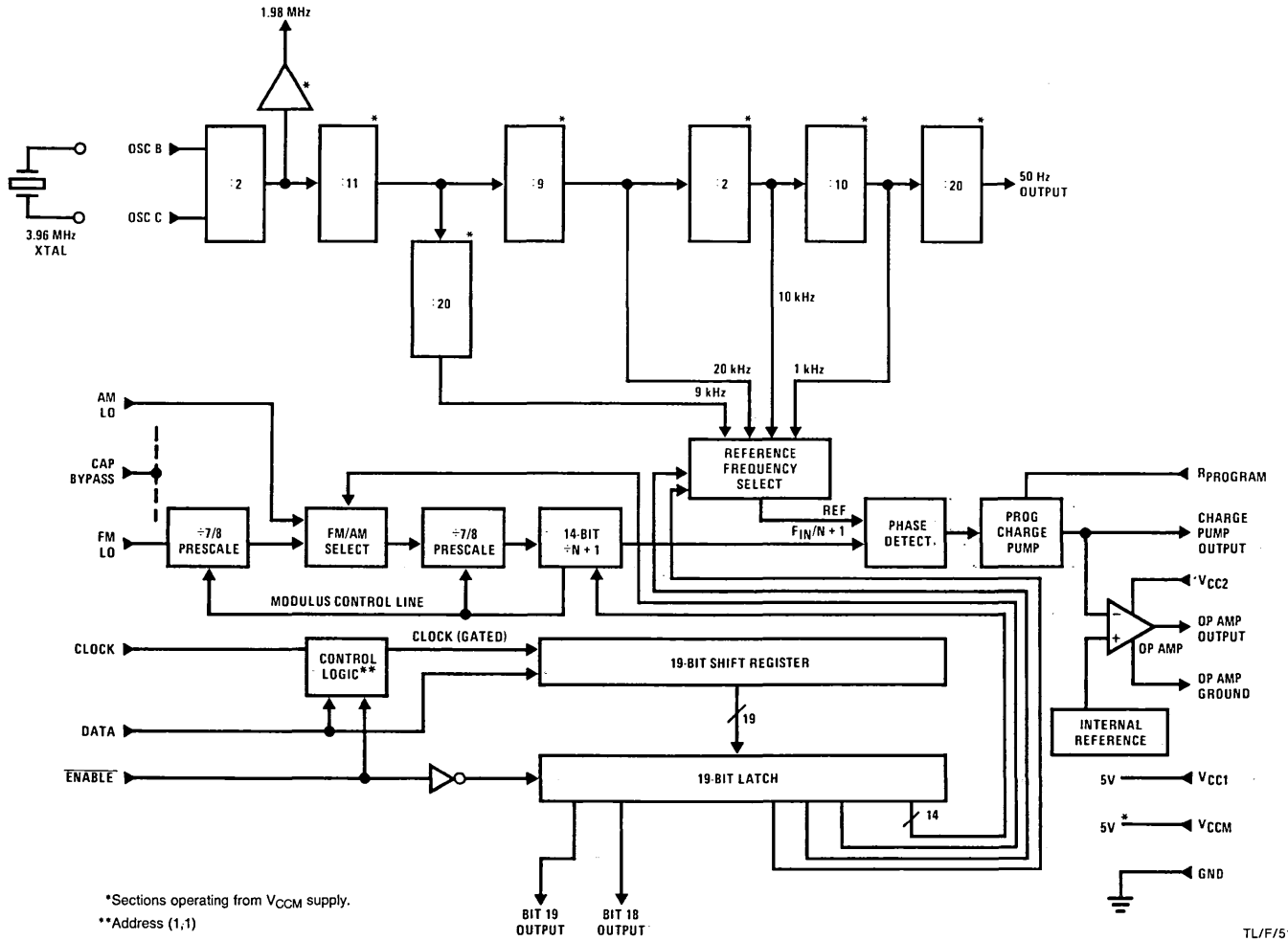
Typical Application Additional application notes are located at the back of section 11.

Electronically Tuned Radio Controller System; Direct Drive LED



TL/F/5111–13

AM/FM PLL/Synthesizer (Serial Data 20-Pin Package)



National Semiconductor Corporation

PRELIMINARY

DS8911/DS8912 AM/FM/TV Sound Up-Conversion Frequency Synthesizers

General Description

The DS8911 and DS8912 are digital Phase-Locked Loop (PLL) frequency synthesizers intended for use as Local Oscillators (LO) in electronically tuned radios. The devices are used in conjunction with a serial data controller, a loop filter, some varactor diodes and several passive elements to provide the local oscillator function for both AM and FM tuning.

The conventional superheterodyne AM receiver utilizes a low IF or down conversion tuning approach whereby the IF is chosen to be below the frequencies to be received. The DS8911 and DS8912 PLL's on the other hand, utilize an up-conversion technique in the AM mode whereby the first IF frequency is chosen to be well above the RF frequency range to be tuned. This approach eliminates the need for tuned circuits in the AM frontend since the image, half IF, and other spurious responses occur far beyond the range of frequencies to be tuned. Sufficient selectivity and second IF image protection is provided by a crystal filter at the output of the first mixer.

A significant cost savings can be realized utilizing this up-conversion approach to tuning. Removal of the AM tuned circuits eliminates the cost of expensive matched varactor diodes and reduces the amount of labor required for alignment down from 6 adjustments to 2. Additional cost savings are realized because up-conversion enables both the AM and FM bands to be tuned using a single Voltage Controlled Oscillator (VCO) operating between 98 and 120 MHz. (The 2 to 1 LO tuning range found in conventional AM down conversion radios is reduced to a 10% tuning range; 9.94 MHz to 11.02 MHz).

Up-conversion AM tuning is accomplished by first dividing the VCO signal down by a modulus 10 (DS8911) or 20 (DS8912) to obtain the LO signal. This LO in turn is mixed on chip with the RF signal to obtain a first IF at the MIXER output pins. This first IF after crystal filtering is mixed (externally) with a reference frequency provided by the PLL to obtain a 450 kHz second IF frequency. The DS8911 derives the 450 kHz second IF by mixing an 11.55 MHz first IF with a 12.00 MHz reference frequency. The DS8912 derives the 450 kHz second IF by mixing a 4.45 MHz first IF with a 4.00 MHz reference frequency.

FM and WB (weather band) tuning is done using the conventional down conversion approach. Here the VCO signal is buffered to produce the LO signal and then mixed on chip with the RF signal to obtain an IF frequency at the MIXER output pins. This IF frequency is typically chosen to be 10.7 MHz although placement at 11.50 MHz can further enhance AM mode performance and minimize IF circuitry.

The DS8911 was designed to utilize an 11.55 MHz crystal filter because of its superior phase noise and temperature drift characteristics. The DS8912 on the other hand was designed to utilize a 4.45 MHz ceramic filter for cost savings in applications not requiring high performance.

Both PLLs provide phase comparator reference frequencies of 10, 12.5, 25, and 100 kHz. The tuning resolutions resulting from these reference frequencies are determined by dividing the reference by the premix modulus. Table II shows the tuning resolutions possible.

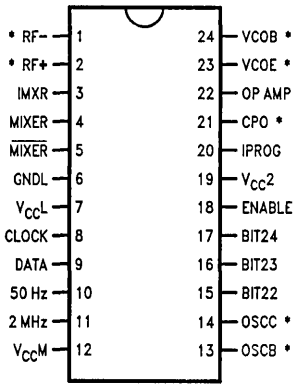
The DS8911 and DS8912 contain the following logic elements: a voltage controlled oscillator, a reference oscillator, a 14-bit programmable dual-modulus counter, a reference frequency divider chain, a premix divider, a mixer, a phase comparator, a charge pump, an operational amplifier, and control circuitry for latched serial data entry.

Features

- Direct synthesis of LW, MW, SW, FM, and WB frequencies
- Serial data entry for simplified processor control
- 10, 12.5, 25, and 100 kHz reference frequencies
- 8 possible tuning resolutions (see Table II)
- An op amp with high impedance inputs for loop filtering
- Programmable mixer with high dynamic range
- Fast-lock feature for Automatic Road Information (ARI) systems

Connection Diagrams

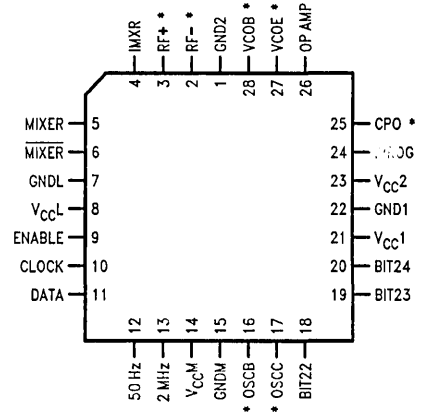
SO and Dual-In-Line Packages



TL/F/7398–7

Top View

Plastic Chip Carrier



TL/F/7398–8

Top View

Order Number DS8911M, DS8911N, DS8911V, DS8912M, DS8912N or DS8912V
See NS Package Number M24B, N24C or V28A

Note: Device pins marked with an asterisk (*) are not guaranteed to meet the NSC standard requirement for Electrostatic Discharge (ESD) protection of 2000 volts. The functional requirements of the application prohibit the additional resistive or capacitive components required for ESD protection on these pins.

Pin Descriptions

V_{CC1}: The V_{CC1} pin provides a 5V supply source for all circuitry except the reference divider chain, op amp and mixer sections of the die.

V_{CC2}: The V_{CC2} pin provides a 12V supply source for the Op amp.

V_{CCL}: The V_{CCL} pin provides an isolated 5V supply source for the premix divider and mixer functions.

V_{CCM}: The V_{CCM} pin provides a 5V supply source for the reference oscillator and divider chain down through the 50 Hz output, thus enabling low standby current for time-of-day clock applications.

GND1, GND2, GNDL and GNDM: Provide isolated circuit ground for the various sections of the device.

DATA and CLOCK: The DATA and CLOCK inputs are for serial data entry from a controller. They are CMOS inputs with TTL logic thresholds. The 24-bit data stream is loaded into the PLL on the positive transition of the CLOCK. The first 14 bits of the data stream select PLL divide code in binary form MSB first. The 15th through 24th bits select the premix modulus, the reference frequency, the loop response mode, the bit output status, and the test/operate modes as shown in Tables I through V.

ENABLE: The ENABLE input is a CMOS input with a TTL logic threshold. The ENABLE input enables data when at a logic "one" and latches data on the transition to a logic "zero".

BIT Outputs: The open-collector BIT outputs provide either the status of shift register bits 22, 23, and 24 or enable access to key internal circuit test nodes. The mode for the bit outputs is controlled by shift register bits 20 and 21. In operation, the bit outputs are intended to drive radio functions such as gain, mute, and AM/FM status. These outputs can also be used to program the loop gain by connection of an external resistor to IPROG. Bit 24 output can also be used as a 300 millisecond timer under control of shift register bit 19. During service testing, these pins can be used for the purpose of either monitoring or driving internal logic points as indicated in the TEST MODES description under Table V.

VCOb and VCOe: The Voltage Controlled Oscillator inputs drive the 14-bit programmable counter and the premix divider. These inputs are the base and emitter leads of a transistor which require connection of a coil, varactor, and several capacitors to function as a Colpitts oscillator. The VCO is designed to operate up to 225 MHz. The VCO's minimum operating frequency may be limited by the choice of reference frequency and the 961 minimum modulus constraint of the 31/32 dual modulus counter.

RF+ and RF−: The Radio Frequency inputs are fed differentially into the mixer.

IMXR: The bias current for the mixer is programmed by connection of external resistors.

MIXER and $\overline{\text{MIXER}}$: The MIXER outputs are the collectors of the double balanced pair mixer transistors. They are intended to operate at voltages greater than V_{CC1}.

OSCb and OSCc: The Reference Oscillator inputs are part of an on-chip Pierce oscillator designed to work in conjunction with 2 capacitors and a crystal resonator. The DS8911 requires a 12 MHz crystal, while the DS8912 requires a 4 MHz crystal.

The OSC input signal is mixed externally with the 1st AM IF output to obtain a 450 kHz 2nd IF frequency in the AM mode.

2 MHz: The 2 MHz output is provided to drive a controller's clock input.

Pin Descriptions (Continued)

50 Hz: The 50 Hz output is provided as a time reference for radios with time-of-day clocks.

IPROG: The IPROG pin enables the charge pump to be programmed from .5 mA to 1.5 mA by connection of an extenal resistor to ground.

CPO: The Charge Pump Output circuit sources current if the VCO frequency is high and sinks current if the VCO frequency is low. The CPO is wired directly to the negative input of the loop filter op amp.

OP AMP: The OP AMP output is provided for loop filtering. The op amp has high impedance PMOS gate inputs and is wired as a transconductance amplifier/filter. The op amp's positive input is internally referenced while it's negative input is common with the CPO output.

Reference Tables DS8911 (DS8912)

TABLE I

Bit 15	Premix Modulus
0	÷1
1	÷10 (÷20)

TABLE II

Bit		Reference Frequency	Tuning Resolution	
16	17		÷1 Premix	÷10 (÷20) Premix
0	0	10 kHz	10 kHz	1 (.5) kHz
0	1	12.5 kHz	12.5 kHz	1.25 (.625) kHz
1	0	25 kHz	25 kHz	2.5 (1.25) kHz
1	1	100 kHz	100 kHz	10 (5) kHz

TABLE III

Bit 18	Loop Response
0	Normal Lock
1	Fast Lock

FAST LOCK OPERATION

The fast lock mode provides a means of moving from one frequency selection to another frequency selection anywhere on the band in a very short time frame. This is accomplished by setting a bit in the microprocessor serial data stream when loading a new frequency. When fast lock is activated the charge pump output (CPO) is latched into the pump up or down state, which drives the CPO at the maximum rate to correct the VCO frequency. The PLL meanwhile operates in a frequency lock mode, constantly comparing the frequencies and reducing any phase discrepancies. When the VCO passes beyond the desired lock frequency the CPO unlatches and reverts back to the phase lock mode of operation. The frequency lock mode of operation (during CPO latchup) ensures that the phases are always close and will quickly settle into phase lock once the CPO unlatches.

TABLE IV

Bit 19	Timer
0	Bit 24 Status
1	$\overline{\text{Bit 24}}$ for 300 ms

TIMER OPERATION

The timer function is provided for use as a retriggerable "one shot" to enable muting for approximately 300 milliseconds after station changes. The timer is enabled at bit 24's output if the normal operating mode is selected (shift register bits 20 and 21 = "LOW") and shift register bit 19 data is latched as a "HI". The timer's output state will invert immediately upon latching bit 19 "HI" and remain inverted for approximately 300 milliseconds. If the user readdresses the device with bit 19 data "LOW" before the timer finishes its cycle the timer's BIT 24 output will finish out the 300 ms pulse. Readdressing the device with bit 19 "HI" before the timer finishes its cycle will extend the BIT 24 output pulse width by 300 ms. Addressing should be performed immediately after the 50 Hz output transitions "HI". BIT 24's output state is not guaranteed during the first 300 ms after V_{CC_1} power up as a result of a timer reset in progress.

TABLE V

Bit		FUNCTION OF PINS 3, 4, & 5
20	21	
0	0	Status of Bits 22-24
0	1	Test mode 1
1	0	Test mode 2
1	1	Test mode 3

TEST MODE OPERATION

Test Mode 1: Enables the BIT output pins to edge trigger the phase comparator inputs and monitor an internal lock detector. BIT 22 negative edge triggers the reference divider input of the phase comparator if the reference divider state is low. BIT 23 provides the open collector ORing of the phase comparator's pump up and down outputs. BIT 24 negative edge triggers the N counter input of the phase comparator if the N counter state is preconditioned low.

Test Mode 2: Enables the BIT outputs to clock the programmable N counter, monitor its output, and force either its load or count condition. BIT 22 provides the N counter output which negative edge triggers the phase comparator and which appears low one N counter clock pulse before it reloads. BIT 23 positive edge triggers the N counter's clock input if the prescaler's output is preconditioned HI. BIT 24 clears the N counter output so that loading will occur on the next N counter clock edge.

Test Mode 3: Enables the BIT outputs to clock the 50 Hz and 10 kHz reference dividers, monitor the reference divider input to the phase comparator, and reset the fast lock latch. BIT 22 positive edge clocks the 10 kHz reference divider chain if the 10 kHz output is preconditioned HI. BIT 22 also positive edge clears the fast lock latch condition. Bit 23 positive edge clocks the 50 Hz divider chain. BIT 24 is the reference divider negative edge trigger input to the phase comparator.

Absolute Maximum Ratings (Note 1)

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage
- V_{CCM} 7V
- V_{CC1} 7V
- V_{CC2} 15V

Input Voltage 7V

Output Voltage
- Logic 7V
- Op Amp and Mixer Outputs 15V

Storage Temperature Range −65°C to +150°C

Lead Temp. (Soldering, 10 seconds) 300°C

Operating Conditions

	Min	Max	Units
V_{CCM}	3.5	5.5	V
V_{CC1}	4.5	5.5	V
V_{CC2}	7.0	12.0	V
Temperature, T_A	−40	+85	°C

DC Electrical Characteristics (Notes 2 and 3)

Symbol	Parameter		Test Conditions		Min	Typ	Max	Units
V_{IH}	Logic "1" Input Voltage				2.0			V
V_{IL}	Logic "0" Input Voltage						0.8	V
I_{IH}	Logic "1" Input Current		V_{IN} = 5.5V				10	μA
I_{IL}	Logic "0" Input Current		Data, Clock and Enable Inputs, V_{IN} = 0V				−10	μA
V_{OH}	Logic "1" Output Voltage	2 MHz	I_{OH} = −20 μA		V_{CCM}−0.3			V
			I_{OH} = −400 μA		V_{CCM}−2			V
		Op Amp	I_{OH} = −1.5 mA		V_{CC_2}−1.5			V
V_{OL}	Logic "0" Output Voltage	2 MHz	I_{OL} = 20 μA				0.3	V
			I_{OL} = 400 μA				0.4	V
		50 Hz	I_{OL} = 250 μA				0.3	V
		Bit Outputs	I_{OL} = 1 mA				0.3	V
		Op Amp	I_{OL} = 1.5 mA				1.5	V
V_{BIAS}	Op Amp Input VΔ		Op Amp I/O Shorted, V_{CC1} = 5.5V, V_{CC2} = 12V, CPO = TRI-STATE®, Op Amp I_{OH} vs. I_{OL} Applied				150	mV
I_{CEX}	High Level Output Current	Bit Outputs	V_{CC_1} = 4.5V, V_O = 8.8V				100	μA
		50 Hz	V_{CC_M} = 3.5V, V_O = 5.5V				10	μA
		Mixers	V_{CC_L} = V_{CC_1} = 4.5V, V_O = 8.8V				100	μA
I_{CPO}	Charge Pump Program Current		0.5 mA < I_{CPO} < 1.5 mA, 2 I_{PROG} = V_{CC_1}/R_{PROG}, Measured I_{PROG} to CPO	Pump-up	−20	2 I_{PROG}	+20	%
				Pump-down	−20	2 I_{PRCG}	+20	%
				TRI-STATE		0	100	nA
I_{CCM}	V_{CCM} Supply Current		V_{CCM} = 5.5V			0.5	1.0	mA
I_{CC1} + I_{CCL}	V_{CC1} + V_{CCL} Supply Current		V_{CC} = 5.5V, Bits Hi, I_{MXR} and I_{PROG} Open			25	35	mA
I_{CC2}	V_{CC2} Supply Current		V_{CC2} = 12V			1.5	2.5	mA
R_{IN}	Mixer Input Impedance		I_{MXR} = 2.5 mA I_{MXR} = 7.5 mA			TBD TBD		Ω Ω
R_{OUT}	Mixer Output Impedance		I_{MXR} = 2.5 mA I_{MXR} = 7.5 mA			TBD TBD		Ω Ω
g_m	Mixer Transconductance		I_{MXR} = 2.5 mA I_{MXR} = 7.5 mA			TBD TBD		mhos mhos
NF	Noise Figure		I_{MXR} = 2.5 mA, R_S = 50Ω I_{MXR} = 7.5 mA, R_S = 10Ω			TBD TBD		dB dB
XMOD	Cross Modulation		I_{MXR} = 2.5 mA I_{MXR} = 7.5 mA			TBD TBD		mVRms mVRms
I_{MXR}	Mixer Current		V_{CC1} = 5.5V		1		7.5	mA
VCO_{MAX}	VCO_{MAX} frequency						225	MHz

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range", they are not meant to imply that the device should be operated at these limits.

Note 2: Unless otherwise specified, min/max limits apply across the −40°C to +85°C temperature range for DS8911 and DS8912.

Note 3: All currents into device pins are shown as positive, out of device pins as negative, all voltage referenced to ground unless otherwise noted. All values shown as maximum or minimum on absolute value basis.

AC Electrical Characteristics (Note 2)

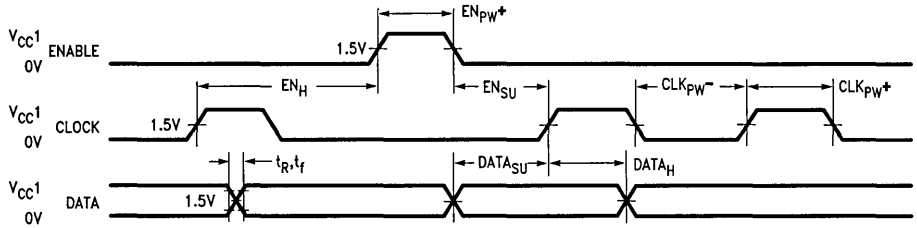
Symbol	Parameter	Conditions	Min	Max	Units
t_r	20%–80% Rise Time	V_{CC_1} = 4.5V to 5.5V		200	ns
t_f	80%–20% Fall Time			200	ns
$DATA_{SU}$	Data Setup Time		100		ns
$DATA_H$	Data Hold Time		100		ns
EN_{SU}	Enable Setup Time		100		ns
EN_H	Enable Hold Time		100		ns
EN_{PW+}	Enable Positive Pulse Width		200		ns
CLK_{PW+}	Clock Positive Pulse Width		200		ns
CLK_{PW-}	Clock Negative Pulse Width		200		ns

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range", they are not meant to imply that the device should be operated at these limits.

Note 2: Unless otherwise specified, min/max limits apply across the −40°C to +85°C temperature range for DS8911 and DS8912.

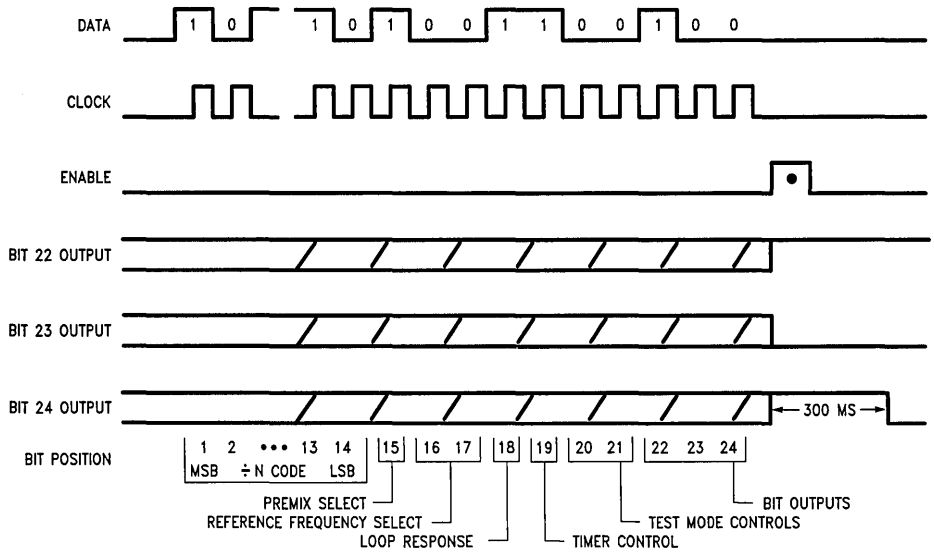
Note 3: All currents into device pins are shown as positive, out of device pins as negative, all voltage referenced to ground unless otherwise noted. All values shown as maximum or minimum on absolute value basis.

Timing Diagram



TL/F/7398–10

MICROWIRE™ Bus Format



TL/F/7398–19

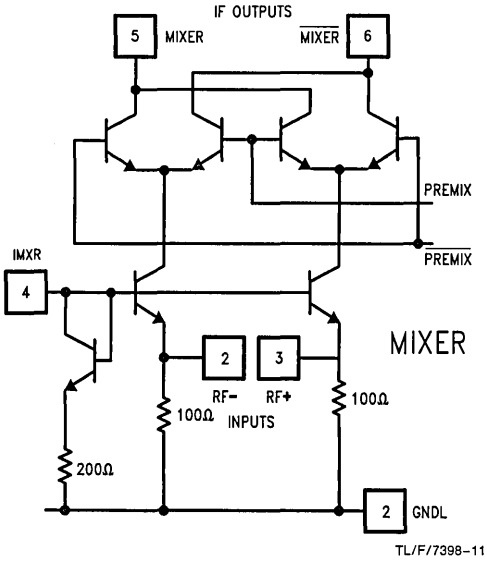
TABLE VI. DS8911 Tuning Characteristics

Mode	IF Frequency (MHz)	Tuning Range (MHz)	VCO Range (MHz)	Premix Modulus	Reference Frequency (kHz)	Tuning Resolution (kHz)	Image (MHz)
LW	11.55/.450	.145–.290	112.4–114.1	10	10	1	22–23
MW	11.55/.450	.515–1.61	99.4–110.2	10	10, 12.5, 25, 100	1, 1.25, 2.5, 10	21–23
SW	11.55/.450	5.94–6.2	53.5 to 56.1	10	10, 12.5, 25	1, 1.25, 2.5	28–30
FM	10.7	87.4–108.1	98.1–118.8	1	10, 12.5, 25, 100	10, 12.5, 25, 100	109–130
WB	10.7	162.4–162.6	151–152	1	12.5, 25	12.5, 25	140–142
TV_1	10.7	59.75–87.75	70.45–98.45	1	25	25	81–109
TV_2	10.7	179.75–215.75	169.1–205.1	1	25	25	158–194

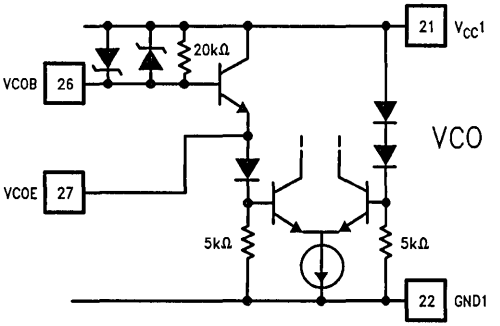
TABLE VII. DS8912 Tuning Characteristics

Mode	IF Frequency (MHz)	Tuning Range (MHz)	VCO Range (MHz)	Premix Modulus	Reference Frequency (kHz)	Tuning Resolution (kHz)	Image (MHz)
LW	4.45/.450	.145–.290	91.9–94.8	20	10	.5	13–14
MW	4.45/.450	.515–1.61	99.6–121.2	20	10, 12.5, 25, 100	.5, .625, 1.25, 5	14–17
SW	4.45/.450	5.94–6.2	207.9–213.1	20	25, 100	1.25, 5	14–16
FM	10.7	87.4–108.1	98.1–118.8	1	10, 12.5, 25, 100	10, 12.5, 25, 100	109–130
WB	10.7	162.4–162.6	151–152	1	12.5, 25	12.5, 25	140–142

Input and Output Schematics

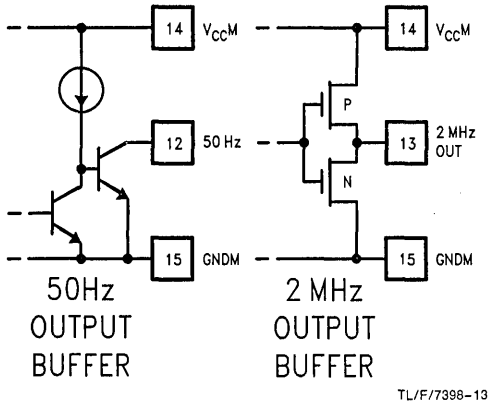


TL/F/7398–11

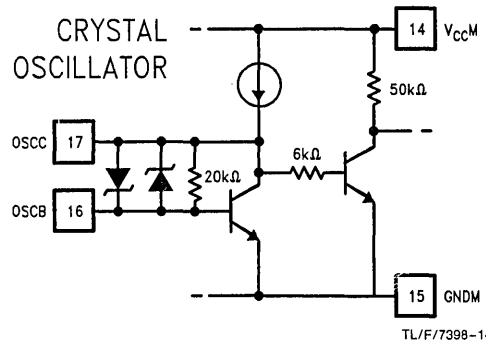


TL/F/7398–12

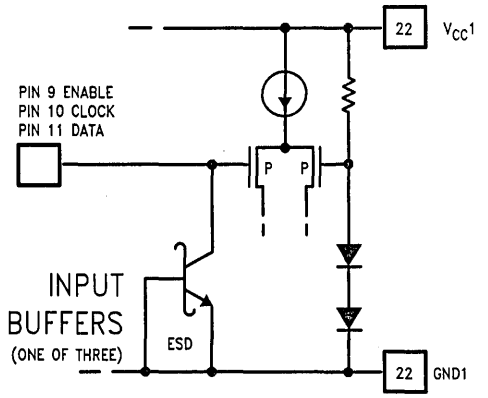
Input and Output Schematics (Continued)



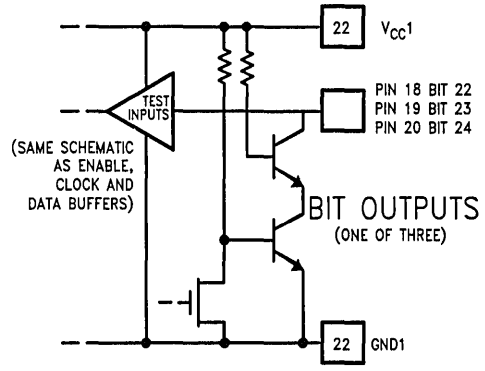
TL/F/7398–13



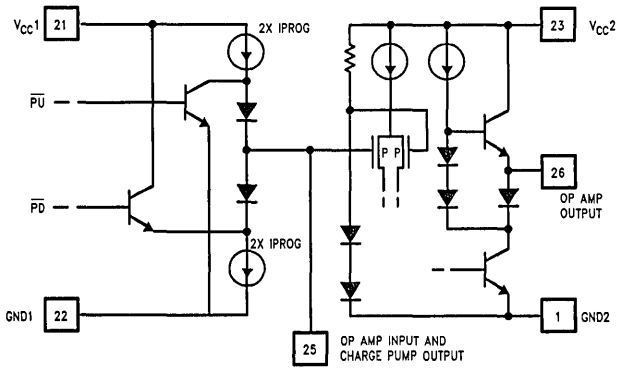
TL/F/7398–14



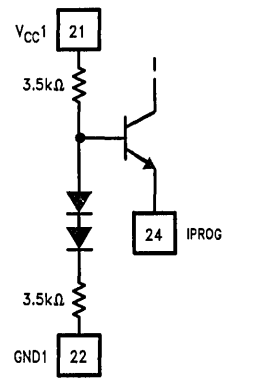
TL/F/7398–15



TL/F/7398–16

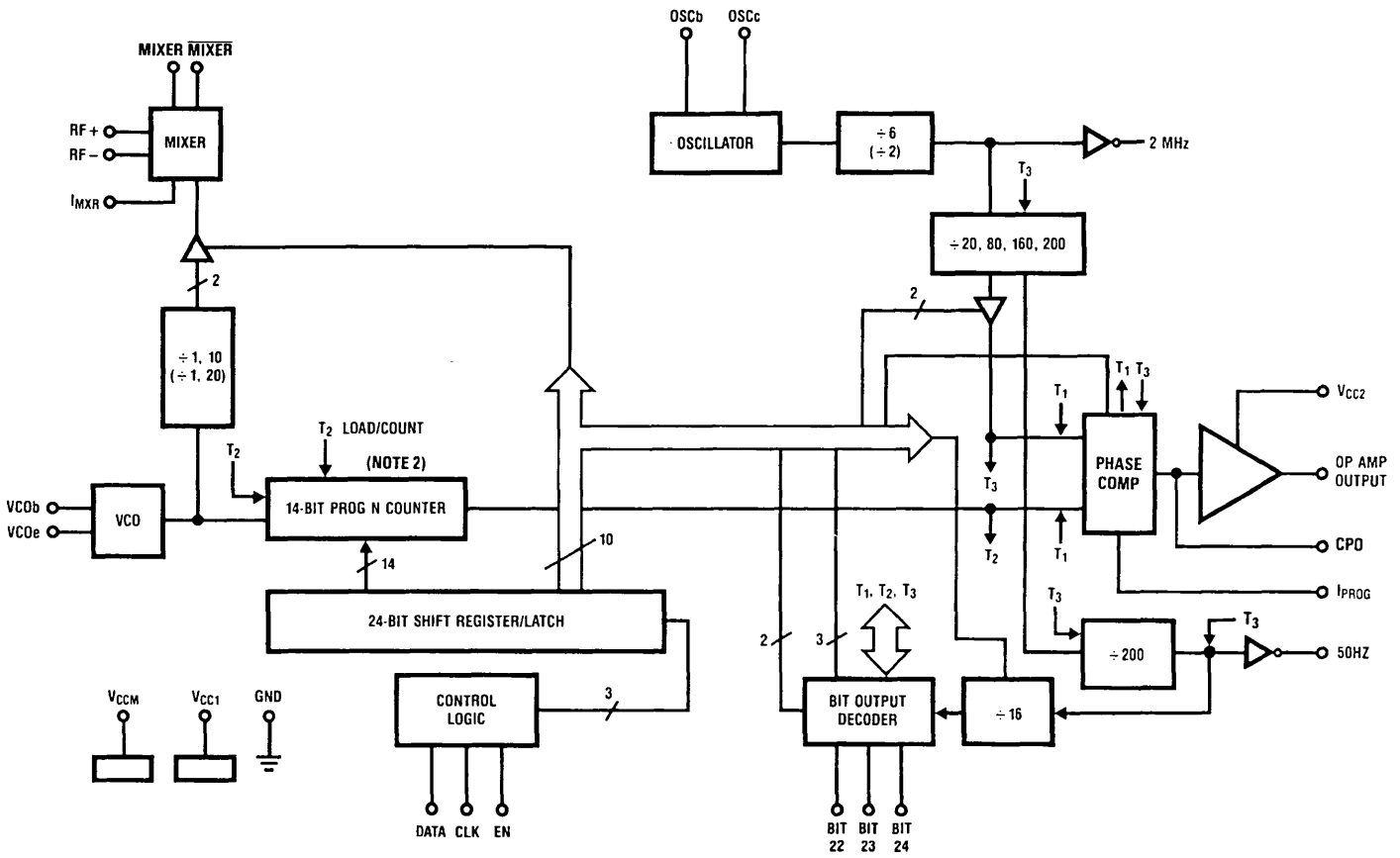


TL/F/7398–17



TL/F/7398–18

DS8911 (DS8912) PLL Synthesizer



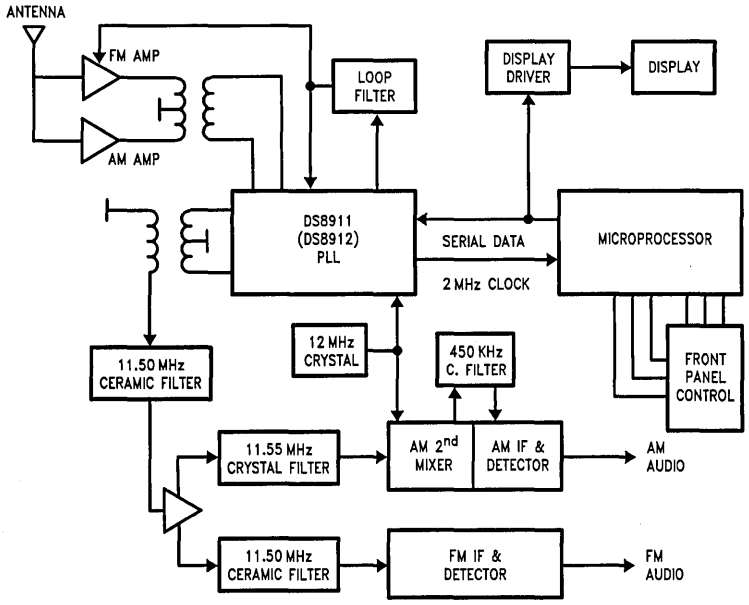
Note 1: Logic labels (in parentheses) pertain to the DS8912 option.

TL/F/7398–4

Note 2: The 14 bit programmable N counter is a dual modulus counter with 31/32 prescaler. The minimum continuous modulus of the N counter is 961. (There are a limited number of valid modulus codes below 961.)

Typical Application

AM/FM ETR Radio Application



TL/F/7398–5

National Semiconductor Corporation

MM5450/MM5451 LED Display Drivers

General Description

The MM5450 and MM5451 are monolithic MOS integrated circuits utilizing N-channel metal-gate low threshold, enhancement mode, and ion-implanted depletion mode devices. They are available in 40-pin molded or cavity dual-in-line packages. A single pin controls the LED display brightness by setting a reference current through a variable resistor connected to V_{DD}.

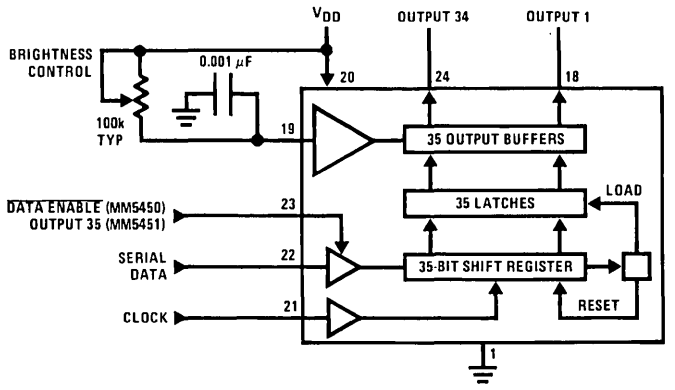
Features

- Continuous brightness control
- Serial data input
- No load signal required
- Enable (on MM5450)
- Wide power supply operation
- TTL compatibility
- 34 or 35 outputs, 15 mA sink capability
- Alphanumeric capability

Applications

- COPS™ or microprocessor displays
- Industrial control indicator
- Relay driver
- Digital clock, thermometer, counter, voltmeter
- Instrumentation readouts

Block Diagram



TL/F/6136–1

FIGURE 1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin	V_{SS} to V_{SS} + 12V
Operating Temperature	−25°C to +85°C
Storage Temperature	−65°C to +150°C
Power Dissipation	560 mW at +85°C 1W at +25°C
Junction Temperature	+150°C
Lead Temperature (Soldering, 10 sec.)	300°C

Electrical Characteristics

T_A within operating range, V_{DD} = 4.5V to 11.0V, V_{SS} = 0V unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
Power Supply		4.75		11	V
Power Supply Current	Excluding Output Loads			7	mA
Input Voltages					
Logical "0" Level (V_L)	±10 μA Input Bias	−0.3		0.8	V
Logical "1" Level (V_H)	4.75V ≤ V_{DD} ≤ 5.25V	2.2		V_{DD}	V
	V_{DD} > 5.25V	V_{DD} − 2V		V_{DD}	V
Brightness Input (Note 2)		0		0.75	mA
Output Sink Current					
Segment OFF	V_{OUT} = 3.0V			10	μA
Segment ON	V_{OUT} = 1V (Note 3)	0		15	mA
	Brightness Input = 0 μA	0		10	μA
	Brightness Input = 100 μA	2.0	2.7	4	mA
	Brightness Input = 750 μA	15		25	mA
Brightness Input Voltage (Pin 19)	Input Current 750 μA	3.0		4.3	V
Output Matching (Note 1)				±20	%
Clock Input	(Notes 5 and 6)				
Frequency, f_C				500	kHz
High Time, t_h		950			ns
Low Time, t_l		950			ns
Data Input					
Set-Up Time, t_{DS}		300			ns
Hole Time, t_{DH}		300			ns
Data Enable Input					
Set-Up Time, t_{DES}		100			ns

Note 1: Output matching is calculated as the percent variation $(I_{MAX} + I_{MIN})/2$.

Note 2: With a fixed resistor on the brightness input pin, some variation in brightness will occur from one device to another. Maximum brightness input current can be 2 mA as long as Note 3 and junction temperature equation are complied with.

Note 3: See *Figures 5, 6,* and *7* for Recommended Operating Conditions and limits. Absolute maximum for each output should be limited to 40 mA.

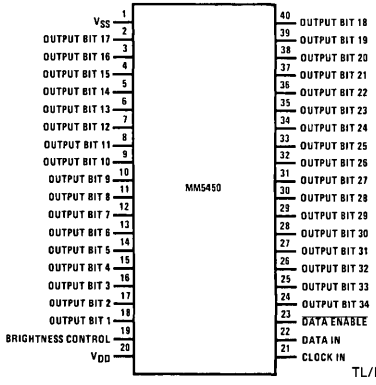
Note 4: The V_{OUT} voltage should be regulated by the user. See *Figures 6* and *7* for allowable V_{OUT} vs I_{OUT} operation.

Note 5: AC input waveform specification for test purpose: $t_r \leq 20$ ns, $t_f \leq 20$ ns, f = 500 kHz, 50% ±10% duty cycle.

Note 6: Clock input rise and fall times must not exceed 300 ns.

Connection Diagrams

Dual-In-Line Package

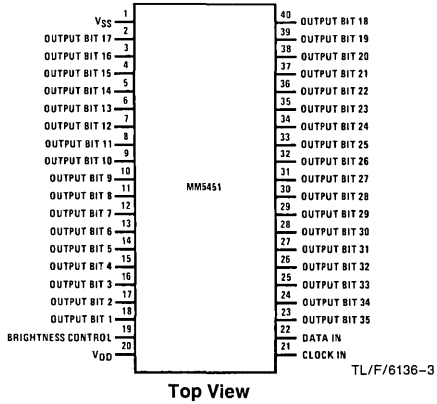


TL/F/6136–2

Top View

FIGURE 2a

Dual-In-Line Package



TL/F/6136–3

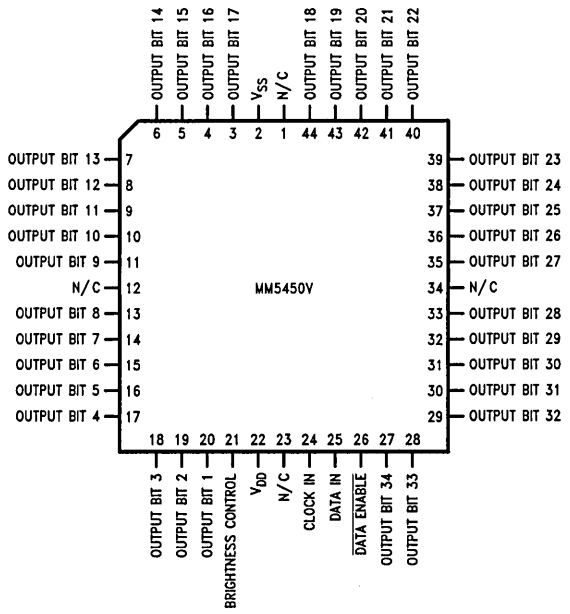
Top View

FIGURE 2b

Order Number MM5450N, MM5451N, MM5450V or MM5451V
See NS Package Number N40A or V44A

Connection Diagrams (Continued)

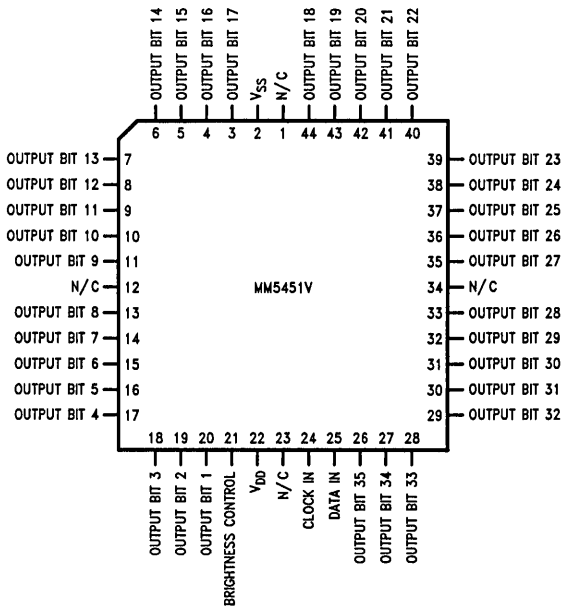
Plastic Chip Carrier



TL/F/6136–13

Top View

Plastic Chip Carrier



TL/F/6136–14

Top View

Functional Description

Both the MM5450 and the MM5451 are specifically designed to operate 4- or 5-digit alphanumeric displays with minimal interface with the display and the data source. Serial data transfer from the data source to the display driver is accomplished with 2 signals, serial data and clock. Using a format of a leading "1" followed by the 35 data bits allows data transfer without an additional load signal. The 35 data bits are latched after the 36th bit is complete, thus providing non-multiplexed, direct drive to the display. Outputs change only if the serial data bits differ from the previous time. Display brightness is determined by control of the output current for LED displays. A 0.001 capacitor should be connected to brightness control, pin 19, to prevent possible oscillations.

A block diagram is shown in *Figure 1.* For the MM5450 a $\overline{\text{DATA ENABLE}}$ is used instead of the 35th output. The $\overline{\text{DATA ENABLE}}$ input is a metal option for the MM5450. The output current is typically 20 times greater than the current into pin 19, which is set by an external variable resistor. There is an internal limiting resistor of 400Ω nominal value.

Figure 4 shows the input data format. A start bit of logical "1" precedes the 35 bits of data. At the 36th clock a LOAD signal is generated synchronously with the high state of the clock, which loads the 35 bits of the shift registers into the latches. At the low state of the clock a RESET signal is generated which clears all the shift registers for the next set of data. The shift registers are static master-slave configuration. There is no clear for the master portion of the first shift register, thus allowing continuous operation.

There must be a complete set of 36 clocks or the shift registers will not clear.

When the chip first powers ON an internal power ON reset signal is generated which resets all registers and all latches. The START bit and the first clock return the chip to its normal operation.

Figure 2 shows the pin-out of the MM5450 and MM5451. Bit 1 is the first bit following the start bit and it will appear on pin 18. A logical "1" at the input will turn on the appropriate LED.

Figure 3 shows the timing relationships between data, clock and $\overline{\text{DATA ENABLE}}$. A max clock frequency of 0.5 MHz is assumed.

For applications where a lesser number of outputs are used, it is possible to either increase the current per output, or operate the part at higher than 1V V_{OUT}. The following equation can be used for calculations.

$$T_j = (V_{OUT})(I_{LED})(\text{No. of segments})(124°C/W) + T_A$$

where:

T_j = junction temperature + 150°C max

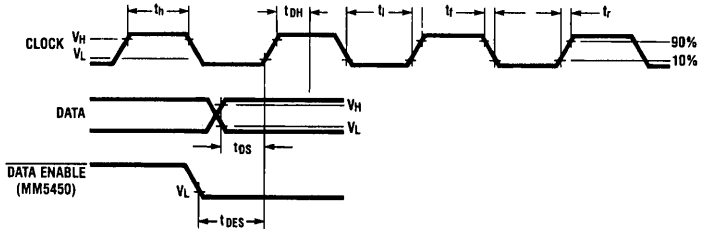
V_{OUT} = the voltage at the LED driver outputs

I_{LED} = the LED current

124°C/W = thermal coefficient of the package

T_A = ambient temperature

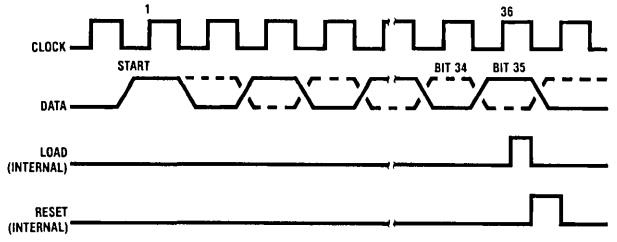
The above equation was used to plot *Figure 5, Figure 6* and *Figure 7.*



TL/F/6136–4

FIGURE 3

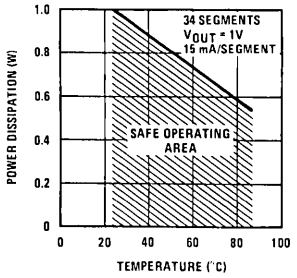
Functional Description (Continued)



TL/F/6136–5

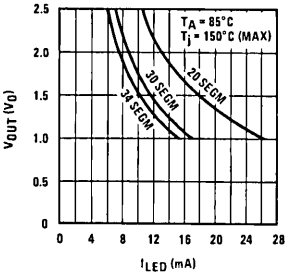
FIGURE 4. Input Data Format

Typical Performance Characteristics



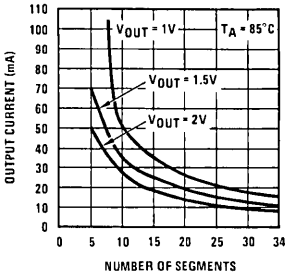
TL/F/6136–6

FIGURE 5



TL/F/6136–7

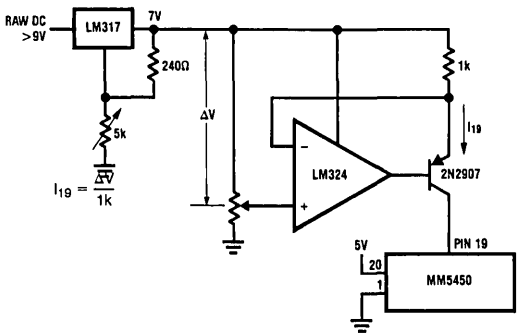
FIGURE 6



TL/F/6136–8

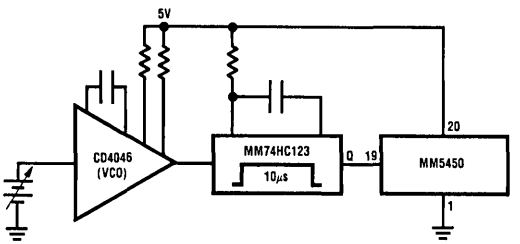
FIGURE 7

Typical Applications



TL/F/6136–9

FIGURE 8. Typical Application of Constant Current Brightness Control

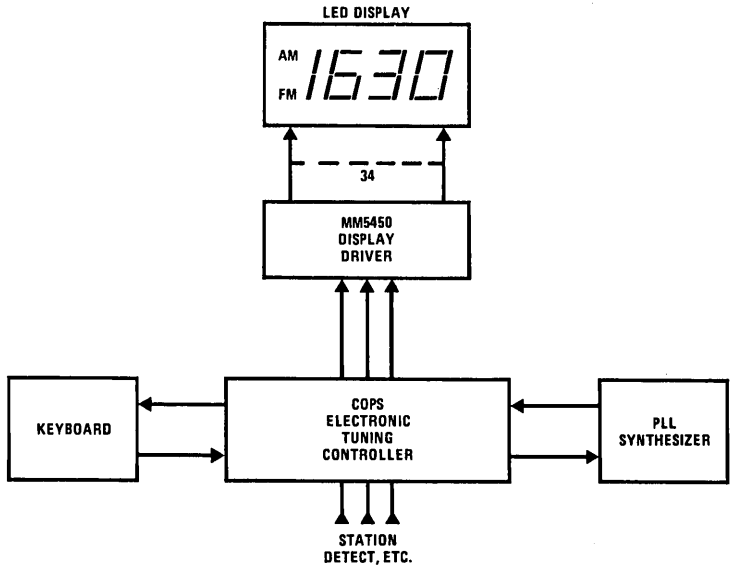


TL/F/6136–10

FIGURE 9. Brightness Control Varying the Duty Cycle

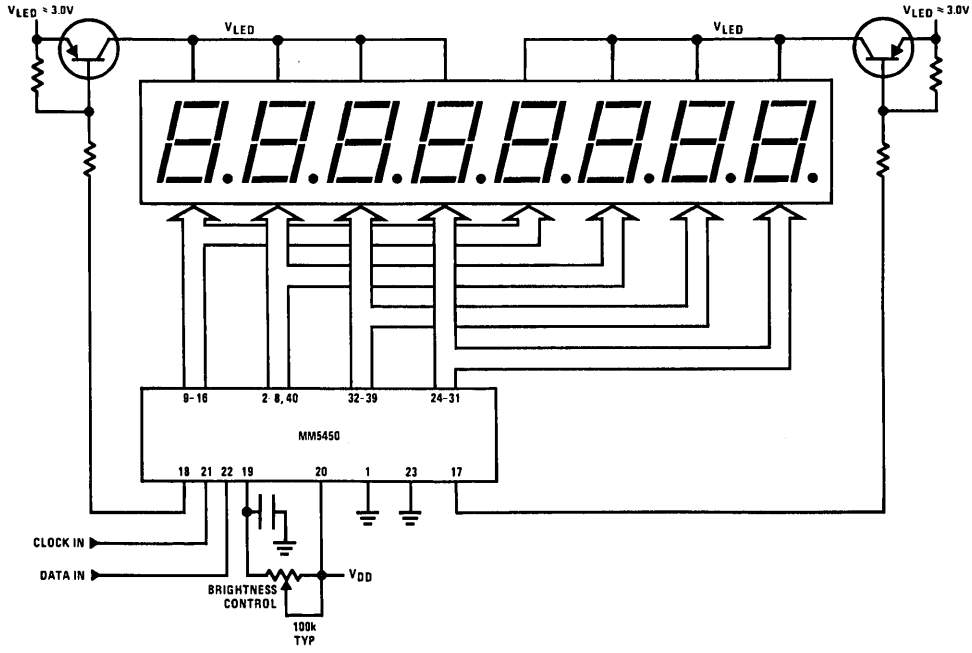
Typical Applications (Continued)

Basic Electronically Tuned Radio System



TL/F/6136–11

Duplexing 8 Digits with One MM5450



TL/F/6136–12

National Semiconductor Corporation

MM5452/MM5453 Liquid Crystal Display Drivers

General Description

The MM5452 is a monolithic integrated circuit utilizing CMOS metal gate, low threshold enhancement mode devices. It is available in a 40-pin molded package. The chip can drive up to 32 segments of LCD and can be paralleled to increase this number. The chip is capable of driving a 4 ½-digit 7-segment display with minimal interface between the display and the data source.

The MM5452 stores display data in latches after it is clocked in, and holds the data until new display data is received.

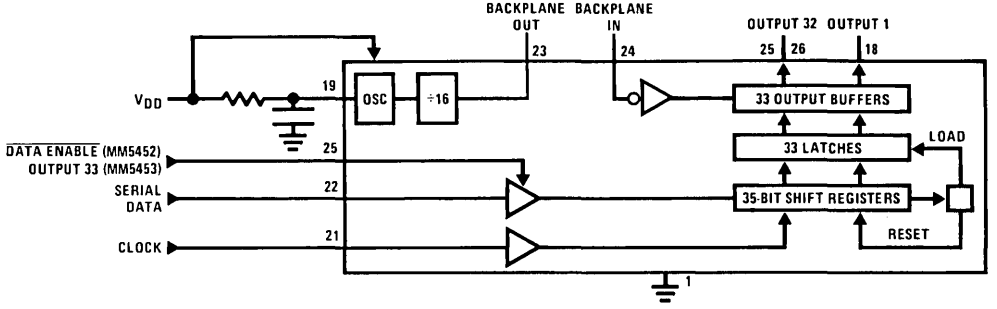
Features

- Serial data input
- No load signal required
- $\overline{\text{DATA ENABLE}}$ (MM5452)
- Wide power supply operation
- TTL compatibility
- 32 or 33 outputs
- Alphanumeric and bar graph capability
- Cascaded operation capability

Applications

- COPS™ or microprocessor displays
- Industrial control indicator
- Digital clock, thermometer, counter, voltmeter
- Instrumentation readouts
- Remote displays

Block Diagram



TL/F/6137–1

FIGURE 1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin	V_{SS} to V_{SS} + 10V
Operating Temperature	0°C to +70°C
Storage Temperature	−65°C to +150°C
Power Dissipation	300 mW at +70°C 350 mW at +25°C
Junction Temperature	+150°C
Lead Temperature (Soldering, 10 sec.)	300°C

Electrical Characteristics

T_A within operating range, V_{DD} = 3.0V to 10V, V_{SS} = 0V, unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
Power Supply		3		10	V
Power Supply Current	Excluding Outputs				
	OSC = V_{SS}, BP IN @ 32 Hz			40	μA
	V_{DD} = 5V, Open Outputs, No Clock			10	μA
Clock Frequency				500	kHz
Input Voltages					
Logical '0' Level	V_{DD} < 4.75	−0.3		0.1 V_{DD}	V
	V_{DD} ≥ 4.75	−0.3		0.8	V
Logical '1' Level	V_{DD} > 5.25	0.8 V_{DD}		V_{DD}	V
	V_{DD} ≤ 5.25	2.0		V_{DD}	V
Output Current Levels					
Segments					
Sink	V_{DD} = 3V, V_{OUT} = 0.3V			−20	μA
Source	V_{DD} = 3V, V_{OUT} = V_{DD} − 0.3V	20			μA
Backplane					
Sink	V_{DD} = 3V, V_{OUT} = 0.3V			−320	μA
Source	V_{DD} = 3V, V_{OUT} = V_{DD} − 0.3V	320			μA
Output Offset Voltage	Segment Load 250 pF Backplane Load 8750 pF (Note 1)			±50	mV
Clock Input Frequency, f_C	(Notes 2 and 3)			500	kHz
High Time, t_h		950			ns
Low Time, t_l		950			ns
Data Input					
Set-Up Time, t_{DS}		300			ns
Hold Time, t_{DH}		300			ns
Data Enable Input					
Set-Up Time, t_{DES}		100			ns

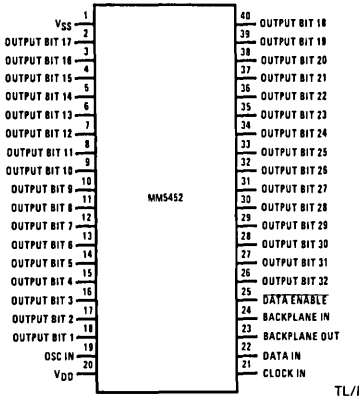
Note 1: This parameter is guaranteed (not 100% production tested) over operating temperature and supply voltage ranges. Not to be used in Q.A. testing.

Note 2: AC input waveform for test purpose: t_r ≤ 20 ns, t_f ≤ 20 ns, f = 500 kHz, 50% ±10% duty cycle.

Note 3: Clock input rise and fall times must not exceed 300 ns.

Connection Diagrams

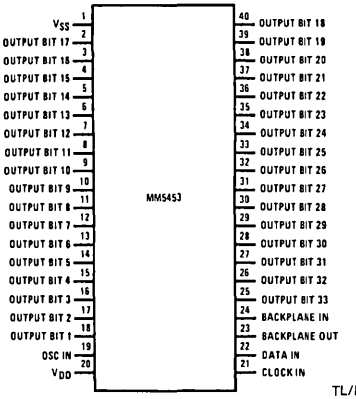
Dual-In-Line Package



TL/F/6137–2

Top View
FIGURE 2a

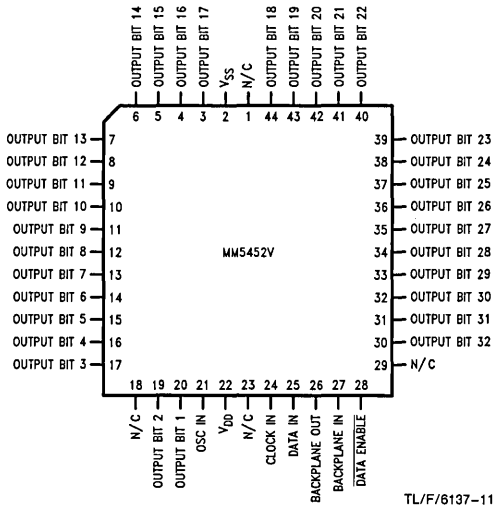
Dual-In-Line Package



TL/F/6137–3

Top View
FIGURE 2b

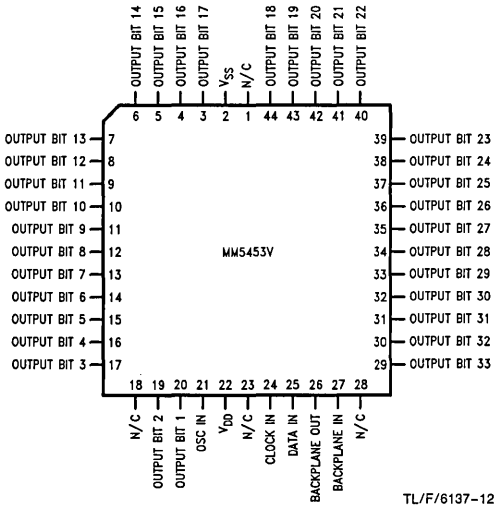
Plastic Chip Carrier



TL/F/6137–11

Top View

Plastic Chip Carrier



TL/F/6137–12

Top View

Order Number MM5452N, MM5453N,
MM5452V or MM5453V
See NS Package Number N40A or V44A

Functional Description

The MM5452 is specifically designed to operate 4 ½-digit 7-segment displays with minimal interface with the display and the data source. Serial data transfer from the data source to the display driver is accomplished with 2 signals, serial data and clock. Since the MM5452 does not contain a character generator, the formatting of the segment information must be done prior to inputting the data to the MM5452. Using a format of a leading "1" followed by the 32 data bits allows data transfer without an additional load signal. The 32 data bits are latched after the 36th clock is complete, thus providing non-multiplexed, direct drive to the display. Outputs change only if the serial data bits differ from the previous time.

A block diagram is shown in *Figure 1.* For the MM5452 a $\overline{\text{DATA ENABLE}}$ is used instead of the 33rd output. If the $\overline{\text{DATA ENABLE}}$ signal is not required, the 33rd output can be brought out. This is the MM5453 device.

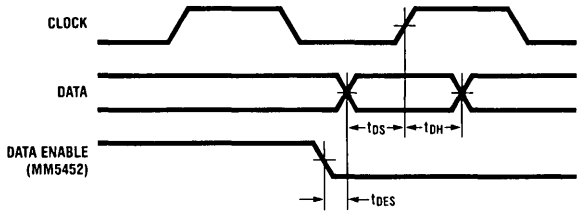
Functional Description (Continued)

Figure 4 shows the input data format. A start bit of logical "1" precedes the 32 bits of data. At the 36th clock a LOAD signal is generated synchronously with the high state of the clock, which loads the 32 bits of the shift registers into the latches. At the low state of the clock a RESET signal is generated which clears all the shift registers for the next set of data. The shift registers are static master-slave configuration. There is no clear for the master portion of the first shift register, thus allowing continuous operation.

If the clock is not continuous, there must be a complete set of 36 clocks otherwise the shift registers will not clear.

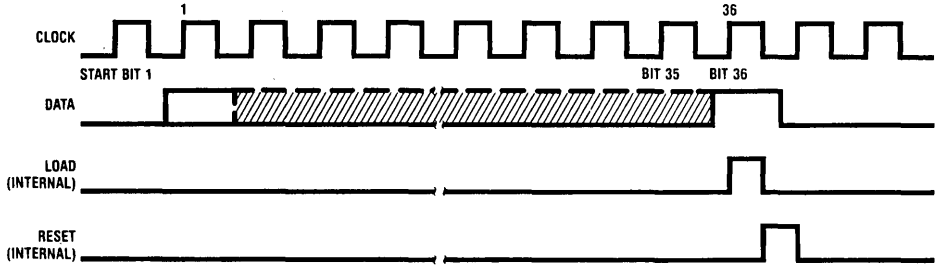
Figure 2a shows the pin-out of the MM5452. Bit 1 is the first bit following the start bit and it will appear on pin 18.

Figure 3 shows the timing relationships between data, clock and $\overline{\text{DATA ENABLE}}$.



TL/F/6137–4

FIGURE 3

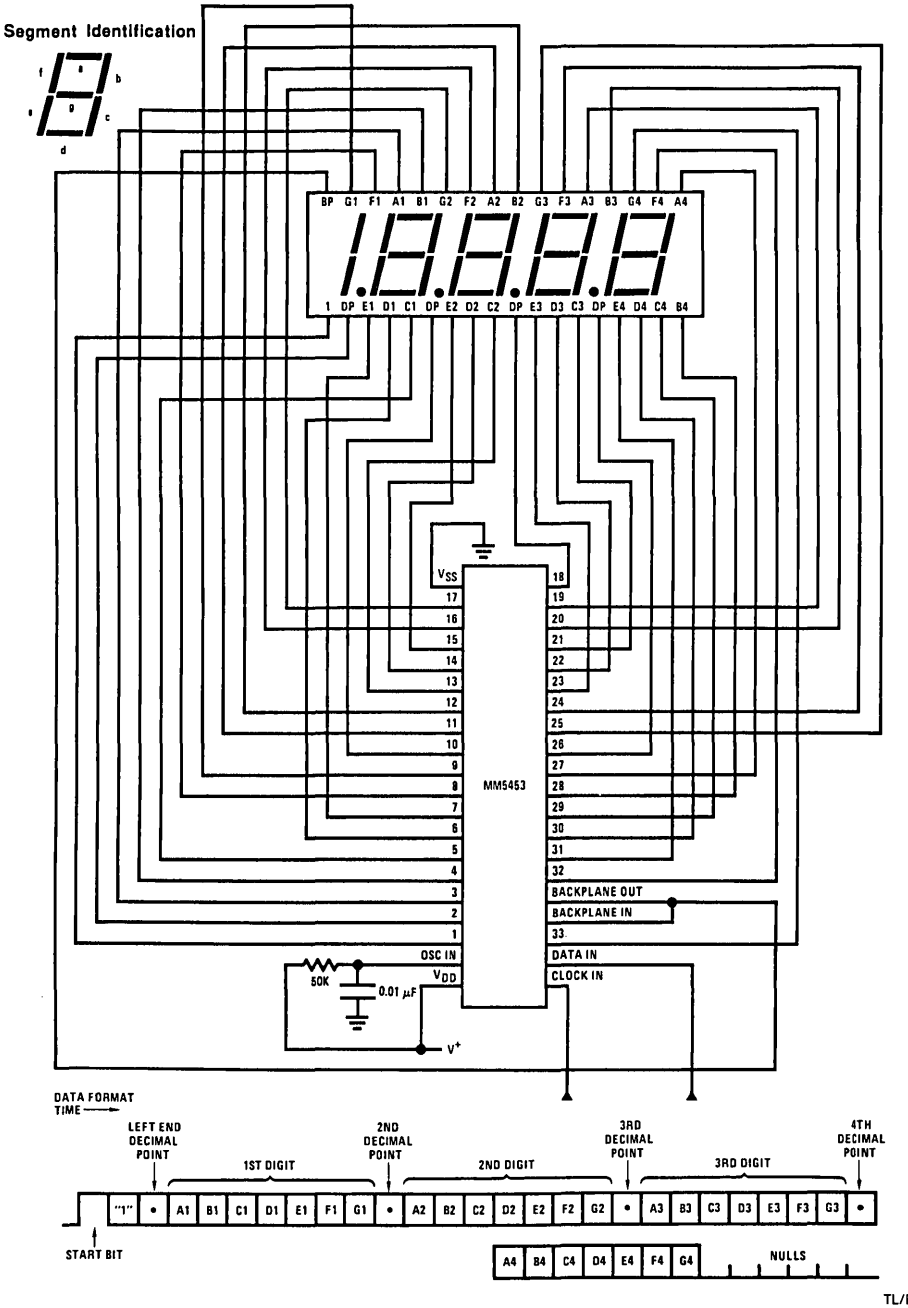


TL/F/6137–5

FIGURE 4. Input Data Format

Functional Description (Continued)

Figure 5 shows a typical application. Note how the input data maps to the output pins and the display. The MM5452 and MM5453 do not have format restrictions, as all outputs are controllable. This application assumes a specific display pinout. Different display/driver connection patterns will, of course, yield a different input data format.

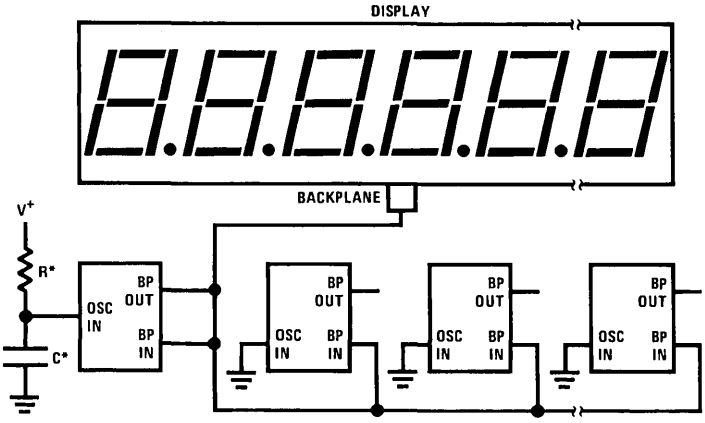


TL/F/6137–6

Consult LCD manufacturer's data sheet for specific pinouts.

FIGURE 5. Typical 4½-Digit Display Application

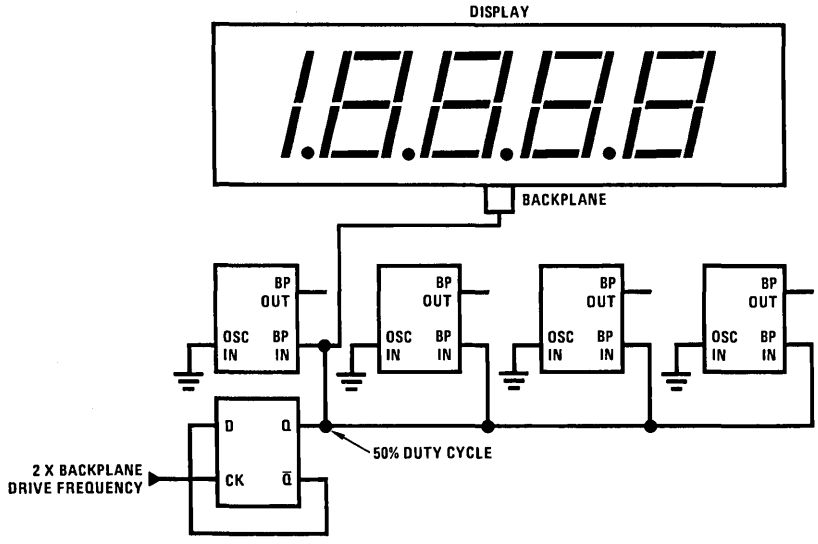
Functional Description (Continued)



TL/F/6137–7

*The minimum recommended value for R for the oscillator input is 9 kΩ. An RC time constant of approximately 4.91×10^{-4} should produce a backplane frequency between 30 Hz and 150 Hz.

FIGURE 6. Parallel Backplane Outputs



TL/F/6137–8

FIGURE 7. External Backplane Clock

Figure 8 shows a four wire remote display that takes advantage of the device's serial input to move many bits of display information on a few wires.

USING AN EXTERNAL CLOCK

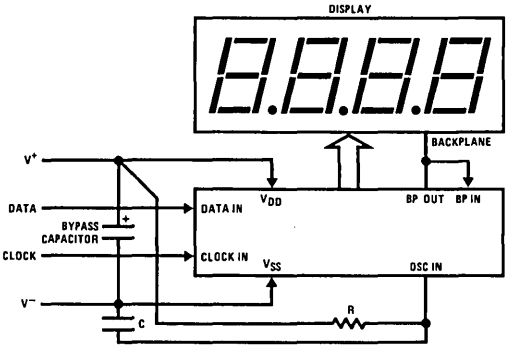
The MM5452/MM5453 LCD Drivers can be used with an externally supplied clock, provided it has a duty cycle of 50%. Deviations from a 50% duty cycle result in an offset voltage on the LCD. In *Figure 7,* a flip-flop is used to assure a 50% duty cycle. The oscillator input is grounded to prevent oscillation and reduce current consumptions in the chips. The oscillator is not used.

Using an external clock allows synchronizing the display drive with AC power, internal clocks, or DVM integration time to reduce interference from the display.

Figure 9 is a general block diagram that shows how the device's serial input can be used to advantage in an analog display. The analog voltage input is compared with a staircase voltage generated by a counter and a digital-to-analog converter or resistor array. The result of this comparison is clocked into the MM5452, MM5453. The next clock pulse increments the staircase and clocks the new data in.

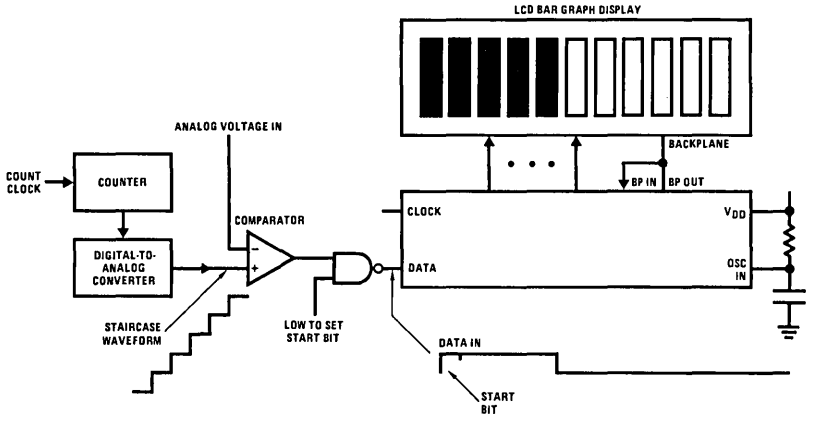
With a buffer amplifier, the same staircase waveform can be used for many displays. The digital-to-analog converter need not be linear; logarithmic or other non-linear functions can be displayed by using weighted resistors or special DACs. This system can be used for status indicators, spectrum analyzers, audio level and power meters, tuning indicators, and other applications.

Functional Description (Continued)



TL/F/6137–9

FIGURE 8. Four Wire Remote Display



TL/F/6137–10

Data is high until staircase > input

FIGURE 9. Analog Display

National Semiconductor Corporation

MM5483 Liquid Crystal Display Driver

General Description

The MM5483 is a monolithic integrated circuit utilizing CMOS metal-gate low-threshold enhancement mode devices. It is available in a 40-pin molded package. The chip can drive up to 31 segments of LCD and can be cascaded to increase this number. This chip is capable of driving a 4½-digit 7-segment display with minimal interface between the display and the data source.

The MM5483 stores the display data in latches after it is latched in, and holds the data until another load pulse is received

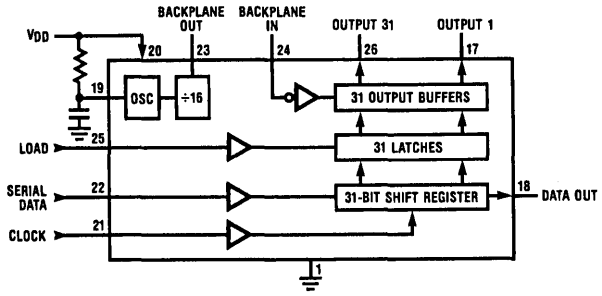
Features

- Serial data input
- Serial data output
- Wide power supply operation
- TTL compatibility
- 31 segment outputs
- Alphanumeric and bar graph capability
- Cascade capability

Applications

- COPS™ or microprocessor displays
- Industrial control indicator
- Digital clock, thermometer, counter, voltmeter
- Instrumentation readouts
- Remote displays

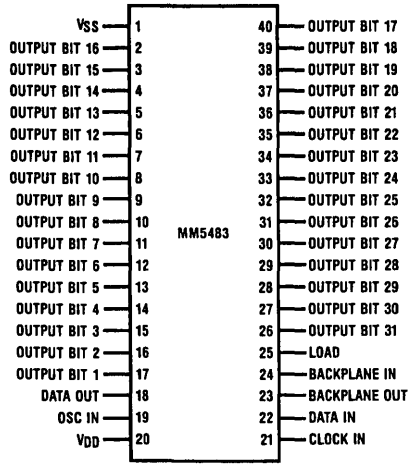
Block and Connection Diagrams



TL/F/6140–1

FIGURE 1

Dual-In-Line Package



Order Number MM5483N
See NS Package Number N40A

TL/F/6140–2

Top View

FIGURE 2

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin	V_{SS} to V_{SS} + 10V
Operating Temperature	−40°C to +85°C
Storage Temperature	−65°C to +150°C
Power Dissipation	300 mW at +85°C 350 mW at +25°C
Junction Temperature	+150°C
Lead Temperature (Soldering, 10 seconds)	300°C

DC Electrical Characteristics

T_A within operating range, V_{DD} = 3.0V to 10V, V_{SS} = 0V, unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
Power Supply		3.0		10	V
Power Supply Current	R = 1M, C = 470 pF, Outputs Open				
	V_{DD} = 3.0V		9	15	μA
	V_{DD} = 5.0V		17	25	μA
	V_{DD} = 10.0V		35	45	μA
	OSC = 0V, Outputs Open, BPIN = 32 Hz, V_{DD} = 3.0V		1.5	2.5	μA
Input Voltage Levels	Load, Clock, Data				
Logic "0"	V_{DD} = 5.0V			0.9	V
Logic "1"	V_{DD} = 5.0V	2.4			V
Logic "0"	V_{DD} = 3.0V			0.4	V
Logic "1"	V_{DD} = 3.0V	2.0			V
Output Current Levels Segments and Data Out					
Sink	V_{DD} = 3.0V, V_{OUT} = 0.3V	20			μA
Source	V_{DD} = 3.0V, V_{OUT} = 2.7V	20			μA
BP OUT					
Sink	V_{DD} = 3.0V, V_{OUT} = 0.3V	320			μA
Source	V_{DD} = 3.0V, V_{OUT} = 2.7V	320			μA

AC Electrical Characteristics $V_{DD} \geq 4.7V$, V_{SS} = 0V unless otherwise specified

Symbol	Parameter		Min	Typ	Max	Units
f_C	Clock Frequency, V_{DD} = 3V				500	kHz
t_{CH}	Clock Period High	(Notes 1, 2)	500			ns
t_{CL}	Clock Period Low		500			ns
t_{DS}	Data Set-Up before Clock		300			ns
t_{DH}	Data Hold Time after Clock		100			ns
t_{LW}	Minimum Load Pulse Width		500			ns
t_{LTC}	Load to Clock		400			ns
t_{CDO}	Clock to Data Valid			400	750	ns

Note 1: AC input waveform specification for test purpose: $t_r \leq 20$ ns, $t_f \leq 20$ ns, f = 500 kHz, 50% ±10% duty cycle.

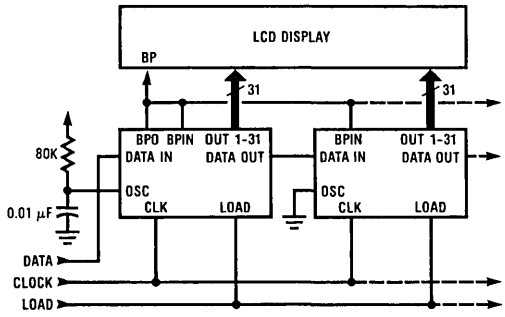
Note 2: Clock input rise and fall times must not exceed 300 ns.

Note 3: Output offset voltage is ±50 mV with $C_{SEGMENT}$ = 250 pF, C_{BP} = 8750 pF.

Functional Description

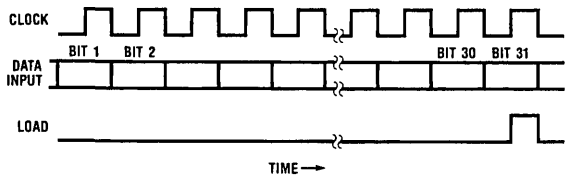
A block diagram for the MM5483 is shown in *Figure 1* and a package pinout is shown in *Figure 2*. *Figure 3* shows a possible 3-wire connection system with a typical signal format for *Figure 3*. Shown in *Figure 4*, the load input is an asynchronous input and lets data through from the shift register to the output buffers any time it is high. The load input can be connected to V_{DD} for 2-wire control as shown in *Figure 5*. In the 2-wire control mode, 31 bits (or less depending on the number of segments used) of data are clocked into the MM5483 in a short time frame (with less than 0.1 second there probably will be no noticeable flicker) with no more clocks until new information is to be displayed. If data was slowly clocked in, it can be seen to "walk" across the display in the 2-wire mode. An AC timing diagram can be seen in *Figure 6*. It should be noted that data out is not a TTL-compatible output.

Functional Description (Continued)



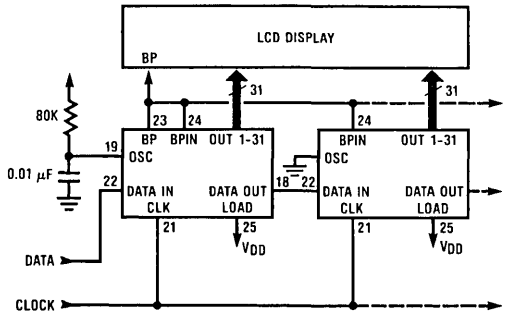
TL/F/6140-3

FIGURE 3. Three-Wire Control Mode



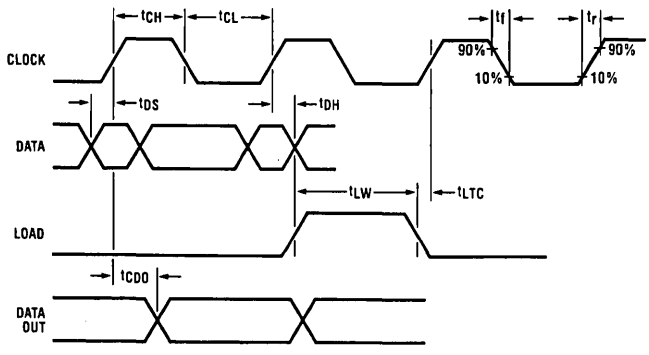
TL/F/6140-4

FIGURE 4. Data Format Diagram



TL/F/6140-5

FIGURE 5. Two-Wire Control Mode



TL/F/6140-6

FIGURE 6. Timing Diagram

National Semiconductor Corporation

MM5484 16-Segment LED Display Driver

General Description

The MM5484 is a low threshold N-channel metal gate circuit using low threshold enhancement and ion implanted depletion devices. The MM5484 is available in a 22-pin molded package and is capable of driving 16 LED segments.

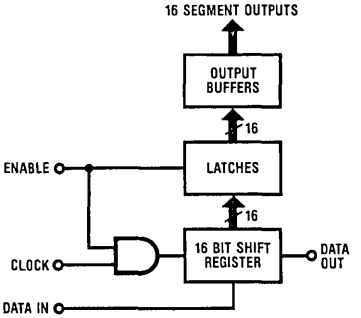
Features

- Serial data input
- Wide power supply operation
- 16 output, 15 mA sink capability
- MM5484 is cascadeable
- TTL compatibility
- No load signal required
- Non multiplex display
- 2½ digit capability—MM5484

Applications

- COPS™ or microprocessor displays
- Instrumentation readouts
- Industrial control indicator
- Relay driver

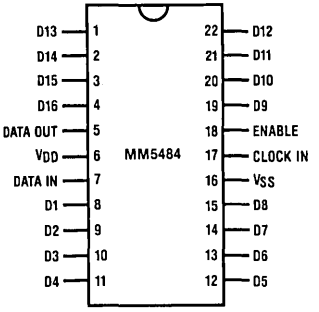
Block and Connection Diagrams



TL/F/6141–1

FIGURE 1. MM5484

Dual-In-Line Package



TL/F/6141–3

Top View

Order Number MM5484N
See NS Package Number N22A

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at LED Outputs	V_{SS} − 0.5V to V_{SS} + 12V
Voltage at Other Pins	V_{SS} − 0.5V to V_{SS} + 10V
Operating Temperature	−40°C to +85°C
Storage Temperature	−40°C to +150°C
Maximum Power Dissipation	
MM5484	500 mW
Lead Temperature (Soldering, 10 sec.)	300°C

DC Electrical Characteristics V_{DD} = 4.5V to 9V, T_A = −40°C to +85°C unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
Supply Voltage		4.5		9	V
Supply Current			5	10	mA
Logic One Input High Level V_{IH}		2.4		V_{DD} + 0.5	V
Logic Zero Input Low Level V_{IL}		0		0.8	V
Input Current	High or Low Level			±1	μA
Input Capacitance				7.5	pF
OUTPUTS					
Data Output Voltage High Level V_{OH}	I_{OUT} = 0.1 mA	V_{DD} − 0.5			V
Low Level V_{OL}	I_{OUT} = −0.1 mA			0.5	V
Segment Off (Logic Zero on Input)	V_{OUT} = 12V, R_{EXT} = 400Ω			50	μA
Output Current Segment On (Logic One on Input) Output Voltage	I_{OUT} = 15 mA, $V_{DD} \geq$ 6V		0.5	1.0	V

AC Electrical Characteristics

(See *Figure 3.*) V_{DD} = 4.5V to 9V, T_A = −40°C to +85°C unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_C	Clock Frequency				0.5	MHz
t_h	High Time		0.95			μs
t_l	Low Time		0.95			μs
t_{S1}	Data Setup Time		0.5			μs
t_{H1}	Data Hold Time		0.5			μs
t_{S2}	Enable Setup Time		0.5			μs
t_{H2}	Enable Hold Time		0.5			μs
t_{pd}	Data Out Delay				0.5	μs

Note 1: Under no condition should the power dissipated by the segment driver exceed 50 mW nor the entire chip power dissipation exceed 500 mW.

Note 2: AC input waveform specification for test purpose: $t_r \leq$ 20 ns, $t_f \leq$ 20 ns, f = 500 kHz, 50% ±10% duty cycle.

Note 3: Clock input rise and fall times must not exceed 500 ns.

Functional Description

The MM5484 is designed to drive LED displays directly. Serial data transfer from the data source to the display driver is accomplished with 3 signals, DATA IN, CLOCK and ENABLE. The signal ENABLE acts as an envelope and only while this signal is at a logic '1' do the circuits recognize the clock signal.

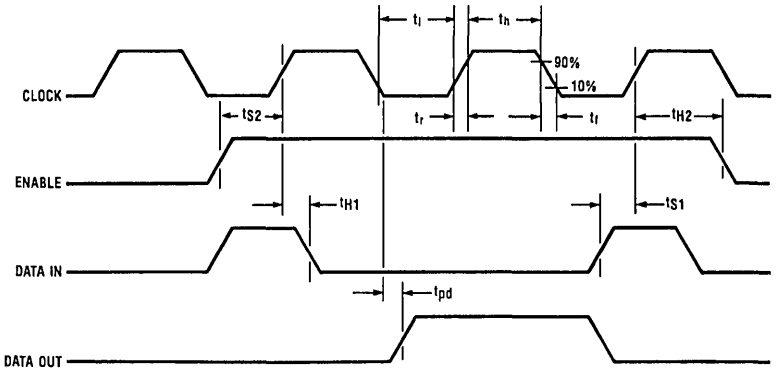
While ENABLE is high, data on the serial data input is transferred and shifted in the internal shift register on the rising clock edge, i.e. a logic '0' to logic '1' transition.

When the ENABLE signal goes to a low (logic zero state), the contents of the shift register is latched and the display will show the new data. While new data is being loaded into the SR the display will continue to show the old data.

For the MM5484, data is output from the serial DATA OUT pin on the falling edge of clock so cascading is made simple with race hazards eliminated.

When the chip first powers on, an internal power on reset signal is generated which resets the SR and latches to zero so that the display will be off.

Timing Diagram



TL/F/6141–5

FIGURE 3

National Semiconductor Corporation

MM5486 LED Display Driver

General Description

The MM5486 is a monolithic MOS integrated circuit utilizing N-channel metal-gate low-threshold, enhancement mode and ion-implanted depletion mode devices. It is available in a 40-pin molded dual-in-line package. A single pin controls the LED display brightness by setting a reference current through a variable resistor connected to V_{DD}.

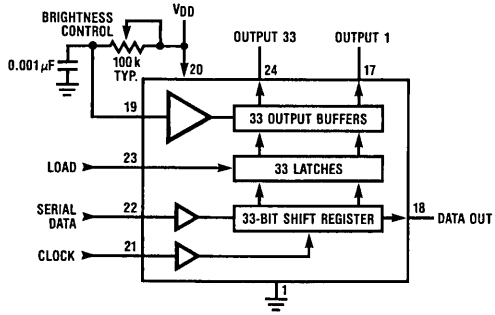
Features

- Continuous brightness control
- Serial data input/outut
- External load input
- Cascaded operation capability
- Wide power supply operation
- TTL compatibility
- 33 outputs, 15 mA sink capability
- Alphanumeric capability

Applications

- COPS™ or microprocessor displays
- Industrial control indicator
- Relay driver
- Digital clock, thermometer, counter, voltmeter
- Instrumentation readouts
- Reference MOS Brief #1

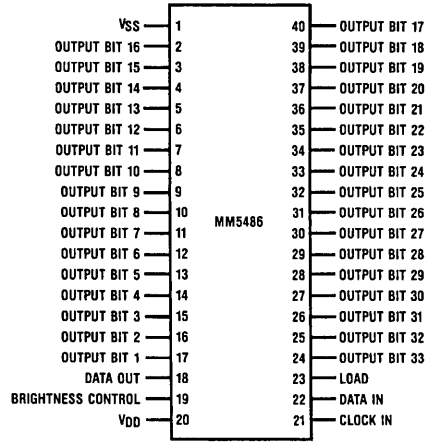
Block and Connection Diagrams



TL/F/6142–1

FIGURE 1

Dual-In-Line Package



Order Number MM5486N
See NS Package Number N40A

TL/F/6142–2

Top View

FIGURE 2

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin	V_{SS} to V_{SS} + 12V
Operating Temperature	−25°C to +85°C
Storage Temperature	−65°C to +150°C
Power Dissipation	560 mW at +85°C 1W at +25°C
Junction Temperature	+150°C
Lead Temperature (Soldering, 10 seconds)	300°C

Electrical Characteristics

T_A within operating range, V_{DD} = 4.75V to 11.0V, V_{SS} = 0V, unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{DD}	Power Supply		4.75		11	V
I_{DD}	Power Supply Current	Excluding Output Loads			7	mA
	Input Voltages					
V_{IL}	Logic "0" Level	±10 μA Input Bias	−0.3		0.8	V
V_{IH}	Logic "1" Level	4.75 ≤ V_{DD} ≤ 5.25	2.2		V_{DD}	V
		V_{DD} > 5.25	V_{DD}−2		V_{DD}	V
I_{BR}	Brightness Input (Note 2)		0		0.75	mA
	Output Sink Current (Note 3)					
I_{OH}	Segment OFF	V_{OUT} = 3.0V			10	μA
I_{OL}	Segment ON	V_{OUT} = 1V (Note 4)				
		Brightness Input = 0 μA	0		10	μA
		Brightness Input = 100 μA	2.0	2.7	4	mA
		Brightness Input = 750 μA	15		25	mA
I_O	Maximum Segment Current				40	mA
V_{IBR}	Brightness Input Voltage (Pin 19)	Input Current = 750 μA	3.0		4.3	V
OM	Output Matching (Note 1)				±20	%
	Data Output					
V_{OL}	Logical "0" Level	I_{OUT} = 0.5 mA	V_{SS}		0.4	V
V_{OH}	Logical "1" Level	I_{OUT} = 100 μA	2.4		V_{DD}	V
	Clock Input	(Notes 5 and 6)				
f_C	Frequency				500	kHz
t_h	High Time		950			ns
t_l	Low Time		950			ns
	Data Input					
t_{DS}	Set-Up Time		300			ns
t_{DH}	Hold Time		300			ns
	Data Enable Input					
t_{DES}	Set-Up Time		100			ns

Note 1: Output matching is calculated as the percent variation $(I_{MAX} + I_{MIN})/2$.

Note 2: With a fixed resistor on the brightness input pin, some variation in brightness will occur from one device to another. Maximum brightness input current can be 2 mA as long as Note 3 and junction temperature equation are complied with.

Note 3: Absolute maximum for each output should be limited to 40 mA.

Note 4: The V_{OUT} voltage should be regulated by the user. See *Figures 6* and *7* for allowable V_{OUT} vs I_{OUT} operation.

Note 5: AC input waveform specification for test purpose: $t_r \leq 20$ ns, $t_f \leq 20$ ns, f = 500 kHz, 50% ±10% duty cycle.

Note 6: Clock input rise and fall times must not exceed 300 ns.

Functional Description

The MM5486 is specifically designed to operate four-digit alphanumeric displays with minimal interface with the display and the data source. Serial data transfer from the data source to the display driver is accomplished with 3 signals, serial data, clock, and load. The data bits are latched by a positive-level load signal, thus providing non-multiplexed, direct drive to the display. When load is high, the data in the shift registers is displayed on the output drivers. Outputs change only if the serial data bits differ from the previous time. Display brightness is determined by control of the output current for LED displays. A 0.001 μF capacitor should be connected to brightness control, pin 19, to prevent possible oscillations. The output current is typically 20 times greater than the current into pin 19, which is set by an external variable resistor. There is an internal limiting resistor of 400Ω nominal value.

A block diagram is shown in *Figure 1*.

Figure 4 shows the input data format. Bit "1" is the first bit into the data input pin and it will appear on pin 17. A logical "1" at the input will turn on the appropriate LED. The load signal latches the 33 bits of the shift register into the latches. The data out pin allows for cascading the shift registers for more than 33 output drivers.

When the chip first powers ON, an internal power ON reset signal is generated which resets all registers and latches. The leading clock returns the chip to its normal operation.

Figure 3 shows the timing relationship between data, clock and data enable. A maximum clock frequency of 0.5 MHz is assumed.

For applications where a lesser number of outputs are used, it is possible to either increase the current per output, or operate the part at higher than 1V V_{OUT}. The following equation can be used for calculations:

$$T_J = (V_{OUT})(I_{LED})(\text{No. of segments})(124°C/W) + T_A$$

where:

T_J = junction temperature +150°C max.

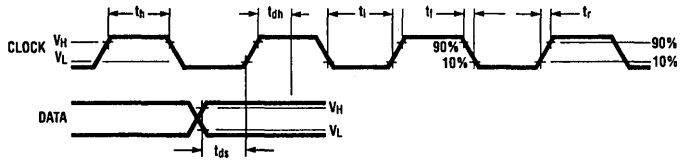
V_{OUT} = the voltage at the LED driver outputs

I_{LED} = the LED current

124°C/W = thermal coefficient of the package

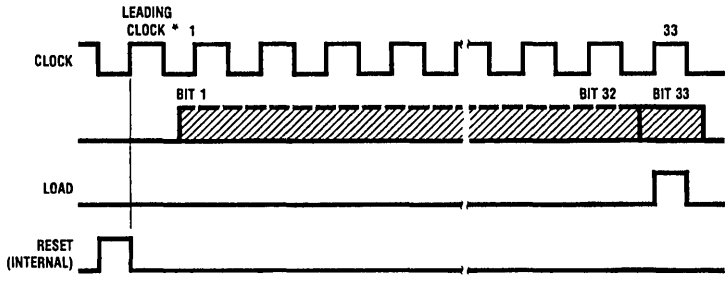
T_A = ambient temperature

The above equation was used to plot *Figure 6, Figure 7,* and *Figure 8*.



TL/F/6142–3

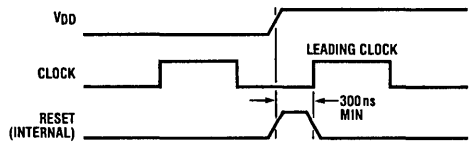
FIGURE 3



TL/F/6142–4

*This leading clock is necessary only after power ON.

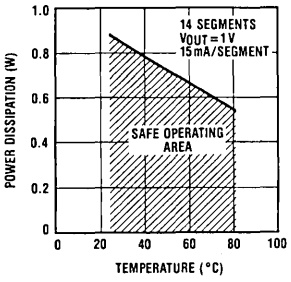
FIGURE 4. Input Data Format



TL/F/6142–5

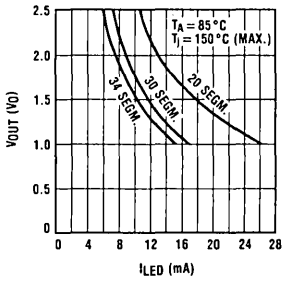
FIGURE 5

Typical Applications



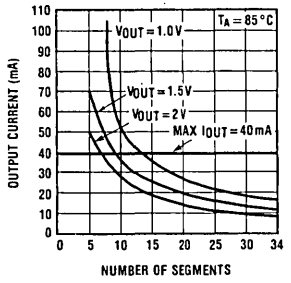
TL/F/6142–6

FIGURE 6



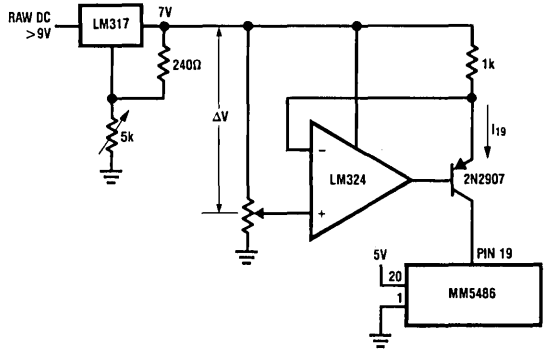
TL/F/6142–7

FIGURE 7



TL/F/6142–8

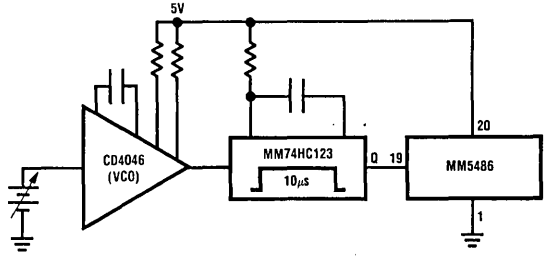
FIGURE 8



TL/F/6142–9

$$I_{19} = \frac{\Delta V}{1k}$$

FIGURE 9. Constant Current Brightness Control

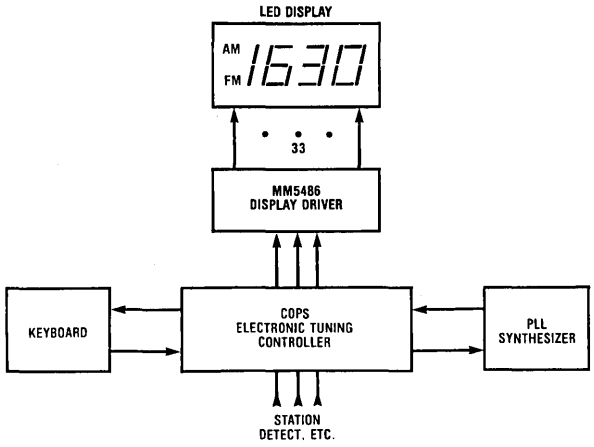


TL/F/6142–10

FIGURE 10. Brightness Control Varying the Duty Cycle

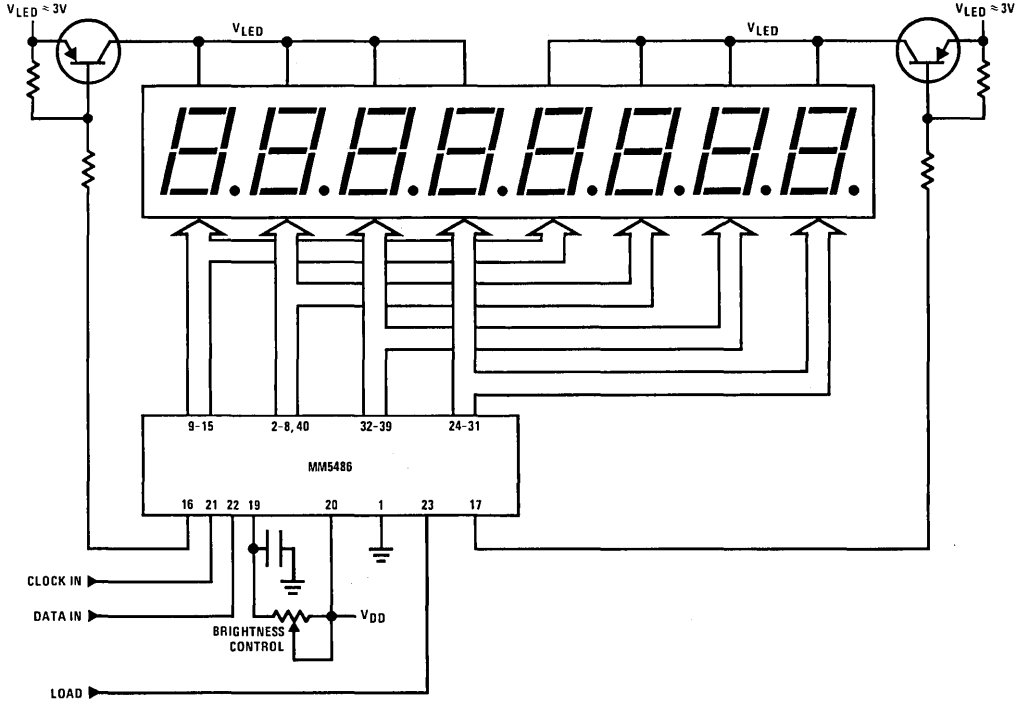
Typical Applications (Continued)

Basic Electronically Tuned Radio System



TL/F/6142–11

Duplexing 8 Digits with One MM5486



TL/F/6142–12

*This driver has 7 segments only.

National Semiconductor Corporation

MM58201 Multiplexed LCD Driver

General Description

The MM58201 is a monolithic CMOS LCD driver capable of driving up to 8 backplanes and 24 segments. A 192-bit RAM stores the data for the display. Serial input and output pins are provided to interface with a controller. An RC oscillator generates the timing necessary to refresh the display. The magnitude of the driving waveforms can be adjusted with the V_{TC} input to optimize display contrast. Four additional bits of RAM allow the user to program the number of backplanes being driven, and to designate the driver as either a master or slave for cascading purposes. When two or more drivers are cascaded, the master chip drives the backplane lines, and the master and each slave chip drive 24 segment lines. Synchronizing the cascaded drivers is accomplished by tying the RC OSC pins together and the BP1 pins together.

The MM58201 is packaged in a 40-lead dual-in-line package, or 44 lead plastic chip carrier package.

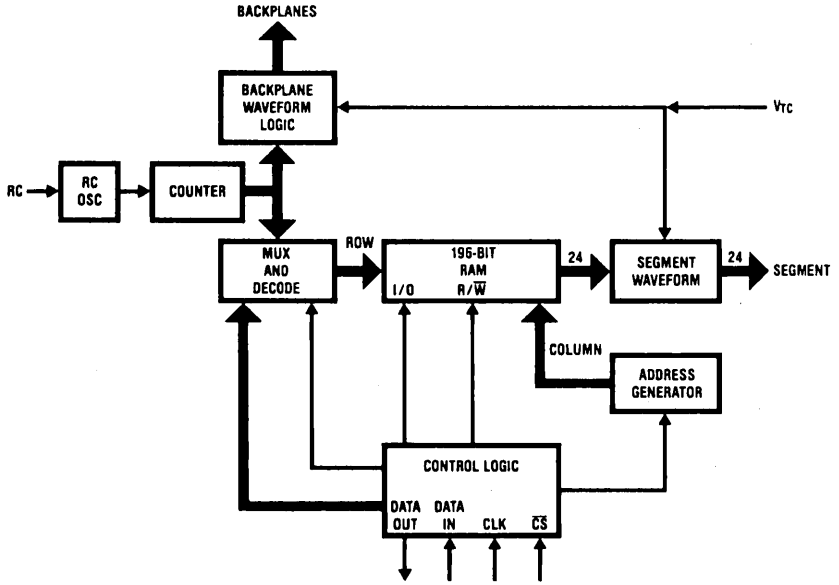
Features

- Drives up to 8 backplanes and 24 segment lines
- Stores data for display
- Cascadable
- Low power
- Fully static operation

Applications

- Dot matrix LCD driver
- Multiplexed 7-segment LCD driver
- Serial in/Serial out memory

Block Diagram



TL/F/6146–1

FIGURE 1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin	V_{SS} −0.3V to V_{SS} +18V
Operating Temperature Range	0°C to 70°C
Storage Temperature Range	−65°C to +150°C
Package Dissipation	500 mW
Operating V_{DD} Range	V_{SS} +7.0V to V_{SS} +18.0V
Lead Temperature (Soldering, 10 seconds)	300°C

DC Electrical Characteristics

Min/max limits apply across temperature range unless otherwise noted.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I_{CC}	Quiescent Supply Current				0.3	mA
$V_{IN(1)}$	Logical "1" Input Voltage		0.45 V_{DD}		V_{DD} +0.3	V
$V_{IN(0)}$	Logical "0" Input Voltage		V_{SS}−0.3		1.0	V
$V_{OUT(0)}$	Logical "0" Output Voltage	I_{SINK} = 0.6 mA			0.4	V
$I_{OUT(1)}$	Logical "1" Output Leakage Current	V_{OUT} = V_{DD}	0		±10	μA
$I_{IN(1)}$	Logical "1" Input Leakage Current	V_{IN} = V_{DD}	0		1.0	μA
$I_{IN(0)}$	Logical "0" Input Leakage Current	V_{IN} = V_{SS}	−1.0		0	μA
V_{TC}	Input Voltage		4.5		V_{DD}+0.3	V
V_{TC}	Input Impedance		10		30	kΩ
Z_{OUT}	Output Impedance	Backplane and Segment Outputs			10	kΩ
Z_{OUT}	DC Offset Voltage	Between Any Backplane and Segment Output	0		±10	mV

AC Electrical Characteristics

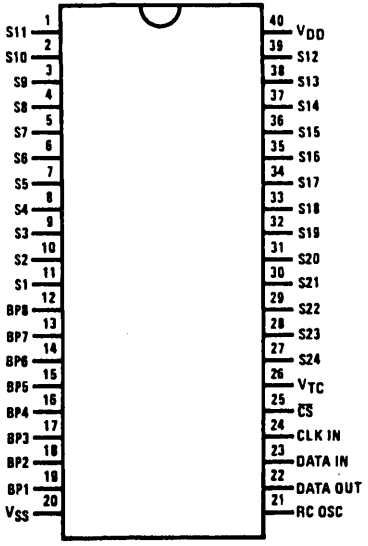
T_A and V_{DD} within operating range unless otherwise noted.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_{OSC}	Oscillator Frequency*		128η		400η	Hz
$f_{CLK\ IN}$	Clock Frequency		DC		100	kHz
t_{ON}	Clock Pulse Width		5.0			μs
t_{OFF}	Clock OFF Time		5.0			μs
t_s	Input Data Set-Up Time		2.0			μs
t_H	Input Data Hold Time		1.0			μs
t_{ACC}	Access Time		5.0			μs
t_r	Rise Time	Backplane, Segment Outputs C_L = 2000 pF			60	μs
t_f	Fall Time	Backplane, Segment Outputs C_L = 2000 pF			60	μs

* η is the number of backplanes programmed.

Connection Diagrams

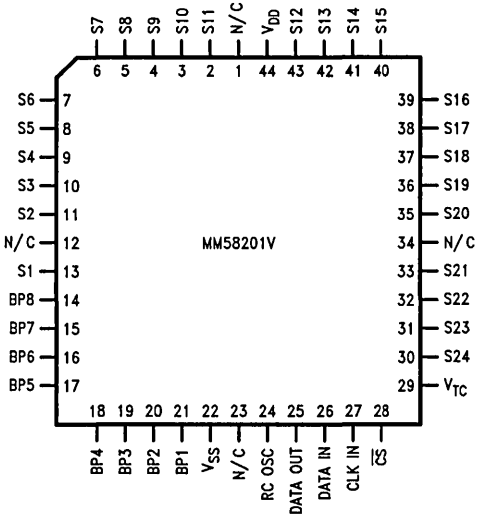
Dual-In-Line Package



TL/F/6146-2

Top View

Plastic Chip Carrier



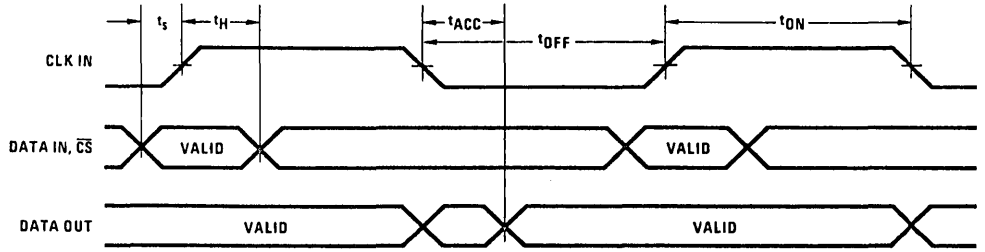
TL/F/6146-10

Top View

FIGURE 2

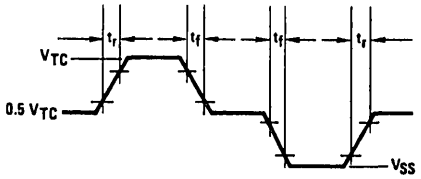
Order Number MM58201N or MM58201V
See NS Package Number N40A or V44A

Switching Time Waveforms



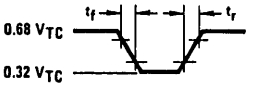
TL/F/6146-3

Backplane Output



TL/F/6146-4

Segment Output



TL/F/6146-5

Functional Description

A functional diagram of the MM58201 LCD driver is shown in *Figure 1*. Connection diagrams are shown in *Figure 2*.

SERIAL INPUTS AND OUTPUTS

A negative-going edge on the $\overline{CS}$ input initiates a frame. The $\overline{CS}$ input must then stay low for at least one rising edge of CLK IN, and may not be pulsed low again for the next 31 clocks. At least one clock must occur while $\overline{CS}$ is high. If CLK IN is held at a logic "1", $\overline{CS}$ is disabled. This allows the signal that drives $\overline{CS}$ to be used for other purposes when the MM58201 is not being addressed.

CLK IN latches data from the DATA IN input on its rising edge. Data from the DATA OUT pin changes on the falling edge of CLK IN and is valid before the next rising edge.

The first five bits of data following $\overline{CS}$ are the address bits *(Figure 3)*. The address selects the column where the operation is to start. Bit 1 is the MSB and bit 5 is the LSB. The sixth bit is the read/write bit. A logic "1" specifies a read operation and a logic "0" specifies a write operation. The next 24 bits are the data bits. The first data bit corresponds to the BP1 row of the display, the second data bit to the BP2 row, and so on. After the eighth and sixteenth data bits, the column pointer is incremented. When starting address 10110 or 10111 is specified, the column pointer increments from 10111 to 00000.

During a read or write cycle, the LCD segment outputs do not reflect the data in the RAM. To avoid disrupting the pattern viewed on the display, the read or write cycle time should be kept short. Since the LCD turn-on time can be as little as 30 ms, a clock rate of at least 10 kHz would be required in order to address the entire contents of the RAM within that time interval. The formula below can be used to estimate the minimum clock rate:

$$f_{CLK\ IN} = \frac{30}{(t_{LCD} - 7t_s)}$$

where t_s is the processor's set-up time between each read or write cycle, and t_{LCD} is the minimum turn-on or turn-off time of the LCD as specified by the LCD manufacturer.

The DATA OUT output is an open drain N-channel device to V_{SS} *(Figure 4)*. With an external pull-up this configuration allows the controller to operate at a lower supply voltage, and also permits the DATA OUT output to be wired in parallel with the DATA OUT outputs from any other drivers in the system.

To program the number of backplanes being driven and the M/$\overline{S}$ bit, load address 11000, a write bit, three bits for the number of backplanes (Table I), and the M/$\overline{S}$ bit. The remaining 20 data bits will be ignored but it is necessary to provide 21 more clocks before initiating another frame.

TABLE I. Backplane Select

Number of Backplanes	B2	B1	B0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

RC OSC Pin

This oscillator generates the timing required for multiplexing the liquid crystal display. The oscillator operates at a frequency that is 4η times the refresh rate of the display, where η is the number of backplanes programmed. Since the refresh rate should be in the range from 32 Hz to 100 Hz, the oscillator frequency must be:

$$128\eta \leq f_{OSC} \leq 400\eta$$

The frequency of oscillation is related to the external R and C components in the following way:

$$f_{OSC} = \frac{1}{1.25\ RC} \pm 30\%$$

The value used for the external resistor should be in the range from 10 kΩ to 1 MΩ.

The value used for the external capacitor should be less than 0.005 μF.

V_{TC} Pin

The V_{TC} pin is an analog input that controls the contrast of the segments on the LCD. If eight backplanes are being driven ($\eta = 8$), a voltage of typically 8V is required at 25°C. The voltage for optimum contrast will vary from display to display. It also has a significant negative temperature coefficient.

The voltage source on the V_{TC} input must be of relatively low impedance since the input impedance of V_{TC} ranges from 10 kΩ to 30 kΩ. A suitable circuit is shown in *Figure 5*.

In a standby mode, the V_{TC} input can be set to V_{SS}. This reduces the supply current to less than 300 μA per driver.

BACKPLANE AND SEGMENT OUTPUTS

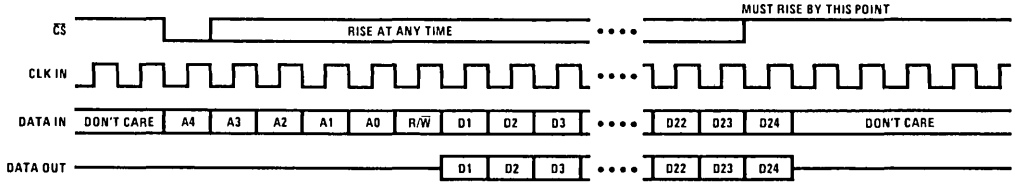
Connect the backplane and segment outputs directly to the LCD row and column lines. The outputs are designed to drive a display with a total ON capacitance of up to 2000 pF.

The output structure consists of transmission gates tapped off of a resistor string driven by V_{TC} *(Figure 6)*.

A critical factor in the lifetime of an LCD is the amount of DC offset between a backplane and segment signal. Typically, 50 mV of offset is acceptable. The MM58201 guarantees an offset of less than 10 mV.

The BP1 output is disabled when the M/$\overline{S}$ bit is set to zero. This allows the BP1 output from the master chip to be connected directly to it so that synchronizing signals can be generated. Synchronization occurs once each refresh cycle, so the cascaded chips are assured of remaining synchronized.

Functional Description (Continued)

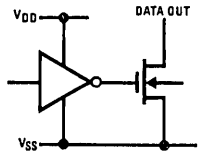


	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22	S23	S24	
BP1													D1	D9	D17										B2
BP2													D2	D10	D18										B1
BP3													D3	D11	D19										B0
BP4													D4	D12	D20										M/S
BP5													D5	D13	D21										
BP6													D6	D14	D22										
BP7													D7	D15	D23										
BP8													D8	D16	D24										
A4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
A3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1
A2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0
A1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
A0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

TL/F/6146–6

Diagram above shows where data will appear on display if starting address 01100 is specified in data format.

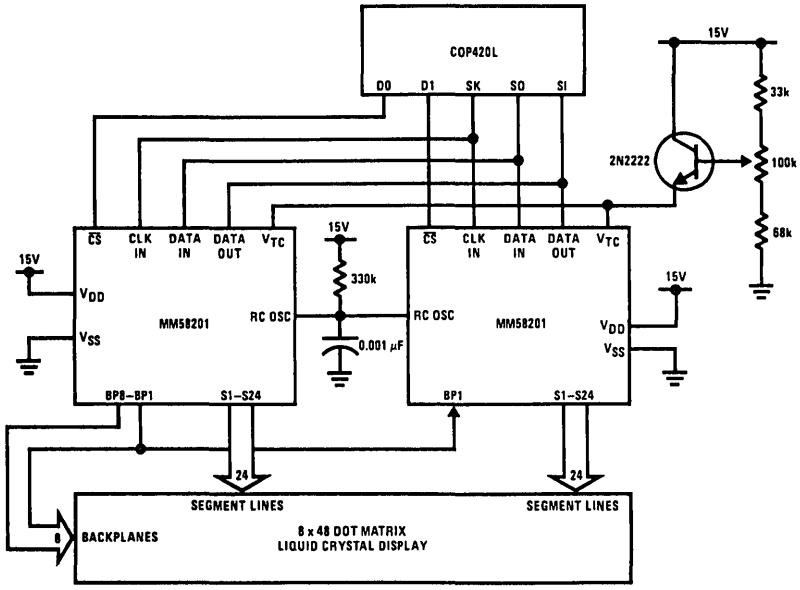
FIGURE 3. Data Format



TL/F/6146–7

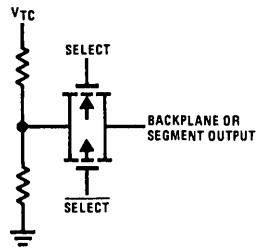
FIGURE 4. DATA OUT Structure

Functional Description (Continued)



TL/F/6146–8

FIGURE 5. Typical Application



TL/F/6146–9

FIGURE 6. Structure of LCD Outputs

National Semiconductor Corporation

MM58248 High Voltage Display Driver

General Description

The MM58248 is a monolithic MOS integrated circuit utilizing CMOS metal gate low threshold P- and N-channel devices. It is available both in 40-pin molded dual-in-line packages or as dice. The MM58248 is particularly suited for driving high voltage (60V max) vacuum fluorescent (VF) displays (e.g., a 5 x 7 dot matrix display).

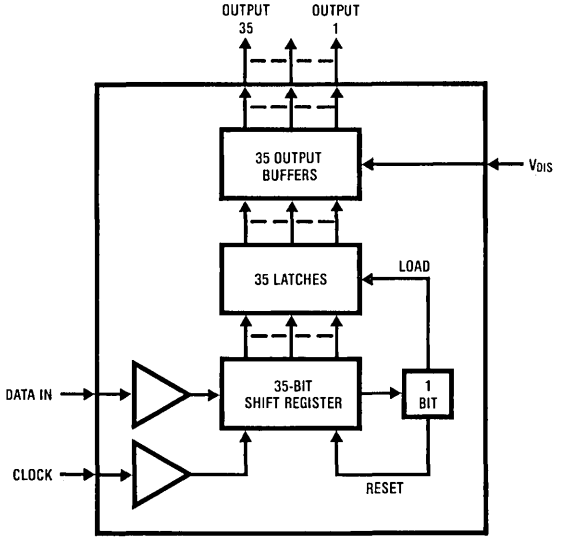
Applications

- COPS™ or microprocessor-driven display
- Instrumentation readouts
- Industrial control indicator
- Digital clock, thermostat, counter, voltmeter
- Word processor text displays
- Automotive dashboards

Features

- Direct interface to high voltage display
- Serial data input
- No external resistors required
- Wide display power supply operation
- LSTTL compatible inputs
- Software compatible with NS display driver family
- Compatible with alphanumeric or dot matrix displays
- No load signal required

Block Diagram



TL/F/5599–1

FIGURE 1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Input Pin	V_{DD} + 0.3V to V_{SS} − 0.3V
Voltage at Any Display Pin	V_{DD} to V_{DD} − 62.5V
$V_{DD} + \lvert V_{DIS} \rvert$	62.5V
Storage Temperature	−65°C to +150°C
Power Dissipation	500 mW at +85°C
Junction Temperature	130°C
Lead Temperature (Soldering, 10 seconds)	260°C

Operating Conditions

	Min	Max	Units
Supply Voltage (V_{DD}) V_{SS} = 0V	4.5	5.5	V
Display Voltage (V_{DIS})	−55	−25	V
Temperature Range	−40	+85	°C

DC Electrical Characteristics

T_A = −40°C to +85°C, V_{DD} = 5V ±0.5V, V_{SS} = 0V unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I_{DD}	Power Supply Currents	$V_{IN} = V_{SS}$ or V_{DD}, V_{SS} = 0V, V_{DIS} Disconnected			150	μA
I_{DIS}		V_{DD} = 5.5V, V_{SS} = 0V, V_{DIS} = −55V, All Outputs Low			10	mA
V_{IL}	Input Logic Levels DATA IN, CLOCK Logic '0'				0.8	V
V_{IH}	Input Logic Levels DATA IN, CLOCK Logic '1'	(Note 1)	2.4			V
I_{IN}	Input Currents, DATA IN, CLOCK	V_{IN} = 0V or V_{DD}	−10		10	μA
C_{IN}	Input Capacitance, DATA IN, CLOCK				15	pF
R_{OFF}	Display Output Impedances Output Off *(Figure 3a)*	V_{DD} = 5.5V, V_{SS} = 0V V_{DIS} = −25V V_{DIS} = −40V V_{DIS} = −55V	 60 70 80		 400 550 650	 kΩ kΩ kΩ
R_{ON}	Display Output Impedances Output on *(Figure 3b)*	V_{DD} = 5.5V, V_{SS} = 0V V_{DIS} = −25V V_{DIS} = −40V V_{DIS} = −55V		 3.0 2.6 2.3	 4.0 3.7 3.4	 kΩ kΩ kΩ
V_{DOL}	Display Output Low Voltage	V_{DD} = 5.5V, I_{OUT} = Open Circuit, −55V ≤ V_{DIS} ≤ −25V	V_{DIS}		V_{DIS} + 4	V

Note 1: 74LSTTL V_{OH} = 2.7V @ I_{OUT} = −400 μA, TTL V_{OH} = 2.4V @ I_{OUT} = −400 μA.

AC Electrical Characteristics T_A = −40°C to +85°C, V_{DD} = 5V ±0.5V

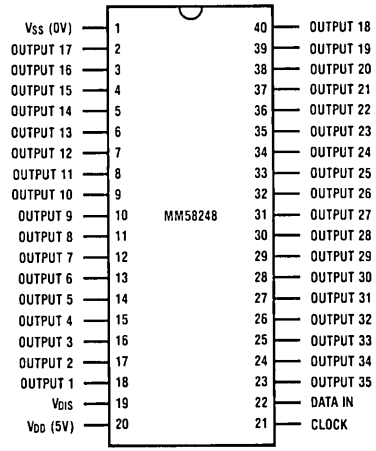
Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_C	Clock Input Frequency	(Notes 2, 3)			1.0	MHz
t_H	Clock Input High Time		300			ns
t_L	Clock Input Low Time		300			ns
t_{DS}	Data Input Setup Time	C_L = 50 pF	100			ns
t_{DH}	Data Input Hold Time		100			ns

Note 2: AC input waveform specification for test purposes: t_r, t_f ≤ 20 ns, f = 1 MHz, 50% ±10% duty cycle.

Note 3: Clock input rise and fall times must not exceed 5 μs.

Connection Diagrams

Dual-In-Line Package

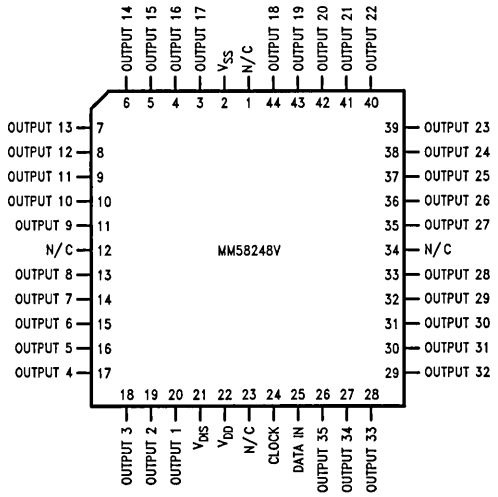


TL/F/5599-2

Top View

Order Number MM58248N
See NS Package Number N40A

Plastic Chip Carrier



TL/F/5599-8

Top View

Order Number MM58248V
See NS Package Number V44A

FIGURE 2

Functional Description

This product is specifically designed to drive multiplexed or non-multiplexed high voltage alphanumeric or dot matrix vacuum fluorescent (VF) displays. Character generation is done externally in the microprocessor, with a serial data path to the display driver. The MM58248 uses two signals, DATA IN and CLOCK, with a format of a leading '1' followed by the 35 data bits, hence allowing data transfer without an additional signal. A block diagram of the MM58248 is shown in *Figure 1.*

Figure 2 shows the pinout of the MM58248 device, where output 1 (pin 18) is equivalent to bit 1, i.e., the first bit of data to be loaded into the shift register following the start bit. A logic '1' at the input will turn on the corresponding display digit/segment/dot output.

A significant reduction in discrete board components can be achieved by the use of the MM58248, because external pull-down resistors are not required. Due to the nature of the output stage, both its on and off impedance values vary as a function of the display voltage applied. However, *Figures 3a* and *3b* show that this output impedance will remain constant for a fixed value of display voltage.

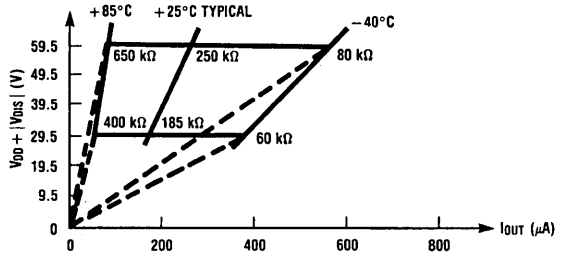
Figure 4 demonstrates the critical timing requirements between CLOCK and DATA IN for the MM58248.

When the chip first powers on, an internal reset is generated, resetting all registers and latches. The chip returns to normal operation on application of the start bit and the first clock pulse, and so all interface signals should be inactive at power on.

In *Figure 5,* a start bit of logic '1' precedes the 35 bits of data, each bit being accepted on the rising edge of CLOCK, i.e., a '0'–'1' transition. At the 36th clock, a LOAD signal is generated synchronously with the high state of the clock, thus loading the 35 bits of the shift register into the latches. At the low state of the clock, a RESET signal is generated, clearing all bits of the shift register for the next set of data. Hence, a complete set of 36 clock pulses is needed for the MM58248, or the shift register will not clear. If, at any given time, it is required that the display be cleared under microprocessor control, i.e., without power on reset, then the following flushing routine may be used. Clock in 36 'zeroes', followed by a 'one' (start bit), followed by 35 'zeroes'. This procedure will completely blank the display.

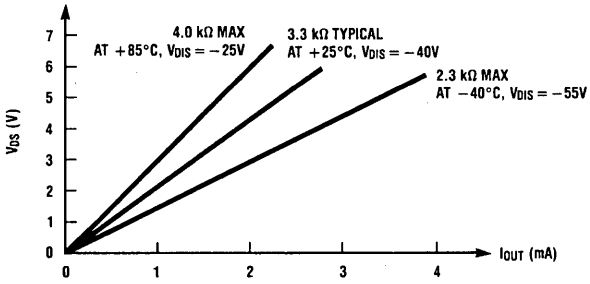
Figure 6 shows a schematic diagram of a microprocessor-based system where the MM58248 is used to provide the anode drive for a 32-digit 5 x 7 dot matrix vacuum fluorescent (VF) display. The grid drive in this example is provided by another member of the high voltage display driver family, namely the MM58241, which has the additional features of a BLANKING CONTROL pin, a DATA OUT pin, and an ENABLE (external load signal) pin.

Functional Description (Continued)



TL/F/5599–3

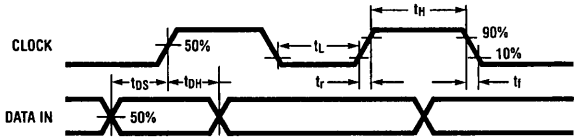
FIGURE 3a. Output Impedance Off



TL/F/5599–4

FIGURE 3b. Output Impedance On

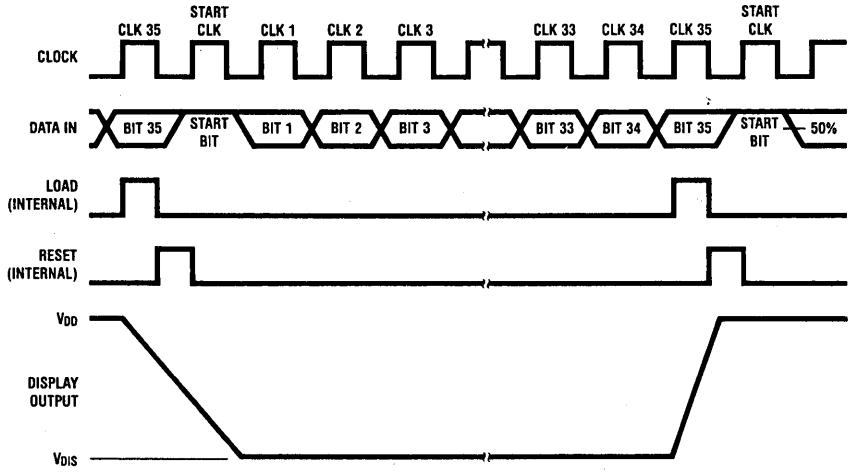
Timing Diagrams



TL/F/5599–5

For the purposes of AC measurement, V_{IH} = 2.4V, V_{IL} = 0.8V.

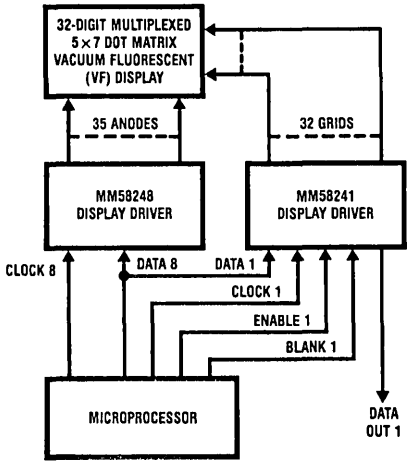
FIGURE 4. Clock and Data Timings



TL/F/5599–6

FIGURE 5. MM58248 Timings (Data Format)

Typical Applications



TL/F/5599–7

FIGURE 6. Microprocessor-Controlled Word Processor

National Semiconductor Corporation

MM58241 High Voltage Display Driver

General Description

The MM58241 is a monolithic MOS integrated circuit utilizing CMOS metal gate low threshold P- and N-channel devices. It is available both in 40-pin molded dual-in-line packages or as dice. The MM58241 is particularly suited for driving high voltage (60V max) vacuum fluorescent (VF) displays (e.g., a 32-digit alphanumeric or dot matrix display).

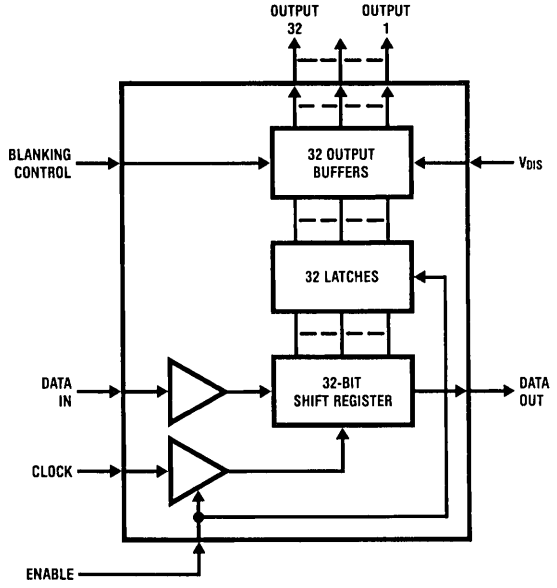
Applications

- COPS™ or microprocessor-driven displays
- Instrumentation readouts
- Industrial control indicator
- Digital clock, thermostat, counter, voltmeter
- Word processor text displays
- Automotive dashboards

Features

- Direct interface to high voltage display
- Serial data input
- No external resistors required
- Wide display power supply operation
- LSTTL compatible inputs
- Software compatible with NS display driver family
- Compatible with alphanumeric or dot matrix displays
- Display blanking control input
- Simple to cascade

Block Diagram



TL/F/5600–1

FIGURE 1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Input Pin	V_{DD} + 0.3V to V_{SS} − 0.3V
Voltage at Any Display Pin	V_{DD} to V_{DD} − 62.5V
V_{DD} + $\lvert V_{DIS} \rvert$	62.5V
Storage Temperature	−65°C to +150°C
Power Dissipation	500 mW at +85°C
Junction Temperature	130°C
Lead Temperature (Soldering, 10 sec.)	260°C

Operating Conditions

	Min	Max	Units
Supply Voltage (V_{DD}) V_{SS} = 0V	4.5	5.5	V
Display Voltage (V_{DIS})	−55	−25	V
Temperature Range	−40	+85	°C

DC Electrical Characteristics

T_A = −40°C to +85°C, V_{DD} = 5V ±0.5V, V_{SS} = 0V unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	Power Supply Currents	V_{IN} = V_{SS} or V_{DD}, V_{SS} = 0V,				
I_{DD}		V_{DIS} Disconnected			150	μA
I_{DIS}		V_{DD} = 5.5V, V_{SS} = 0V, V_{DIS} = −55V All Outputs Low			10	mA
	Input Logic Levels DATA IN, CLOCK ENABLE, BLANK					
V_{IL}	Logic '0'				0.8	V
V_{IH}	Logic '1'	(Note 1)	2.4			V
	Data Output Logic Levels					
V_{OL}	Logic '0'	I_{OUT} = 400 μA			0.4	V
V_{OH}	Logic '1'	I_{OUT} = −10 μA	V_{DD} − 0.5			V
V_{OH}	Logic '1'	I_{OUT} = −500 μA	2.8			V
I_{IN}	Input Currents DATA IN, CLOCK ENABLE, BLANK	V_{IN} = 0V or V_{DD}	−10		10	μA
C_{IN}	Input Capacitance DATA IN, CLOCK ENABLE, BLANK				15	pF
	Display Output Impedances	V_{DD} = 5.5V, V_{SS} = 0V				
R_{OFF}	Output Off *(Figure 3a)*	V_{DIS} = −25V	60		400	kΩ
		V_{DIS} = −40V	70		550	kΩ
		V_{DIS} = −55V	80		650	kΩ
R_{ON}	Output On *(Figure 3b)*	V_{DIS} = −25V		3.0	4.0	kΩ
		V_{DIS} = −40V		2.6	3.7	kΩ
		V_{DIS} = −55V		2.3	3.4	kΩ
V_{DOL}	Display Output Low Voltage	V_{DD} = 5.5V, I_{OUT} = Open Circuit, −55V ≤ V_{DIS} ≤ −25V	V_{DIS}		V_{DIS} + 4	V

Note 1: 74LSTTL V_{OH} = 2.7V @ I_{OUT} = −400 μA, TTL V_{OH} = 2.4V @ I_{OUT} = −400 μA.

AC Electrical Characteristics $T_A = -40°C$ to $+85°C$, $V_{DD} = 5V \pm 0.5V$

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	Clock Input	(Notes 3 and 4)				
f_C	Frequency				800	kHz
t_H	High Time		300			ns
t_L	Low Time		300			ns
	Data Input					
t_{DS}	Set-Up Time		100			ns
t_{DH}	Hold Time		100			ns
	Enable Input					
t_{ES}	Set-Up Time		100			ns
t_{EH}	Hold Time		100			ns
	Data Output	$C_L = 50$ pF				
t_{CDO}	CLOCK Low to Data Out Time				500	ns

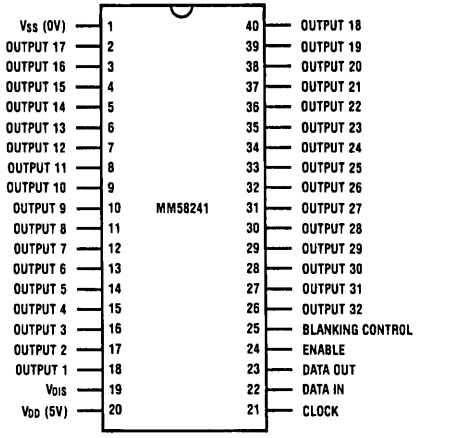
Note 2: For timing purposes, the signals ENABLE and BLANK can be considered to be totally independent of each other.

Note 3: AC input waveform specification for test purposes: t_r, $t_f \leq 20$ ns, f = 800 kHz, 50% ±10% duty cycle.

Note 4: Clock input rise and fall times must not exceed 5 μs.

Connection Diagrams

Dual-In-Line Package



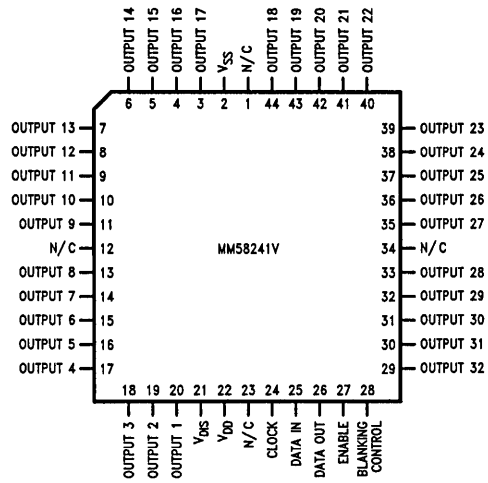
TL/F/5600-2

Top View

FIGURE 2

Order Number MM58241N or MM58241V
See NS Package Number N40A or V44A

Plastic Chip Carrier



TL/F/5600-8

Top View

Functional Description

This product is specifically designed to drive multiplexed or non-multiplexed high voltage alphanumeric or dot matrix vacuum fluorescent (VF) displays. Character generation is done externally in the microprocessor, with a serial data path to the display driver. The MM58421 uses three signals, DATA IN, CLOCK and ENABLE, where ENABLE acts as an external load signal. Display blanking can be achieved by means of the BLANKING CONTROL input, and a logic '1' will turn off all sections of the display. A block diagram of the MM58241 is shown in *Figure 1.*

Figure 2 shows the pinout of the MM58241 device, where output 1 (pin 18) is equivalent to bit 1, i.e., the first bit of data to be loaded into the shift register following ENABLE high. A logic '1' at the input will turn on the corresponding display digit/segment/dot output.

A significant reduction in discrete board components can be achieved by use of the MM58241, because external pull-down resistors are not required. Due to the nature of the output stage, both its on and off impedance values vary as a function of the display voltage applied. However, *Figures 3a* and *3b* show that this output impedance will remain constant for a fixed value of display voltage.

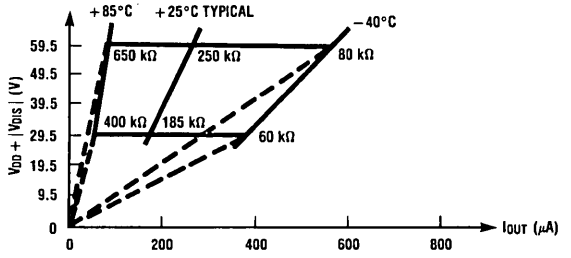
Functional Description (Continued)

Figure 4 demonstrates the critical timing requirements between CLOCK and DATA IN for the MM58241.

When the chip first powers on, an internal reset is generated, resetting all registers and latches. The chip returns to normal operation on application of ENABLE, and so all interface signals should be inactive at power on.

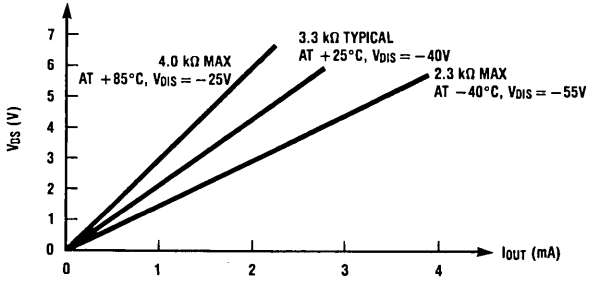
In *Figure 5,* the ENABLE signal acts as an envelope, and only while this signal is at a logic '1' does the circuit accept CLOCK input signals. Data is transferred and shifted in the internal shift register on the rising clock edge, i.e., '0'–'1' transition. When the ENABLE signal goes low, the contents of the shift registers are latched, and the display will show new data. During data transfer, the display will show old data. DATA OUT is also provided on the MM58241, being output on the falling edge. At any time, the display may be blanked under processor control, using the BLANKING CONTROL input.

Figure 6 shows a schematic diagram of a microprocessor-based system where the MM58241 is used to provide the grid drive for a 32-digit 5 x 7 dot matrix vaccum fluorescent (VF) display. The anode drive in this example is provided by another member of the high voltage display driver family, namely the MM58248, which does not require an externally generated load signal.



TL/F/5600–3

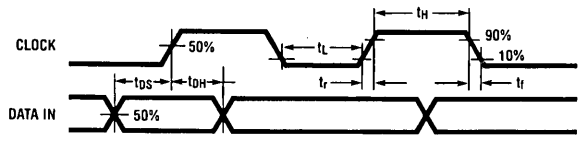
FIGURE 3a. Output Impedance Off



TL/F/5600–4

FIGURE 3b. Output Impedance On

Timing Diagrams

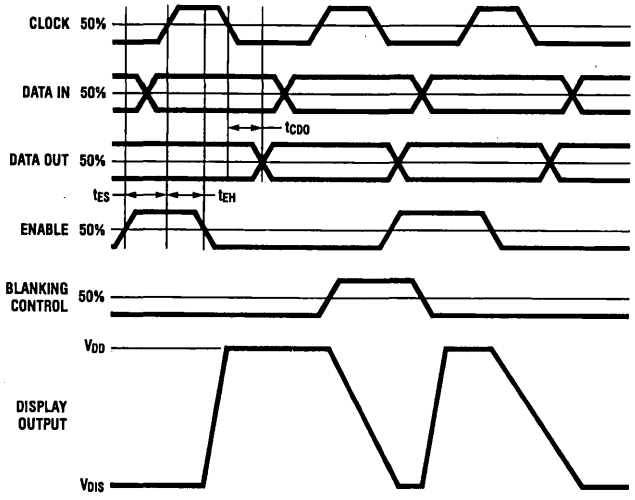


TL/F/5600–5

For the purposes of AC measurements, V_{IH} = 2.4V, V_{IL} = 0.8V.

FIGURE 4. Clock and Data Timings

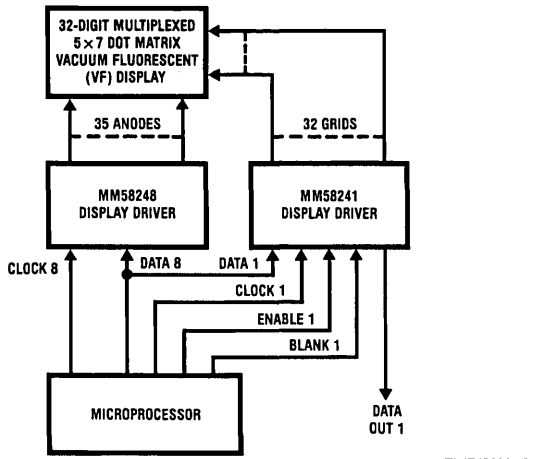
Timing Diagrams (Continued)



TL/F/5600–6

FIGURE 5. MM58241 Timings (Data Format)

Typical Application



TL/F/5600–7

FIGURE 6. Microprocessor-Controlled Word Processor

National Semiconductor Corporation

MM58242 High Voltage Display Driver

General Description

The MM58242 is a monolithic MOS integrated circuit utilizing CMOS metal gate low threshold P and N-channel devices. It is available both in 28-pin molded dual-in-line packages or as dice. The MM58242 is particularly suited for driving high voltage (60V max) vacuum fluorescent (VF) displays (e.g., a 20-digit alphanumeric or dot matrix display).

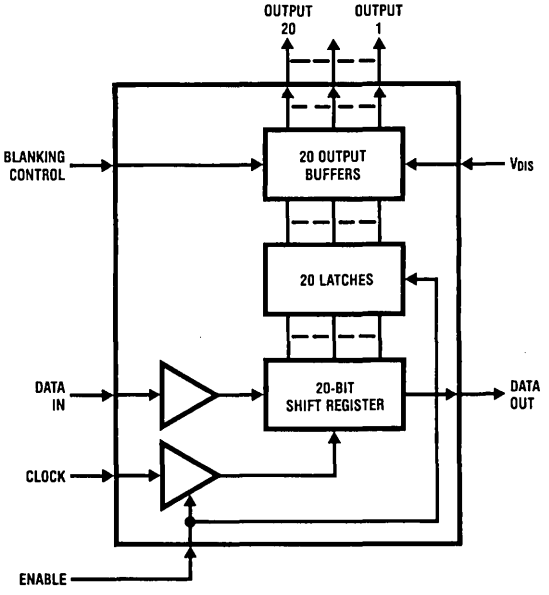
Applications

- COPS™ or microprocessor-driven displays
- Instrumentation readouts
- Industrial control indicator
- Digital clock, thermostat, counter, voltmeter
- Word processor text displays
- Automotive dashboards

Features

- Direct interface to high voltage display
- Serial data input
- No external resistors required
- Wide display power supply operation
- LSTTL compatible inputs
- Software compatible with NS display driver family
- Compatible with alphanumeric or dot matrix displays
- Display blanking control input
- Simple to cascade

Block Diagram



TL/F/7924-1

FIGURE 1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Input Pin	V_{DD} + 0.3V to V_{SS} −0.3V
Voltage at Any Display Pin	V_{DD} to V_{DD} −62.5V
$V_{DD} + \lvert V_{DIS} \rvert$	62.5V
Storage Temperature	−65°C to +150°C
Power Dissipation	500 mW at +85°C
Junction Temperature	130°C
Lead Temperature (Soldering, 10 sec.)	260°C

Operating Conditions

	Min	Max	Units
Supply Voltage (V_{DD}) V_{SS} = 0V	4.5	5.5	V
Display Voltage (V_{DIS})	−55	−25	V
Temperature Range	−40	+85	°C

DC Electrical Characteristics

T_A = −40°C to +85°C, V_{DD} = 5V ±0.5V, V_{SS} = 0V unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I_{DD}	Power Supply Currents	V_{IN} = V_{SS} or V_{DD}, V_{SS} = 0V, V_{DIS} Disconnected			150	μA
I_{DIS}		V_{DD} = 5.5V, V_{SS} = 0V, V_{DIS} = 55V All Outputs Low			10	mA
	Input Logic Levels DATA IN, CLOCK ENABLE, BLANK					
V_{IL}	Logic '0'				0.8	V
V_{IH}	Logic '1'	(Note 1)	2.4			V
	Data Output Logic Levels					
V_{OL}	Logic '0'	I_{OUT} = 400 μA			0.4	V
V_{OH}	Logic '1'	I_{OUT} = −10 μA	V_{DD}−0.5			V
V_{OH}	Logic '1'	I_{OUT} = −500 μA	2.8			V
I_{IN}	Input Currents DATA IN, CLOCK ENABLE, BLANK	V_{IN} = 0V or V_{DD}	−10		10	μA
C_{IN}	Input Capacitance DATA IN, CLOCK ENABLE, BLANK				15	pF
	Display Output Impedances	V_{DD} = 5.5V, V_{SS} = 0V				
R_{OFF}	Output Off *(Figure 3a)*	V_{DIS} = −25V	60		400	kΩ
		V_{DIS} = −40V	70		550	kΩ
		V_{DIS} = −55V	80		650	kΩ
R_{ON}	Output On *(Figure 3b)*	V_{DIS} = −25V		3.0	4.0	kΩ
		V_{DIS} = 40V		2.6	3.7	kΩ
		V_{DIS} = −55V		2.3	3.4	kΩ
V_{DOL}	Display Output Low Voltage	V_{DD} = 5.5V, I_{OUT} = Open Circuit, −55V ≤ V_{DIS} ≤ −25V	V_{DIS}		V_{DIS}+4	V

Note 1: 74LSTTL V_{OH} = 2.7V @ I_{OUT} = −400 μA, TTL V_{OH} = 2.4V @ I_{OUT} = −400 μA.

AC Electrical Characteristics $T_A = -40°C$ to $+85°C$, $V_{DD} = 5V \pm 0.5V$

Parameter	Conditions	Min	Typ	Max	Units
Clock Input	(Notes 3 and 4)				
Frequency, f_C				800	kHz
High Time, t_H		300			ns
Low Time, t_L		300			ns
Data Input					
Set-Up Time, t_{DS}		100			ns
Hold Time, t_{DH}		100			ns
Enable Input	(Note 2)				
Set-Up Time, t_{ES}		100			ns
Hold Time, t_{EH}		100			ns
Data Output	$C_L = 50$ pF				
CLOCK Low to Data Out Time, t_{CDO}				500	ns

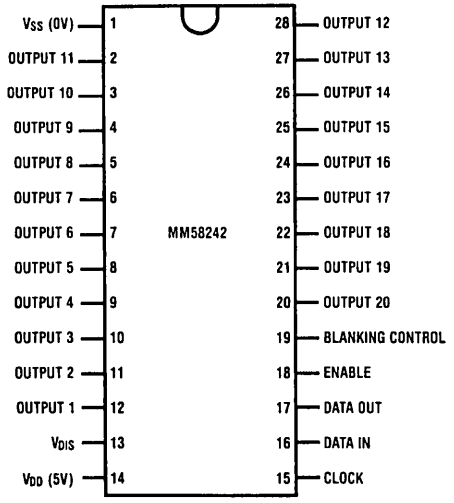
Note 2: For timimg purposes, the signals ENABLE and BLANK can be considered to be totally independent of each other.

Note 3: AC input waveform specification for test purposes: $t_r \leq 20$ ns, $t_f \leq 20$ ns, f = 800 kHz, 50% ±10% duty cycle.

Note 4: Clock input rise and fall times must not exceed 5 μs.

Connection Diagrams

Dual-In-Line Package

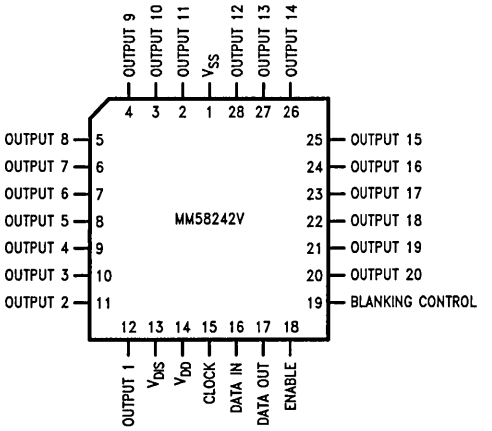


TL/F/7924-2

Top View
FIGURE 2

Order Number MM58242N
See NS Package Number N28B

Plastic Chip Carrier



TL/F/7924-8

Top View
Order Number MM58242V
See NS Package Number V28A

Functional Description

This product is specifically designed to drive multiplexed or non-multiplexed high voltage alphanumeric or dot matrix vacuum fluorescent (VF) displays. Character generation is done externally in the microprocessor, with a serial data path to the display driver. The MM58242 uses three signals, DATA IN, CLOCK and ENABLE, where ENABLE acts as an external load signal. Display blanking can be achieved by means of the BLANKING CONTROL input, and a logic '1' will turn off all sections of the display. A block diagram of the MM58242 is shown in *Figure 1*.

Figure 2 shows the pinout of the MM58242 device, where output 1 (pin 12) is equivalent to bit 1 (i.e., the first bit of data to be loaded into the shift register following ENABLE high). A logic '1' at the input will turn on the corresponding display digit/segment/dot output.

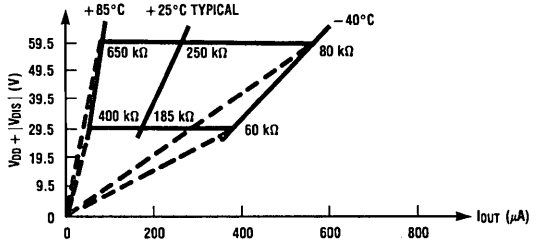
A significant reduction in discrete board components can be achieved by use of the MM58242, because external pull-down resistors are not required. Due to the nature of the output stage, both its on and off impedance values vary as a function of the display voltage applied, However, *Figures 3a* and *3b* show that this output impedance will remain constant for a fixed value of display voltage.

Figure 4 demonstrates the critical timing requirements between CLOCK and DATA IN for the MM58242.

When the chip first powers on, an internal reset is generated, resetting all registers and latches. The chip returns to normal operation on application of ENABLE, and so all interface signals should be inactive at power on.

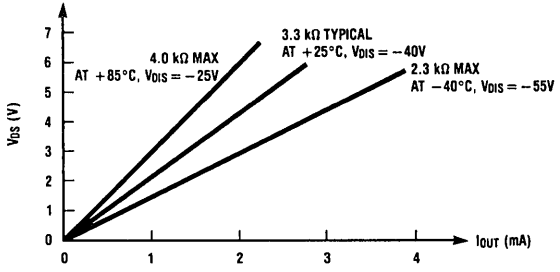
In *Figure 5*, the ENABLE signal acts as an envelope, and only while this signal is at a logic '1' does the circuit accept CLOCK input signals. Data is transferred and shifted in the internal shift register on the rising clock edge, i.e., '0'–'1' transition. When the ENABLE signal goes low, the contents of the shift registers are latched, and the display will show new data. During data transfer, the display will show old data. DATA OUT is also provided on the MM58242 being output on the falling edge. At any time, the display may be blanked under processor control, using the BLANKING CONTROL input.

Figure 6 shows a schematic diagram of a microprocessor-based system where the MM58242 is used to provide the grid drive for a 40-digit 2 line 5 × 7 multiplexed vacuum fluorescent (VF) display. The anode drive in this example is provided by another member of the high voltage display driver family, namely the MM58248, which does not require an externally generated load signal.



TL/F/7924-3

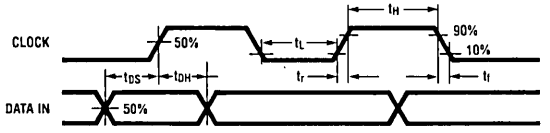
FIGURE 3a. Output Impedance Off



TL/F/7924-4

FIGURE 3b. Output Impedance On

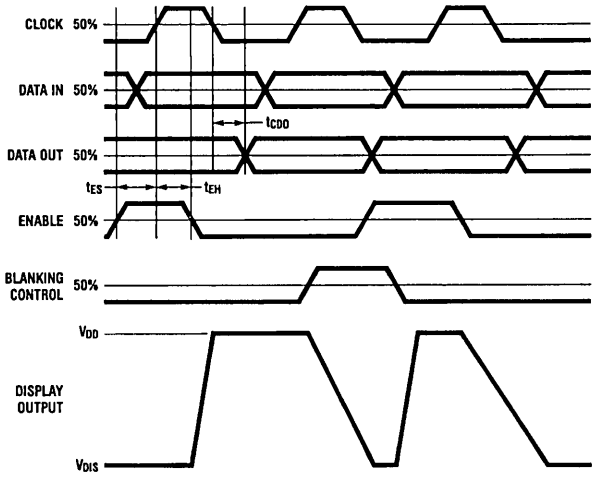
Timing Diagrams



TL/F/7924–5

For the purposes of AC measurement, V_{IH} = 2.4V, V_{IL} = 0.8V.

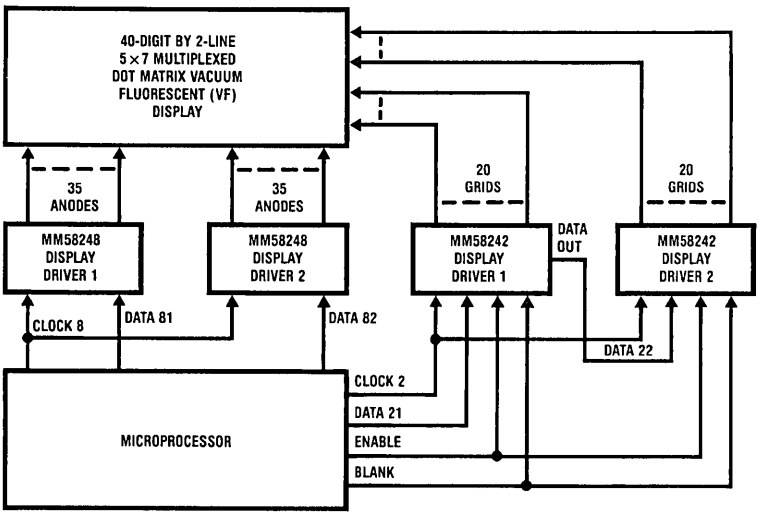
FIGURE 4. Clock and Data Timings



TL/F/7924–6

FIGURE 5. MM58242 Timings (Data Format)

Typical Application



TL/F/7924–7

FIGURE 6. Microprocessor-Controlled Word Processor

National Semiconductor Corporation

MM58342 High Voltage Display Driver

General Description

The MM58342 is a monolithic MOS integrated circuit utilizing CMOS metal gate low threshold P- and N-channel devices. It is available both in 28-pin molded dual-in-line packages or as dice. The MM58342 is particularly suited for driving high voltage (35V max) vacuum fluorescent (VF) displays (e.g., a 20-digit alphanumeric or dot matrix display).

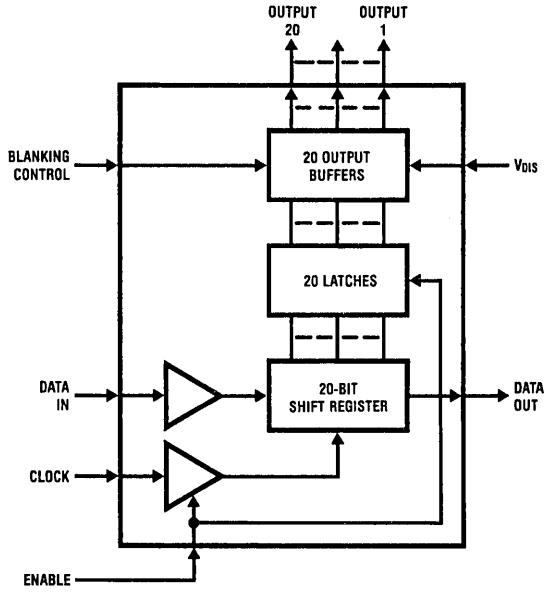
Applications

- COPS™ or microprocessor-driven displays
- Instrumentation readouts
- Industrial control indicator
- Digital clock, thermostat, counter, voltmeter
- Word processor text displays
- Automotive dashboards

Features

- Direct interface to high voltage display
- Serial data input
- No external resistors required
- Wide display power supply operation
- LSTTL compatible inputs
- Software compatible with NS display driver family
- Compatible with alphanumeric or dot matrix displays
- Display blanking control input
- Simple to cascade

Block Diagram



TL/F/7925–1

FIGURE 1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Input Pin	V_{DD} + 0.3V to V_{SS} − 0.3V
Voltage at Any Display Pin	V_{DD} to V_{DD} − 36.5V
V_{DD} + $\|V_{DIS}\|$	36.5V
Storage Temperature	−65°C to +150°C
Power Dissipation	500 mW at +85°C
Junction Temperature	130°C
Lead Temperature (Soldering, 10 sec.)	260°C

Operating Conditions

	Min	Max	Units
Supply Voltage (V_{DD}) V_{SS} = 0V	4.5	5.5	V
Display Voltage (V_{DIS})	−30	−10	V
Temperature Range	−40	+85	°C

DC Electrical Characteristics

T_A = −40°C to +85°C, V_{DD} = 5V ±0.5V, V_{SS} = 0V unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	Power Supply Currents					
I_{DD}		V_{IN} = V_{SS} or V_{DD}, V_{SS} = 0V, V_{DIS} Disconnected			150	μA
I_{DIS}		V_{DD} = 5.5V, V_{SS} = 0V, V_{DIS} = −30V All Outputs Low			10	mA
	Input Logic Levels DATA IN, CLOCK ENABLE, BLANK					
V_{IL}	Logic '0'				0.8	V
V_{IH}	Logic '1'	(Note 1)	2.4			V
	Data Output Logic Levels					
V_{OL}	Logic '0'	I_{OUT} = 400 μA			0.4	V
V_{OH}	Logic '1'	I_{OUT} = −10 μA	V_{DD} − 0.5			V
V_{OH}	Logic '1'	I_{OUT} = −500 μA	2.8			V
I_{IN}	Input Currents DATA IN, CLOCK ENABLE, BLANK	V_{IN} = 0V or V_{DD}	−10		10	μA
C_{IN}	Input Capacitance DATA IN, CLOCK ENABLE, BLANK				15	pF
	Display Output Impedances	V_{DD} = 5.5V, V_{SS} = 0V				
R_{OFF}	Output Off *(Figure 3a)*	V_{DIS} = −10V	55		250	kΩ
		V_{DIS} = −20V	60		300	kΩ
		V_{DIS} = −30V	65		400	kΩ
R_{ON}	Output On *(Figure 3b)*	V_{DIS} = −10V		700	800	Ω
		V_{DIS} = −20V		600	750	Ω
		V_{DIS} = −30V		500	680	Ω
V_{DOL}	Display Output Low Voltage	V_{DD} = 5.5V, I_{OUT} = Open Circuit, −30V ≤ V_{DIS} ≤ −10V	V_{DIS}		V_{DIS} + 2	V

Note 1: 74LSTTL V_{OH} = 2.7V @ I_{OUT} = −400 μA, TTL V_{OH} = 2.4V @ I_{OUT} = −400 μA.

AC Electrical Characteristics $T_A = -40°C$ to $+85°C$, $V_{DD} = 5V \pm 0.5V$

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	Clock Input	(Notes 3 and 4)				
f_C	Frequency				800	kHz
t_H	High Time		300			ns
t_L	Low Time		300			ns
	Data Input					
t_{DS}	Set-Up Time		100			ns
t_{DH}	Hold Time		100			ns
	Enable Input	(Note 2)				
t_{ES}	Set-Up Time		100			ns
t_{EH}	Hold Time		100			ns
	Data Output	$C_L = 50$ pF				
t_{CDO}	CLOCK Low to Data Out Time				500	ns

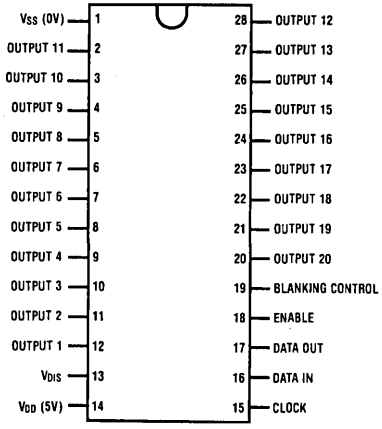
Note 2: For timing purposes, the signals ENABLE and BLANK can be considered to be totally independent of each other.

Note 3: AC input waveform specification for test purposes: t_r, $t_f \leq 20$ ns, f = 800 kHz, 50% ±10% duty cycle.

Note 4: Clock input rise and fall times must not exceed 5 μs.

Connection Diagrams

Dual-In-Line Package



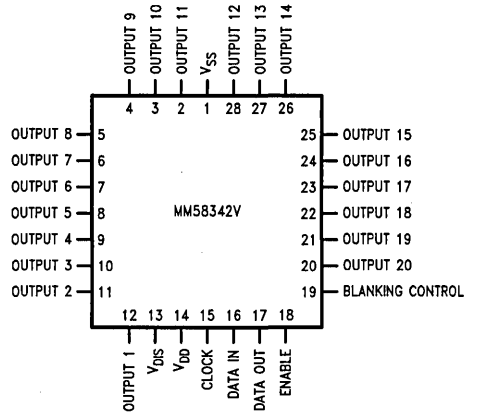
TL/F/7925-2

Top View

FIGURE 2

Order Number MM58342N
See NS Package Number N28B

Plastic Chip Carrier



TL/F/7925-8

Top View
Order Number MM58342V
See NS Package Number V28A

Functional Description

This product is specifically designed to drive multiplexed or non-multiplexed high voltage alphanumeric or dot matrix vacuum fluorescent (VF) displays. Character generation is done externally in the microprocessor, with a serial data path to the display driver. The MM58342 uses three signals, DATA IN, CLOCK and ENABLE, where ENABLE acts as an external load signal. Display blanking can be achieved by means of the BLANKING CONTROL input, and a logic '1' will turn off all sections of the display. A block diagram of the MM58342 is shown in *Figure 1*.

Figure 2 shows the pinout of the MM58342 device, where output 1 (pin 12) is equivalent to bit 1 (i.e., the first bit of data to be loaded into the shift register following ENABLE high). A logic '1' at the input will turn on the corresponding display digit/segment/dot output.

A significant reduction in discrete board components can be achieved by use of the MM58342, because external pull-down resistors are not required. Due to the nature of the output stage, both its on and off impedance values vary as a function of the display voltage applied. However, *Figures 3a*

Functional Description (Continued)

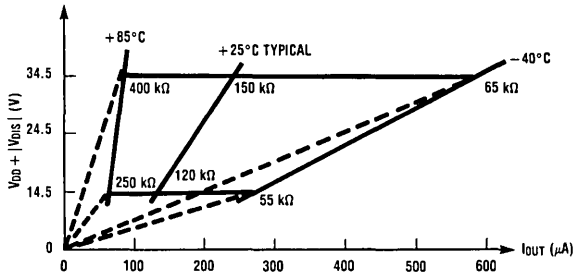
and *3b* show that this output impedance will remain constant for a fixed value of display voltage.

Figure 4 demonstrates the critical timing requirements between CLOCK and DATA IN for the MM58342.

When the chip first powers on, an internal reset is generated, resetting all registers and latches. The chip returns to normal operation on application of ENABLE, and so all interface signals should be inactive at power on.

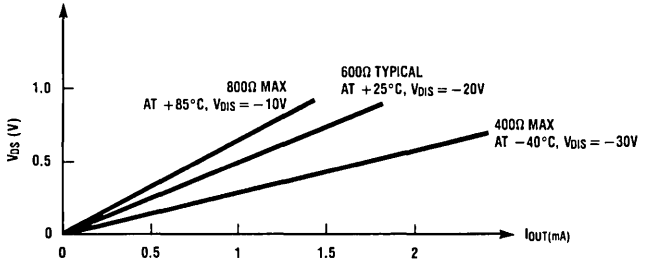
In *Figure 5*, the ENABLE signal acts as an envelope, and only while this signal is at a logic '1' does the circuit accept CLOCK input signals. Data is transferred and shifted in the internal shift register on the rising clock edge, i.e., '0'–'1' transition. When the ENABLE signal goes low, the contents of the shift registers are latched, and the display will show new data. During data transfer, the display will show old data. DATA OUT is also provided on the MM58342 being output on the falling edge. At any time, the display may be blanked under processor control, using the BLANKING CONTROL input.

Figure 6 shows a schematic diagram of a microprocessor-based system where the MM58342 is used to provide the grid drive for a 40-digit 2 line 5 x 7 multiplexed vacuum fluorescent (VF) display. The anode drive in this example is provided by another member of the high voltage display driver family, namely the MM58348, which does not require an externally generated load signal.



TL/F/7925-3

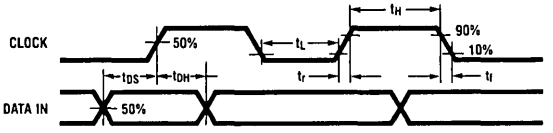
FIGURE 3a. Output Impedance Off



TL/F/7925-4

FIGURE 3b. Output Impedance On

Timing Diagrams

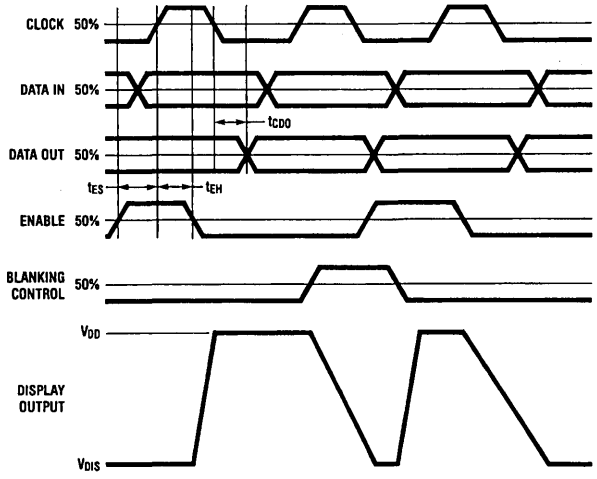


TL/F/7925-5

For the purposes of AC measurement, V_{IH} = 2.4V, V_{IL} = 0.8V.

FIGURE 4. Clock and Data Timings

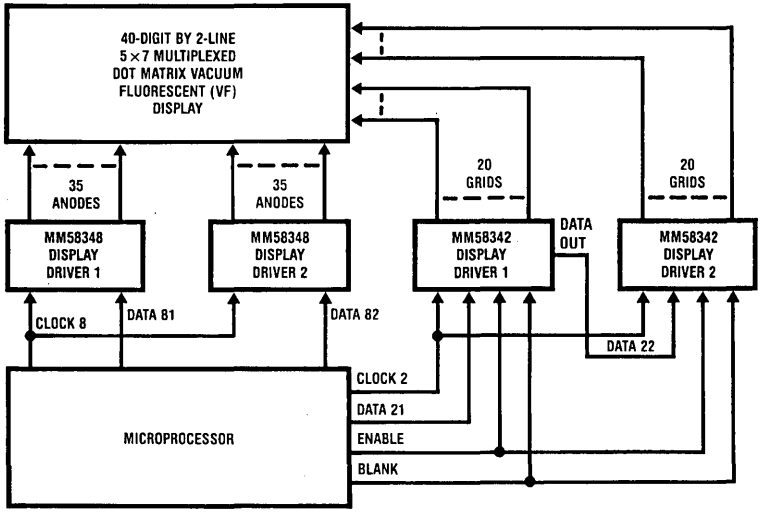
Timing Diagrams (Continued)



TL/F/7925–6

FIGURE 5. Timings (Data Format)

Typical Application



TL/F/7925–7

FIGURE 6. Microprocessor-Controlled Word Processor

National Semiconductor Corporation

MM58348 High Voltage Display Driver

General Description

The MM58348 is a monolithic MOS integrated circuit utilizing CMOS metal gate low threshold P and N-channel devices. It is available both in 40-pin molded dual-in-line packages or as dice. The MM58348 is particularly suited for driving high voltage (35V max) vacuum fluorescent (VF) displays (e.g., a 5 x 7 dot matrix display).

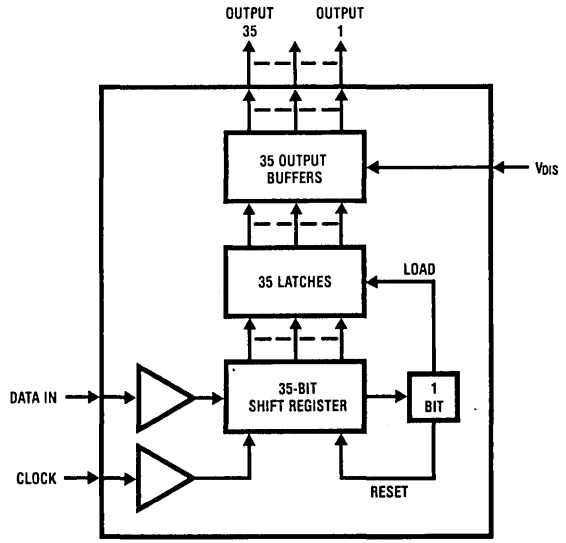
Applications

- COPS™ or microprocessor-driven displays
- Instrumentation readouts
- Industrial control indicator
- Digital clock, thermostat, counter, voltmeter
- Word processor text displays
- Automotive dashboards

Features

- Direct interface to high voltage display
- Serial data input
- No external resistors required
- Wide display power supply operation
- LSTTL compatible inputs
- Software compatible with NS display driver family
- Compatible with alphanumeric or dot matrix displays
- No load signal required

Block Diagram



TL/F/5601–1

FIGURE 1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Input Pin	V_{DD} + 0.3V to V_{SS} − 0.3V
Voltage at Any Display Pin	V_{DD} to V_{DD} − 36.5V
$V_{DD} + \lvert V_{DIS} \rvert$	36.5V
Storage Temperature	−65°C to + 150°C
Power Dissipation	500 mW at +85°C
Junction Temperature	130°C
Lead Temperature (Soldering, 10 seconds)	260°C

Operating Conditions

	Min	Max	Units
Supply Voltage (V_{DD}) V_{SS} = 0V	4.5	5.5	V
Display Voltage (V_{DIS})	−30	−10	V
Temperature Range	−40	+85	°C

DC Electrical Characteristics

T_A = −40°C to +85°C, V_{DD} = 5V ±0.5V, V_{SS} = 0V unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I_{DD}	Power Supply Currents	$V_{IN} = V_{SS}$ or V_{DD}, V_{DD} = 5.5V, V_{SS} = 0V, V_{DIS} Disconnected			150	μA
I_{DIS}		V_{DD} = 5.5V, V_{SS} = 0V, V_{DIS} = −30V, All Outputs Low			10	mA
V_{IL}	Input Logic Levels DATA IN, CLOCK Logic '0'				0.8	V
V_{IH}	Logic '1'		2.4			V
I_{IN}	Input Currents DATA IN, CLOCK	V_{IN} = 0V or V_{DD}	−10		10	μA
C_{IN}	Input Capacitance DATA IN, CLOCK				15	pF
R_{OFF}	Display Output Impedances Output Off *(Figure 3a)*	V_{DD} = 5.5V, V_{SS} = 0V				
		V_{DIS} = −10V	55		250	kΩ
		V_{DIS} = −20V	60		300	kΩ
		V_{DIS} = −30V	65		400	kΩ
R_{ON}	Output On *(Figure 3b)*	V_{DIS} = −10V		700	800	Ω
		V_{DIS} = −20V		600	750	Ω
		V_{DIS} = −30V		500	680	Ω
V_{DOL}	Display Output Low Voltage	V_{DD} = 5.5V, I_{OUT} = Open Circuit, −30V ≤ V_{DIS} ≤ −10V	V_{DIS}		V_{DIS} + 2	V

Note 1: 74LSTTL V_{OH} = 2.7V @ I_{OUT} = −400 μA, TTL V_{OH} = 2.4V @ I_{OUT} = −400 μA.

AC Electrical Characteristic T_A = −40°C to +85°C, V_{DD} = 5V ±0.5V

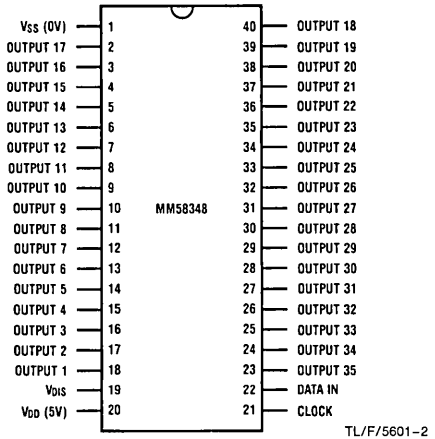
Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_C	Clock Input Frequency	*(Notes 2 and 3)*			1.0	MHz
t_H	Clock Input High Time		300			ns
t_L	Clock Input Low Time		300			ns
t_{DS}	Data Input Set-Up Time		100			ns
t_{DH}	Data Input Hold Time		100			ns

Note 2: AC input waveform specification for test purpose: $t_r \leq 20$ ns, $t_f \leq 20$ ns, f = 1 MHz, 50% ±10% duty cycle.

Note 3: Clock input rise and fall times must not exceed 5 μs.

Connection Diagrams

Dual-In-Line Package



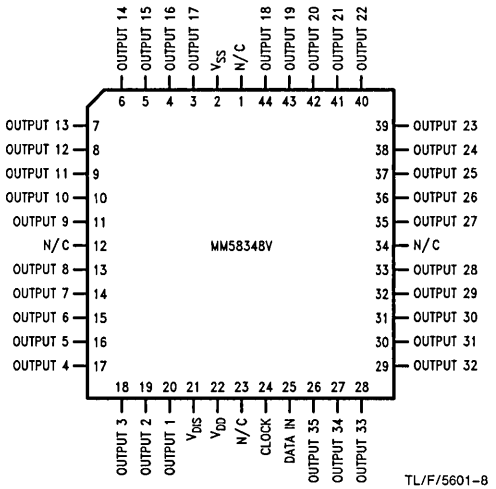
TL/F/5601–2

Top View

FIGURE 2

Order Number MM58348N
See NS Package Number N40A

Plastic Chip Carrier



TL/F/5601–8

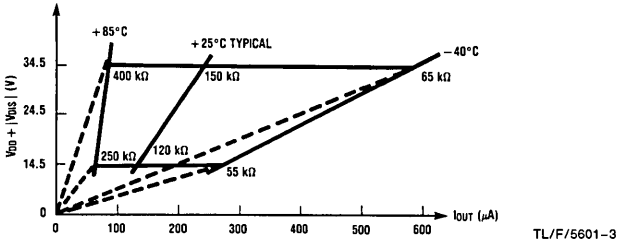
Top View
Order Number MM58348V
See NS Package Number V44A

Functional Description

This product is specifically designed to drive multiplexed or non-multiplexed high voltage alphanumeric or dot matrix vacuum fluorescent (VF) displays. Character generation is done externally in the microprocessor, with a serial data path to the display driver. The MM58348 uses two signals, DATA IN and CLOCK, with a format of a leading "1" followed by the 35 data bits, hence allowing data transfer without an additional signal. A block diagram of the MM58348 is shown in *Figure 1.*

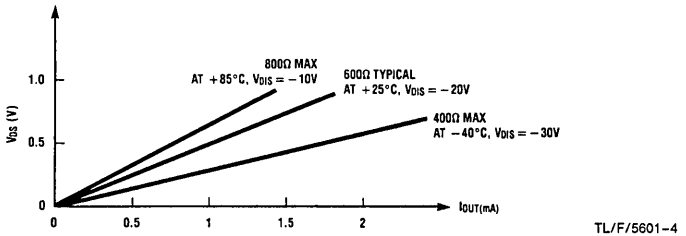
Figure 2 shows the pinout of the MM58348 device, where output 1 (pin 18) is equivalent to bit 1, (i.e., the first bit of data to be loaded into the shift register following the start bit). A logic "1" at the input will turn on the corresponding display digit/segment/dot output.

A significant reduction in discrete board components can be achieved by use of the MM58348, because external pull-down resistors are not required. Due to the nature of the output stage, both its on and off impedance values vary as a function of the display voltage applied. However, *Figure 3a* and *3b* show that this output impedance will remain constant for a fixed value of display voltage.



TL/F/5601–3

FIGURE 3a. Output Impedance Off



TL/F/5601–4

FIGURE 3b. Output Impedance On

Functional Description (Continued)

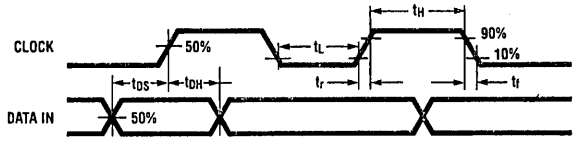
Figure 4 demonstrates the critical timing requirements between CLOCK and DATA IN for the MM58348.

When the chip first powers on, an internal reset is generated, resetting all registers and latches. The chip returns to normal operation on application of the start bit and the first clock pulse, and so all interface signals should be inactive at power on.

In *Figure 5,* a start bit of logic "1" precedes the 35 bits of data, each bit being accepted on the rising edge of CLOCK, i.e., a "0"–"1" transition. At the 36th clock, a LOAD signal is generated synchronously with the high state of the clock, thus loading the 35 bits of the shift register into the latches. At the low state of the clock, a RESET signal is generated, clearing all bits of the shift register for the next set of data. Hence, a complete set of 36 clock pulses is needed for the MM58348, or the shift register will not clear. If, at any given time, it is required that the display be cleared under microprocessor control, i.e., without power on reset, then the following flushing routine may be used. Clock in 36 "zeroes", followed by a "one" (start bit), followed by 35 "zeroes". This procedure will completely blank the display.

Figure 6 shows a schematic diagram of a microprocessor-based system where the MM58348 is used to provide the anode drive for a 32-digit 5 x 7 dot matrix vacuum fluorescent (VF) display. The grid drive in this example is provided by another member of the high voltage display driver family, namely the MM58341, which has the additional features of a BLANKING CONTROL pin, a DATA OUT pin, and an ENABLE (external load signal) pin.

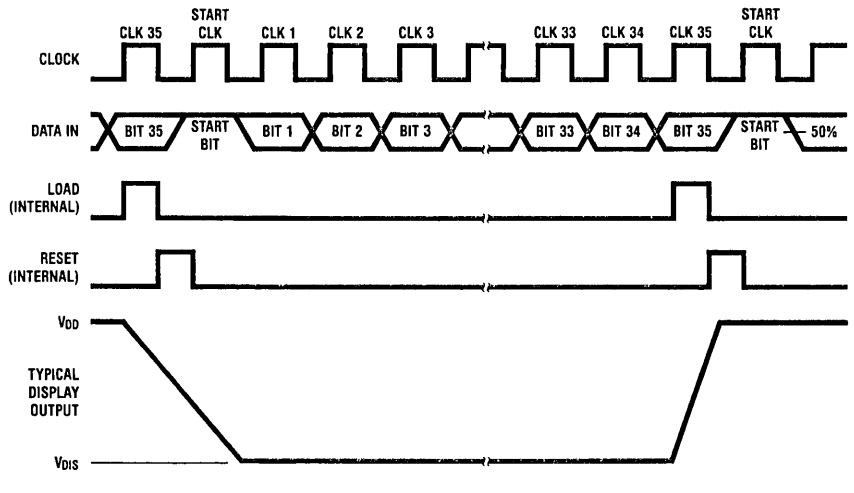
Timing Diagrams



TL/F/5601–5

For the purpose of AC measurement, V_{IH} = 2.4V, V_{IL} = 0.8V

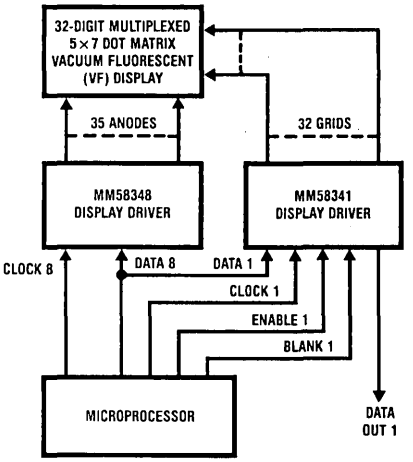
FIGURE 4. Clock and Data Timings



TL/F/5601–6

FIGURE 5. MM58348 Timings (Data Format)

Typical Application



TL/F/5601–7

FIGURE 6. Microprocessor-Controlled Word Processor

National Semiconductor Corporation

MM58341 High Voltage Display Driver

General Description

The MM58341 is a monolithic MOS integrated circuit utilizing CMOS metal gate low threshold P and N-channel devices. It is available both in 40-pin molded dual-in-line packages or as dice. The MM58341 is particularly suited for driving high voltage (35V max) vacuum fluorescent (VF) displays, (e.g., a 32-digit alphanumeric or dot matrix display).

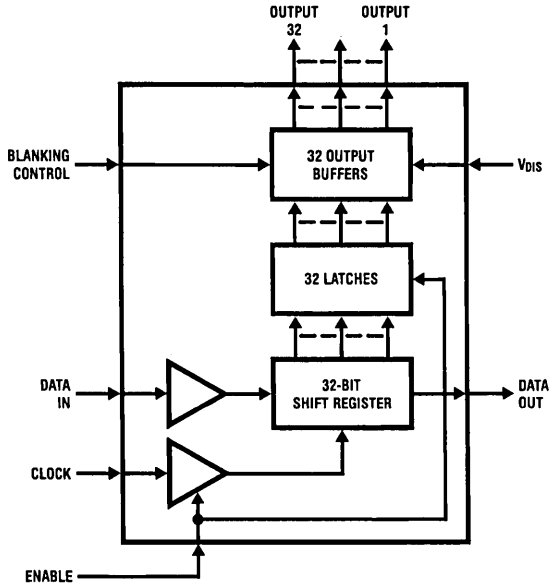
Applications

- COPS™ or microprocessor-driven displays
- Instrumentation readouts
- Industrial control indicator
- Digital clock, thermostat, counter, voltmeter
- Word processor text displays
- Automotive dashboards

Features

- Direct interface to high voltage display
- Serial data input
- No external resistors required
- Wide display power supply operation
- LSTTL compatible inputs
- Software compatible with NS display driver family
- Compatible with alphanumeric or dot matrix displays
- Display blanking control input
- Simple to cascade

Block Diagram



TL/F/5603–1

FIGURE 1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Input Pin	V_{DD} + 0.3V to V_{SS} − 0.3V
Voltage at Any Display Pin	V_{DD} to V_{DD} − 36.5V
$V_{DD} + \lvert V_{DIS} \rvert$	36.5V
Storage Temperature	−65°C to +150°C
Power Dissipation	500 mW at +85°C
Junction Temperature	130°C
Lead Temperature (Soldering, 10 seconds)	260°C

Operating Conditions

	Min	Max	Units
Supply Voltage (V_{DD}) V_{SS} = 0V	4.5	5.5	V
Display Voltage (V_{DIS})	−30	−10	V
Temperature Range	−40	+85	°C

DC Electrical Characteristics

T_A = −40°C to +85°C, V_{DD} = 5V ±0.5V, V_{SS} = 0V unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I_{DD}	Power Supply Currents	$V_{IN} = V_{SS}$ or V_{DD}, V_{SS} = 0V, V_{DIS} Disconnected			150	μA
I_{DIS}		V_{DD} = 5.5V, V_{SS} = 0V, V_{DIS} = −30V, All Outputs Low			10	mA
V_{IL}	Input Logic Levels DATA IN, CLOCK ENABLE, BLANK Logic '0'				0.8	V
V_{IH}	Input Logic Levels DATA IN, CLOCK ENABLE, BLANK Logic '1'	(Note 1)	2.4			V
V_{OH}	Data Output Logic Levels Logic '0'	I_{OUT} = 400 μA			0.4	V
V_{OH}	Data Output Logic Levels Logic '1'	I_{OUT} = −10 μA	V_{DD} − 0.5			V
V_{OH}	Data Output Logic Levels Logic '1'	I_{OUT} = −500 μA	2.8			V
I_{IN}	Input Currents DATA IN, CLOCK ENABLE, BLANK	V_{IN} = 0V or V_{DD}	−10		10	μA
C_{IN}	Input Capacitance DATA IN, CLOCK ENABLE, BLANK				15	pF
R_{OFF}	Display Output Impedances Output Off *(Figure 3a)*	V_{DD} = 5.5V, V_{SS} = 0V V_{DIS} = −10V V_{DIS} = −20V V_{DIS} = −30V	 55 60 65		 250 300 400	 kΩ kΩ kΩ
R_{ON}	Display Output Impedances Output On *(Figure 3b)*	 V_{DIS} = −10V V_{DIS} = −20V V_{DIS} = −30V		 700 600 500	 800 750 680	 Ω Ω Ω
V_{DOL}	Display Output Low Voltage	V_{DD} = 5.5V, I_{OUT} = Open Circuit, −30V ≤ V_{DIS} ≤ −10V	V_{DIS}		V_{DIS} + 2	V

Note 1: 74LSTTL V_{OH} = 2.7V @ I_{OUT} = −400 μA, TTL V_{OH} = 2.4V @ I_{OUT} = −400 μA.

AC Electrical Characteristics $T_A = -40°C$ to $+85°C$, $V_{DD} = 5V \pm 0.5V$

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_C	Clock Input Frequency	(Notes 3, 4)			800	kHz
t_H	Clock Input High Time		300			ns
t_L	Clock Input Low Time		300			ns
t_{DS}	Data Input Setup Time		100			ns
t_{DH}	Data Input Hold Time		100			ns
t_{ES}	Enable Input Setup Time		100			ns
t_{EH}	Enable Input Hold Time		100			ns
t_{CDO}	Data Output Clock Low to Data Out Time	$C_L = 50$ pF			500	ns

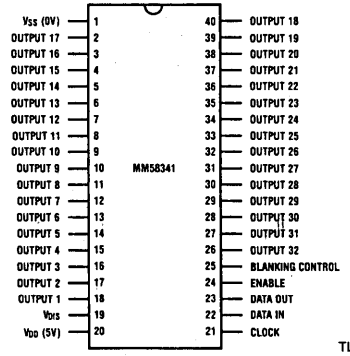
Note 2: Note that, for timing purposes, the signals ENABLE and BLANK can be considered to be totally independent of each other.

Note 3: AC input waveform specification for test purpose: $t_r \leq 20$ ns, $t_f \leq 20$ ns, $f = 800$ kHz, 50% ±10% duty cycle.

Note 4: Clock input rise and fall times must not exceed 5 μs.

Connection Diagrams

Dual-In-Line Package

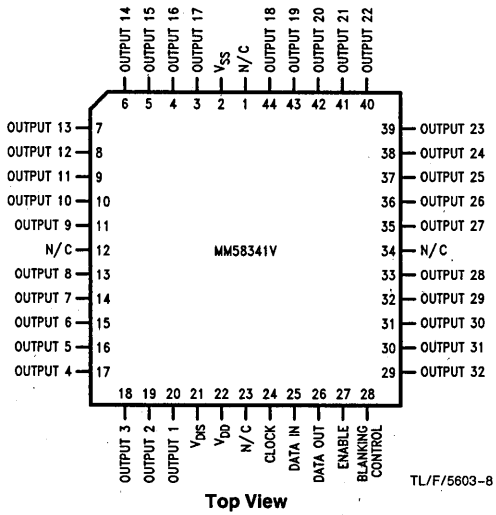


TL/F/5603-2

Top View

Order Number MM58341N
See NS Package Number N40A

Plastic Chip Carrier



TL/F/5603-8

Top View

Order Number MM58341V
See NS Package Number V44A

Functional Description

This product is specifically designed to drive multiplexed or non-multiplexed high voltage alphanumeric or dot matrix vacuum fluorescent (VF) displays. Character generation is done externally in the microprocessor, with a serial data path to the display driver. The MM58341 uses three signals, DATA IN, CLOCK and ENABLE, where ENABLE acts as an external load signal. Display blanking can be achieved by means of the BLANKING CONTROL input, and a logic '1' will turn off all sections of the display. A block diagram of the MM58341 is shown in *Figure 1.*

Figure 2 shows the pinout of the MM58341 device, where output 1 (pin 18) is equivalent to bit 1 (i.e., the first bit of data to be loaded into the shift register following ENABLE high). A logic '1' at the input will turn on the corresponding display digit/segment/dot output.

A significant reduction in discrete board components can be achieved by use of the MM58341, because external pull-down resistors are not required. Due to the nature of the output stage, both its on and off impedance values vary as a function of the display voltage applied. However, *Figures 3a* and *3b* show that this output impedance will remain constant for a fixed value of display voltage.

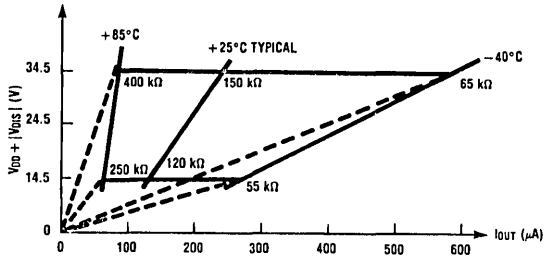
Figure 4 demonstrates the critical timing requirements between CLOCK and DATA IN for the MM58341.

When the chip first powers on, an internal reset is generated, resetting all registers and latches. The chip returns to normal operation on application of ENABLE, and so all interface signals should be inactive at power on.

In *Figure 5,* the ENABLE signal acts as an envelope, and only while this signal is at a logic '1' does the circuit accept CLOCK input signals. Data is transferred and shifted in the internal shift register on the rising clock edge, i.e., '0'–'1' transition. When the ENABLE signal goes low, the contents of the shift registers are latched, and the display will show new data. During data transfer, the display will show old data. DATA OUT is also provided on the MM58341, being output on the falling edge. At any time, the display may be blanked under processor control, using the BLANKING CONTROL input.

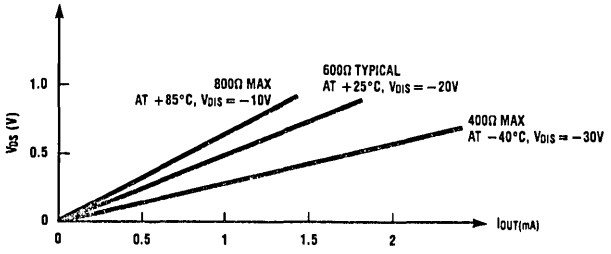
Figure 6 shows a schematic diagram of a microprocessor-based system where the MM58341 is used to provide the grid drive for a 32-digit 5 x 7 dot matrix vacuum fluorescent (VF) display. The anode drive in this example is provided by another member of the high voltage display driver family, namely the MM58348, which does not require an externally generated load signal.

Functional Decription (Continued)



TL/F/5603–3

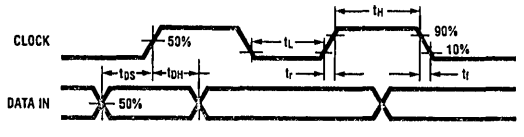
FIGURE 3a. Output Impedance Off



TL/F/5603–4

FIGURE 3b. Output Impedance On

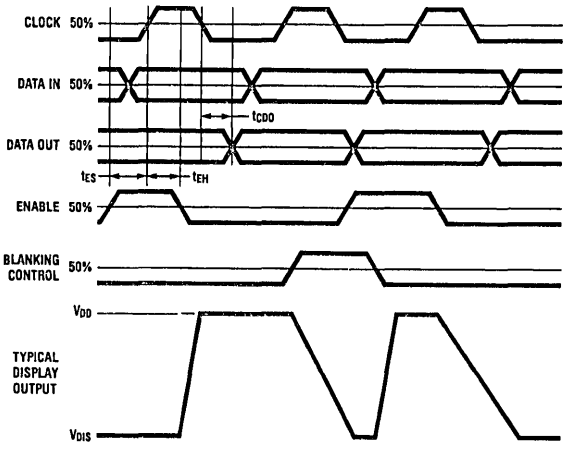
Timing Diagrams



TL/F/5603–5

For the purposes of AC measurements, V_{IH} = 2.4V, V_{IL} = 0.8V.

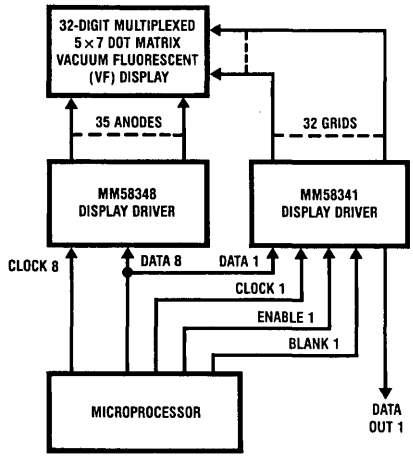
FIGURE 4. Clock and Data Timings



TL/F/5603–6

FIGURE 5. MM58341 Timings (Data Format)

Typical Application



TL/F/5603–7

FIGURE 6. Microprocessor-Controlled Word Processor

New CMOS Vacuum Fluorescent Drivers Enable Three Chip System to Provide Intelligent Control of Dot Matrix VF Display

National Semiconductor Corp.
Application Note 440
Tom Markman

INTRODUCTION

Vacuum Fluorescent (VF) displays are becoming more and more common in a variety of applications. Manufacturers of everything from Automobiles to Video Recorders have taken advantage of these easy to read displays. VF displays are available in a wide variety of configurations; clock displays, calculator displays, multi-segment, and dot matrix displays are readily available at a low cost. This application note develops and covers in some detail a small CMOS system consisting of a single chip microcontroller and two display drivers which control a 20 character, 5 x 7 dot matrix VF display.

Figure 1 shows the schematic of the system. The microcontroller, a COPS™ 424C, receives a character in ASCII form from the host system, stores the ASCII value of the character in its onboard RAM, converts the ASCII value to a 5 byte data word suitable for the display drivers and displays it on the VF display. The COPS also refreshes the display continuously while performing character update, much like a dumb terminal. Not including the address decoding logic, this application requires only the onboard RAM and ROM of the COPS424C, and National's MM58341 and MM58348 VF display drivers. If a steady message or a scrolling sentence is desired, only small changes in the COPS software are required. In this case the messages could be stored in the ROM of the COPS and the need for a host system would be eliminated.

VF DISPLAY AND VF DISPLAY DRIVER REQUIREMENTS

The display used in this application was an Itron #DC205G2. This 20 segment, 5 x 7 dot matrix, multiplexed display required a filament voltage of 5.7 Vac and a filament current of 37 mAac. The anode and grid voltages were supplied by the display drivers. The voltage and current requirements vary considerably for different displays depending on the size and number of characters, and the configuration (dot matrix, 7 segment, 14 segment, etc.). To determine the voltage requirements for a particular display, a simple calculation can be made. If maximum possible brightness of the display is desired, the following equation must be true:

$E_t \geq E_b + E_k + (I_b)(R_{on})$ where:

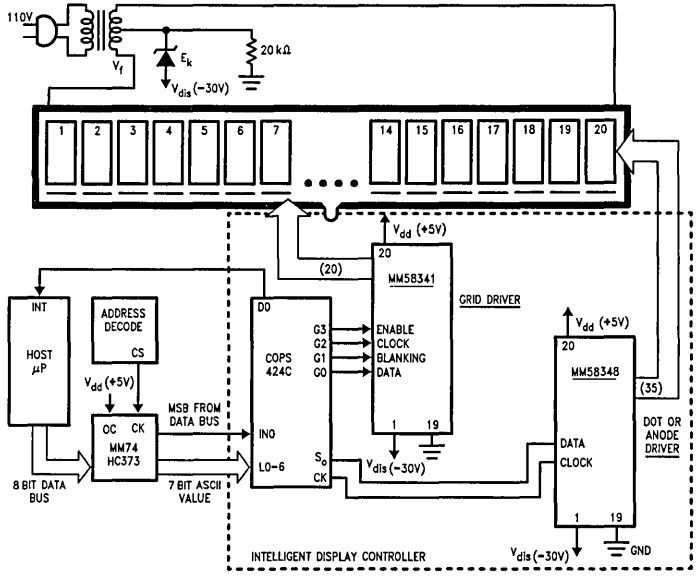
E_t is the total Voltage of the display drvier or $|V_{dis}| + V_{dd}$

E_k is the display Cathode Bias Voltage

$E_b = E_c$ is the typical Anode or Grid Voltage (V_{p-p})

I_b is the typical anode current (mAp-p)

R_{on} is the display driver output impedance (Ω)



TL/F/8683–1

FIGURE 1. System Diagram Showing the Basic 3-Chip Display Controller and the Interface to a Microprocessor System

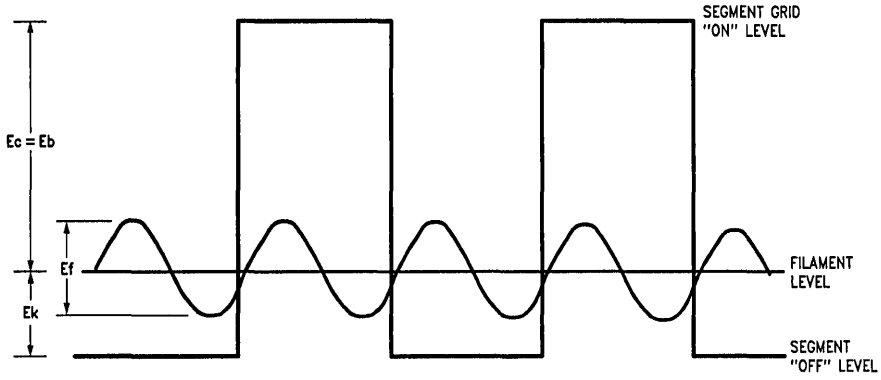
If the maximum brightness is not desired, the following equation can be used: $(E_t)(1.2) \geq E_b + E_K + (I_b)(R_{on})$. In this application, the calculated E_t was 42.25V, however, the display was legible under normal lighting conditions, with an E_t as low as 25V. If your display requires more than the 35V output of the MM58341 and MM58348, pin for pin compatible 60V VF Display Drivers (MM58241, MM58248) are available.

Figure 2 shows the relationship between the required VF display voltages. The cut-off voltage (E_K) is set by the Zener diode on the center tap of the filament transformer. This value is given in the VF display data sheet.

Avoiding Flicker and Pulsing

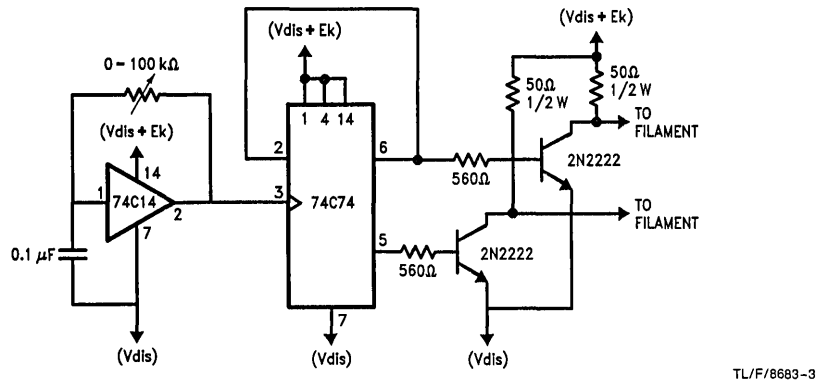
There are two different conditions which may cause the display to appear to flicker. The first is the refresh rate. This is particularly a problem on displays where the micro-controller must up-date more than 25 characters. Since the human eye begins to notice flicker at about 40 Hz, a display with a refresh rate less than that will appear to be flashing on and off.

The second type of flicker occurs when the refresh rate is between 40 Hz and 90 Hz. In this case, the display will appear to be rolling rather than flashing. This condition occurs when the refresh rate and the filament frequency are close together. If a character is only on during the time when the filament voltage is negative, it will appear to be slightly brighter than the character next to it which may only be on during the positive cycle of the filament voltage. If this is the case, as it was in this application, the simplest solution is to increase the frequency of the filament. A DC oscillator circuit, such as the one shown in *Figure 3*, can be used to replace the AC voltage source. The filament frequency can be easily adjusted to eliminate this condition.



TL/F/8683-2

FIGURE 2. Voltage Levels for VF Display



TL/F/8683-3

FIGURE 3. Filament Oscillator Circuit

VF Display Drivers

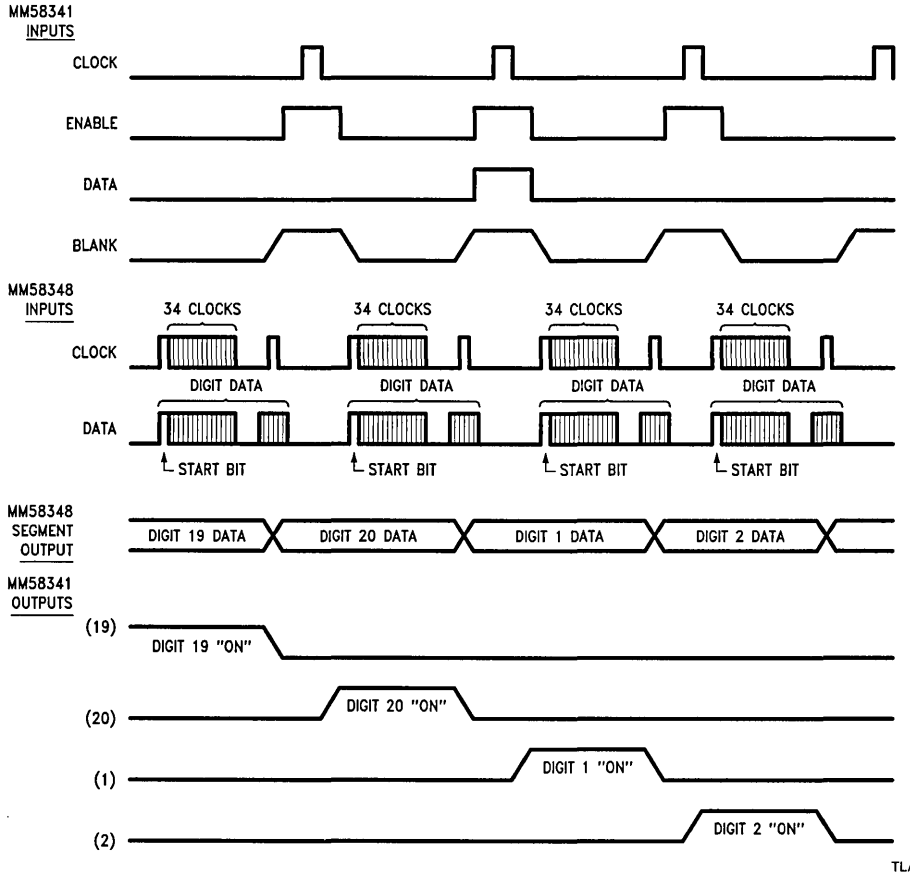
Two high voltage display drivers were needed to control the VF display. A MM58341, was used to control the grids and a MM58348 was used to control the individual pixels or anodes. Both of these drivers receive serial information and output 32 and 35 segments of data respectively.

The MM58341 has three control pins which make it ideal for controlling the grids of a VF display. The blanking control pin will turn off all segments of the display when a logic '1' is applied to this pin. This is particularly important for reducing ghosting, and controlling brightness. Ghosting is a condition where the last characters shadow appears behind the character being displayed. The enable pin acts as an envelope for the input signal. Only while it is at a logic '1' level will the circuit accept clock inputs. When the pin goes low, all the data is latched and displayed. A data out pin is also provided for cascading. If the display has more than 32 grids, a second grid driver can be cascaded by connecting the data out pin to the input data for the second grid driver.

The MM58348 is a 35 bit shift register and latch which is used to control each pixel or dot. When a leading 1, followed by 35 bits of data, is received, the data is latched and displayed. The chip is automatically reset upon power up.

MULTIPLEXED DISPLAY REFRESH TIMING

Considering first the digit driver (MM58341), it becomes clear that the digits must be enabled or refreshed sequentially and that this process must be continuous regardless if the display data has changed. The data for the MM58341 is simply a 1 followed by 19 zeroes where the 1 is shifted through the internal registers of the MM58341. As each digit is enabled, the corresponding segment data is displayed. To insure that no ghosting effects are seen during the transition between digits, the blanking control is activiated just before the data is latched into the dot or anode driver and deactivated just after the data has been latched. During this time when the blanking control is activated, the grid driver is clocked shifting the 1 to the next location. *Figure 4* shows the micro-controller waveforms and the resultant display waveforms for the 20 character display.



TL/F/8683–4

FIGURE 4. Timing Diagram

In between digit strobes, the segment data is updated. The first 34 bits of segment data are set up in the dot driver and the blanking signal is activated to disable all 20 digits. The 35th bit of data is clocked in, updating the segments. Since the MM58348 resets its internal shift register each time the data is latched, it can accept all but the final data bit while still displaying the previous digit. The digit driver is then clocked, shifting the digit strobe to the next position. The enable is then brought low, enabling the next digit. Finally blanking control is deactivated and the data displayed.

During the time which the blanking control is high, the order in which the segments or the digits are updated is not critical. Since this occurs while the display is blank. The digit driver may be clocked first, or the segments could be changed first. In general, the philosophy for the driving this VF multiplexed display is outlined in *Figure 5.*

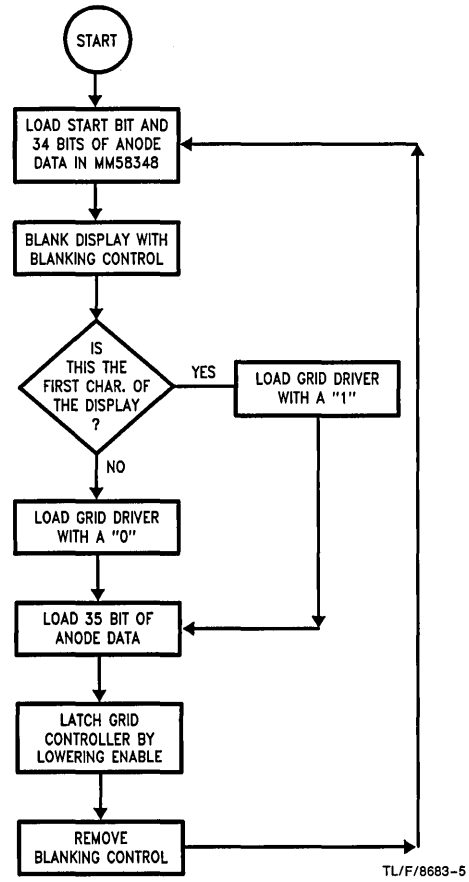


FIGURE 5. Flowchart for Display Drivers

HOST INTERFACE AND PROGRAMMING

With a minimal amount of address decoding and an eight bit latch, COPS can be interfaced with a common microprocessor bus. When a character has been input into the host to be displayed, the ASCII value of that character is latched into the eight bit latch (MM74HC373) and is read on the L port (L0–6) of the COPS. The MSB of the ASCII value must be a logic 1. This MSB is the signal to the COPS that a new character is being presented. Once the character has been stored, an interrupt is sent from the COPS to the host through the D-0 port. The COPS checks for a new character being input every 200 μs. If a character is being sent, 1 ms is required to store that character in the RAM of the COPS. With the COPS controlling the display, the host micro-processor is not being tied down with character look-up and display refresh. A simple flowchart of the host requirements is shown in *Figure 6.*

COPS SOFTWARE

There are four main sections of the COPS software. The first section, the initialization of the RAM, sets up the RAM as shown in *Figure 7.* A '0' is stored in all of the LSB positions and a '2' is stored in all of the MSB positions. Since the COPS is in a constant display loop, this is necessary to insure a blank display. 20H is the ASCII value of a space. With the RAM set up in this way, a maximum of 28 characters can be stored in RAM. Since the display in this application is only 20 characters long, RAM locations M1,4 to M1,11 and M3,4 to M3,11 are not used. RAM locations 1,12 to 1,15 and 3,12 to 3,15 are used as temporary storage throughout the program and cannot be used for character storage.

The second part of the program, stores the new characters sent by the host CPU in RAM. Once a character has been sent, this section of the program checks the ASCII value of that character to see if it is a control character or a display character. If it is a display character, the character is stored in RAM and an interrupt is sent to the host. There are three control characters which the COPS program will recognize. Cursor forward (ASCII value 08H) moves the cursor forward without destroying the data, cursor backwards (ASCII value 0CH) moves the cursor backwards without destroying the data, and return (ASCII value 0DH) will clear the display and put the cursor at the beginning of the display. To recognize and store a character, 1 ms is required.

The third part of the program, the display loop, is the heart of the program. Unless a new character has been detected, the program is always in this loop. This section does the

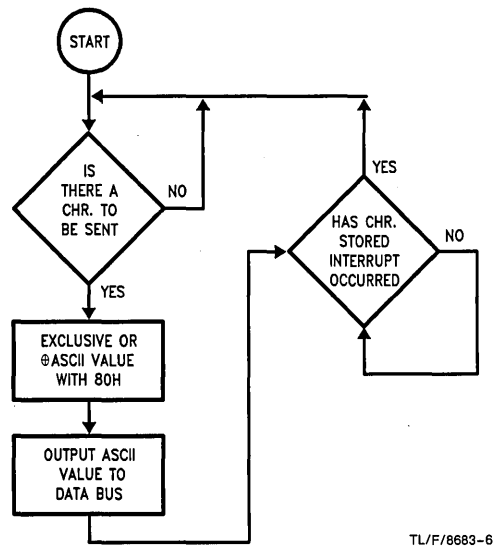


FIGURE 6. Host System Flowchart

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
LSB Chr 1	LSB Chr 2	LSB Chr 3	LSB Chr 4	LSB Chr 5	LSB Chr 6	LSB Chr 7	LSB Chr 8	LSB Chr 9	LSB Chr 10	LSB Chr 11	LSB Chr 12	LSB Chr 13	LSB Chr 14	LSB Chr 15	LSB Chr 16	M0
MSB Pointer	LSB Pointer	Temp. ASCII STORAGE										LSB Chr 17	LSB Chr 18	LSB Chr 19	LSB Chr 20	M1
MSB Chr 1	MSB Chr 2	MSB Chr 3	MSB Chr 4	MSB Chr 5	MSB Chr 6	MSB Chr 7	MSB Chr 8	MSB Chr 9	MSB Chr 10	MSB Chr 11	MSB Chr 12	MSB Chr 13	MSB Chr 14	MSB Chr 15	MSB Chr 16	M2
Temp. Storage of Pointer												MSB Chr 17	MSB Chr 18	MSB Chr 19	MSB Chr 20	M3

FIGURE 7. COPS RAM Map

Matrix	PAD	Column 1	Column 2	Column 3	Column 4	Column 5	PAD
Binary	0 0 0 1 0 0 1 1 1 1 1 0 1 0 1 0 0 0 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 1 1 1 1 1 0						
Hex.	13		EA		24	28	3E

FIGURE 8

character font look-up, shifts the character data out the COPS serial port to the MM58348, and controls the MM58341 through the four bit parallel port (G0–4). Because the most significant nibble of the program counter is used as part of some COPS instructions, it is important that parts of the program are located at specific locations in ROM.

The final part of the program is the data. Each character is represented by a 5 byte data word. Each byte of the data word is stored at a different location in ROM. Fonts for characters with the ASCII values from 20H–5AH have already been stored in ROM. These characters can be changed or more characters can be added. The only limitation to the number of characters is the amount of available ROM.

CREATING THE 5 BYTE DATA WORD

Any number or combination of pixels or dots can be turned on at a time. To create a new character, it is easiest to first create a binary string which represents the character. A '1' in the binary string will turn on the pixel, a '0' will turn it off. To create this string, start in the upper left corner of the matrix and go down the columns.

The letter 'A' (*Figure 9*) would have a binary string shown in *Figure 8*. The data must be padded to make it an even 5 bytes in length. The pad at the beginning of the data (0001) is used as the leading 1 for the MM58348. The one bit pad at the end of the binary string must be a 0. If a 1 were sent as the pad, it would be used as the start bit for the next character.

The 5 byte data word that would be stored in ROM and represent the letter 'A' would then be 13EA24283E.

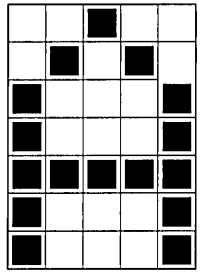
STORING THE DATA IN ROM

The 5 bytes of data are stored in 5 different locations in ROM. The first byte of data will be stored, LSB first, at location 200H plus the ASCII value of the character. For example, the ASCII value of the letter 'A' is 41H. The first byte of data for the letter 'A' would be stored, least significant bit first, at 241H. The second byte of data is stored at the location of the first data byte plus 60H or in this case at 2A1H. The location of the third byte is 40H plus the location of the second byte. In this case, the third byte of data would be stored at 2E1H. The fourth byte of data is stored at 300H plus the ASCII value of the character or at 341H for the letter 'A'. The final byte of data is stored 40H from the fourth byte or at 381H. Remember the LSB of each byte is stored first. Table I shows the locations in ROM and the values stored in them for the letter 'A'.

This application shows a VF display controller designed with a minimum number of IC's. If additional information about VF displays or VF display drivers is required, refer to Application Note AN-371 (The MM58348/342/341/248/242/241 direct drive Vacuum Fluorescent (VF) Displays.

TABLE I. Character Data of 'A' and Its Locations In ROM

Address In ROM	Data Stored
0241H	31
02A1H	AE
02E1H	42
0341H	82
0381H	E3



TL/F/8683–7

FIGURE 9. 5 x 7 Character as Stored In ROM

Section 1 of COPS Software

```
        .CHIP 424C          ;DEFINES COPS CHIP
;THIS SECTION INITIALIZES THE RAM IN THE COPS BY LOADING A
;2 IN THE MSB AND A 0 IN THE LSB LOCATIONS OF EACH CHARACTER.
;IT ALSO STOPS THE CLOCK AND SETS THE POINTER AT THE FIRST
;CHARACTER OF THE DISPLAY.

RESET:          CLRA
                LBI 3,15        ;LOADS A 2 IN ALL
                JSR CLEAR2      ;MSB LOCATIONS
                LBI 2,15        ;LOADS A 2 IN ALL
                JSR CLEAR2      ;MSB LOCATIONS
                LBI 1,15        ;LOADS A 0 IN ALL
                JSR CLEAR       ;LSB LOCATIONS
                LBI 0,15        ;LOADS A 0 IN ALL
                JSR CLEAR       ;LSB LOCATIONS

                CLRA            ;LOADS POINTER IN RAM
                XAD 1,15        ;MSB IN 1,0F
                CLRA
                AISC 15         ;LSB IN 1,0E
                XAD 1,14

                RC              ;RESETS CARRY TO
                XAS             ;    STOP CLOCK
                JMP START

CLEAR:          CLRA            ;CLEARS REGISTORS
                XDS 0
                JMP CLEAR
                RET

CLEAR2:         CLRA            ;PUTS A 2 IN REGISTORS
                AISC 02
                XDS 0
                JMP CLEAR2
                RET
```

Section 2 of COPS Software

```
;THIS SECTION OF CODE IS ONLY EXECUTED WHEN A NEW
;CHARACTER HAS BEEN ENTERED. IF THE CHARACTER IS
;A CONTROL CHARACTER, THE CURSOR IS MOVED ACCORDINGLY,
;OTHERWISE THE CHARACTER IS STORED IN THE RAM OF THE COPS.

          ;NEW CHARACTER HAS BEEN ENTERED
NEW:            LBI 1,0C        ;DUMMY POINTER
                INL             ;READS ASCII FROM
                XIS  0          ;DATA BUS
                X 0
                LDD 1,0D
                RC              ;CHAR.  MSB=0 THEN YES
                AISC 15         ;MSB<>0 THEN NO
                JMP SPECIAL
                AISC 01
                LDD 1,0E        ;STORE ASCII IN RAM
                CAB
                LDD 1,0F
                XABR
                LDD 1,0C        ;MSB IN 1,0C
                X 2
                LDD 1,0D        ;LSB IN 1, 0D
                X 0
```

Section 2 of COPS Software (Continued)

```
                JSR CURFOR
                LBI 0,01        ;SENDS INTERRUPT TO
                OBD             ;  HOST.  CHAR. IS
                LBI 0,0         ;  STORED IN RAM
                OBD
                JMP START

          ;SPECIAL CHARS. (CR, LF, CLEAR DISPLAY)

CURFOR:         LDD 1,0E        ;MOVES CURSOR FORWARD ONE
                COMP            ;SPACE.  IF CURSOR IS
                AISC 01         ;MOVED BEYOND THE END OF
                JMP OK          ;DISPLAY, IT WRAPS AROUND
                AISC 0F         ;TO THE OTHER END.  DATA IS
                XAD 1,0E        ;NOT DESTROYED BY MOVING
                CLRA            ;CURSOR
                AISC 01
                LBI 1,0F
                XOR
                JMP SKIP
OK:             COMP
SKIP:           LBI 1,0E
                X 0
                RET
CURBAC:         LDD 1,0F        ;MOVES CURSOR BACK ONE
                AISC 01         ;CHARACTER.  DOES NOT
                JMP GOOD        ;DESTROY DATA AS IT IS MOVED
                LBI 1,0E        ;IF MOVED BEYOND THE
                CLRA            ;END OF THE DISPLAY IT
                AISC 01         ;WRAPS AROUND TO THE OTHER
                XOR             ;END
                X 0
                JMP START
GOOD:           XAD 1,0F
                JMP START

SPECIAL:        LDD 1,0C        ;CONTROL CHAR. HAS BEEN
                AISC 03         ;DETECTED
                JMP NOTRET
                JMP RESET       ;RETURN CLEARS DISPLAY,STARTS
                                ;PROGRAM OVER
NOTRET:         AISC 01         ;NOT RETURN, CHECK FOR CURSOR
                JMP CFOR        ;FORWARD
                JMP CURBAC      ;BY DEFAULT, CURSOR BACKWARDS

CFOR:           JSR CURFOR
                JMP START

          ;DISPLAY LOOP
```

```
;THIS IS THE DISPLAY LOOP OF THE PROGRAM. UNLESS A NEW CHARACTER
;HAS BEEN ENTERED AND IS BEING STORED, THE PROGRAM IS ALWAYS IN
;THIS DISPLAY LOOP. IT LOOKS UP THE CHARACTER FONT, SHIFTS THE
;CHARACTER DATA OUT THE SERIAL PORT AND CONTROLS THE GRID DRIVER.

START:      LBI 2,15        ;DISPLAY LOOP POINTER
            JSR HERE        ;GOTO DISPLAY LOOP
            LBI 3,03        ;SECOND DISPLAY LOOP POINTER
            JSR HERE        ;GOTO DISPLAY LOOP
            OGI 09          ;LOADS A 1 IN GRID DRIVER
            OGI 0D
            OGI 09

            JMP START

        ;CHECKS FOR NEW CHAR

HERE:       RC
            ININ
            AISC 15
            JMP OLDCHR
            JMP NEW

        ;DISPLAY LOOP FOR OLD CHAR AND
        ;  LOOK UP

OLDCHR:     LD 2            ;LOOKS UP FIRST BYTE OF CHR.FONT
            JSR DATA4       ;    200H+ASCII VALUE
            AISC 06         ;ADDS 06H TO MSB OF ASCII
            JSR DATA2       ;LOOKS UP SECOND BYTE OF CHR FONT
            AISC 0A         ;ADDS 0AH TO MSB OF ASCII
            JSR DATA2       ;LOOKS UP THIRD BYTE OF CHR. FONT
            JSR DATA3       ;LOOKS UP THIRD BYTE OF CHR. FONT
                            ;    AT 300H+ASCII VALUE
            AISC 06         ;ADDS 06H TO MSB OF ASCII VALUE
            OGI 02          ;TURNS ON BLANKING CONTROL
            JSR DATA3       ;LOOKS UP LAST BYTE OF CHR. FONT
        ;CLOCKS A 0 IN GRID DRIVER

            OGI 0A          ;ENABLE,BLANKING CONTROL
            OGI 0E          ;ENABLE,BLANKING CONTROL,CLOCK
            OGI 0A          ;ENABLE,BLANKING CONTROL
            OGI 00          ;A 0 SHIFTED IN

            LD 0
            XDS 2
            JMP HERE
            RET

RIGHT:      LBI 3,15
            CQMA
            JSR SHIFT       ;OUTPUTS A
            X 0             ;NEW DATA
            JSR SHIFT       ;OUTPUTS A
            LEI 01          ;COUNTER MODE
            LDD 3,14        ;1,0 IN A
            XABR            ;A IN BR
            LDD 3,13        ;1,1 IN A
            CAB             ;A IN BD
            LD 2
            RET
```

Section 3 of COPS Software (Continued)

```
POINTER:        LEI 01          ;COUNTER MODE
                XAS             ;A IN SIO
                XABR            ;BR IN A
                AISC 02         ;ADD 2
                XAD 3,14        ;A IN 1,0
                CBA             ;BD IN A
                XAD 3,13        ;A IN 1,1
                LBI 3,15
                XAS             ;SIO IN A
                LEI 08          ;SERIAL MODE
                JMP RIGHT
          ;SHIFTS OUT SERIAL PORT

SHIFT:          LEI 08          ;THIS ROUTINE SHIFTS THE DATA
                SC              ;FROM THE SI/O REGISTER OUT
                XAS             ;THE SERIAL PORT WITH EACH
                NOP             ;CLOCK CYCLE
                NOP
                RC
                XAS
                RET

          .=0200
DATA3:          LQID
                JMP RIGHT
DATA4:          LQID
                JMP POINTER

          .=0300
DATA3:          LQID
                JMP RIGHT
```

Section 4 of COPS Software

```
;THE CHARACTER FONTS FOR THE CHARACTERS WITH ASCII VALUES BETWEEN
20H AND 5AH HAVE BEEN STORED IN THIS SECTION OF THE PROGRAM.

          ;DATA FOR FIRST 2 BYTES OF EACH
          ;    CHAR.

          .=0220
     .WORD 001, 001, 001, 021, 021, 0C1, 061, 001
     .WORD 031, 001, 041, 011, 001, 011, 001, 001
     .WORD 071, 001, 041, 081, 011, 0E1, 031, 081
     .WORD 061, 061, 001, 001, 001, 021, 001, 041
     .WORD 071, 031, 081, 071, 081, 0F1, 0F1, 071
     .WORD 0F1, 081, 081, 0F1, 0F1, 0F1, 0F1, 071
     .WORD 0F1, 071, 0F1, 061, 081, 0F1, 0F1, 0F1
     .WORD 0C1, 0C1, 081
```

```
        ;DATA FOR SECOND 2 BYTES OF EACH
        ;    CHAR.

        .=0280
.WORD 000, 000, 0C1, 0F9, 0A4, 095, 02D, 000
.WORD 088, 000, 054, 020, 000, 020, 000, 014
.WORD 01D, 082, 003, 005, 058, 045, 0AC, 001
.WORD 02D, 023, 000, 000, 020, 058, 001, 001
.WORD 00D, 0AE, 0F3, 00D, 0F3, 02F, 02F, 00D
.WORD 02E, 003, 00D, 02E, 00E, 08E, 08E, 00D
.WORD 02F, 00D, 02F, 025, 001, 00C, 008, 00C
.WORD 056, 040, 017

        ;THIRD 2 BYTES OF DATA FOR EACH CHAR.


        .=02C0
.WORD 000, 0E3, 000, 0AC, 0FB, 040, 0A5, 083
.WORD 00A, 002, 0F3, 0F1, 034, 040, 008, 040
.WORD 046, 0F7, 02E, 046, 021, 086, 046, 02E
.WORD 046, 046, 0A0, 0B4, 0A0, 0A0, 015, 022
.WORD 0E6, 042, 04E, 006, 00E, 046, 042, 046
.WORD 040, 0F7, 006, 0A0, 004, 080, 0E0, 006
.WORD 042, 026, 062, 046, 0F3, 004, 008, 034
.WORD 040, 070, 046

        ;FOURTH TWO BYTES OF DATA FOR EACH CHAR.


        .=0320
.WORD 000, 008, 007, 0F7, 0AA, 031, 028, 000
.WORD 008, 02A, 049, 080, 000, 080, 000, 001
.WORD 01D, 018, 09C, 09D, 0F7, 01D, 09C, 084
.WORD 09C, 0AC, 000, 000, 022, 041, 041, 08C
.WORD 0DC, 082, 09C, 01C, 01C, 09C, 084, 09C
.WORD 080, 01C, 0EF, 022, 018, 002, 020, 01C
.WORD 084, 02C, 0A4, 09C, 00C, 018, 028, 010
.WORD 041, 009, 01D

        ;LAST BYTES OF DATA FOR EACH CHAR.


        .=0380
.WORD 000, 000, 000, 082, 084, 064, 0A0, 000
.WORD 000, 083, 044, 001, 000, 001, 000, 004
.WORD 0C7, 020, 026, 0CC, 080, 0C9, 0C8, 00E
.WORD 0C6, 087, 000, 000, 028, 082, 001, 006
.WORD 027, 0E3, 0C6, 044, 0C7, 028, 008, 0C5
.WORD 0EF, 028, 008, 028, 020, 0EF, 0EF, 0C7
.WORD 006, 0A7, 026, 0C4, 008, 0CF, 08F, 0CF
.WORD 06C, 00C, 02C


.END
```

National Semiconductor Corporation

PRELIMINARY

microCMOS

NMC27C32B 32,768-Bit (4k x 8) UV Erasable CMOS PROM

General Description

The NMC27C32B is a high-speed 32k UV erasable and electrically reprogrammable CMOS EPROM, ideally suited for applications where fast turnaround, pattern experimentation and low power consumption are important requirements.

The NMC27C32B is designed to operate with a single +5V power supply with ±10% tolerance. The CMOS design allows the part to operate over the Extended Temperature Range.

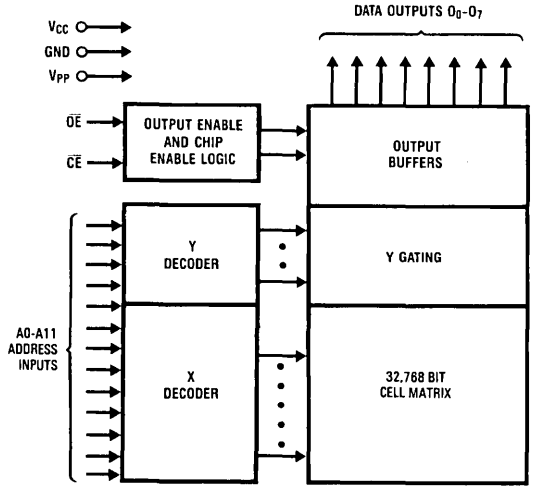
The NMC27C32B is packaged in a 24-pin dual-in-line package with transparent lid. The transparent lid allows the user to expose the chip to ultraviolet light to erase the bit pattern. A new pattern can then be written electrically into the device by following the programming procedure.

This EPROM is fabricated with National's proprietary, time proven microCMOS double-ploy silicon gate technology which combines high performance and high density with low power consumption and excellent reliability.

Features

- Access time down to 200 ns, microCMOS technology
- Low CMOS power consumption
  - — Active Power 27.5 mW max
  - — Standby Power 0.55 mW max
- Performance compatible to NSC800™ CMOS microprocessor
- Single 5V power supply
- Extended temperature range (NMC27C32BQE), −40°C to 85°C, available
- Pin compatible with NMOS 32k EPROMS
- Fast and reliable programming
- Static operation—no clocks required
- TTL, CMOS compatible inputs/outputs
- TRI-STATE® output

Block Diagram



TL/D/8827–1

Pin Names

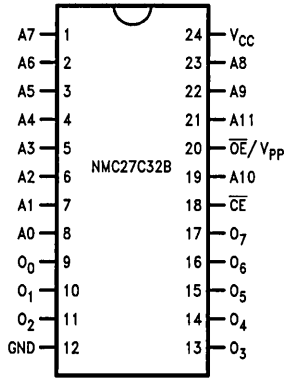
A0–A11	Addresses
$\overline{CE}$	Chip Enable
$\overline{OE}$	Output Enable
O_0–O_7	Outputs
$\overline{PGM}$	Program
NC	No Connect

3

Connection Diagram

27C56 27256	27C128 27128	27C64 2764	27C16 2716
V_{PP}	V_{PP}	V_{PP}	
A12	A12	A12	
A7	A7	A7	A7
A6	A6	A6	A6
A5	A5	A5	A5
A4	A4	A4	A4
A3	A3	A3	A3
A2	A2	A2	A2
A1	A1	A1	A1
A0	A0	A0	A0
O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2
GND	GND	GND	GND

Dual-In-Line Package



TL/D/8827–2

Top View

27C16 2716	27C64 2764	27C128 27128	27C256 27256
	V_{CC} $\overline{PGM}$	V_{CC} $\overline{PGM}$	V_{CC} A14
V_{CC}	NC	A13	A13
A8	A8	A8	A8
A9	A9	A9	A9
VPP	A11	A11	A11
$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$
A10	A10	A10	A10
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3

Note: National's socket compatible EPROM pin configurations are shown in the blocks adjacent to the NMC27C32B pins.

See NS Package Number J24A-Q

Commercial Temp Range (0°C to +70°C)

V_{CC} = 5V ±10%

Parameter/Order Number	Access Time (ns)
NMC27C32BQ200	200
NMC27C32BQ350	350

Extended Temp Range (−40°C to +85°C)

V_{CC} = 5V ±10%

Parameter/Order Number	Access Time (ns)
NMC27C32BQE200	200
NMC27C32BQE350	350

Absolute Maximum Ratings (Note 1)

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Temperature Under Bias	−10°C to +80°C
Storage Temperature	−65°C to +125°C
All Input Voltages with Respect to Ground	+6.5V to −0.3V
All Output Voltages with Respect to Ground	V_{CC} +0.3V to GND−0.3V
V_{PP} Supply and A9 Voltage with Respect to Ground	+14.0V to −0.3V
Power Dissipation	1.0W
Lead Temp. (Soldering, 10 sec.)	300°C

ESD rating to be determined.

Operating Conditions (Note 7)

Temperature Range	
NMC27C32BQ200, 350	0°C to +70°C
NMC27C32BQE200, E350	−40°C to +85°C
V_{CC} Power Supply	+5V ±10%

READ OPERATION

DC Electrical Characteristics

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
I_{LI}	Input Load Current	$V_{IN} = V_{CC}$ or GND			10	μA
I_{LO}	Output Leakage Current	$V_{OUT} = V_{CC}$ or GND, $\overline{CE} = V_{IH}$			10	μA
I_{CC1}	V_{CC} Current (Active) TTL Inputs	$\overline{CE} = V_{IL}$, f = 1 MHz Inputs = V_{IH} or V_{IL} I/O = 0 mA		2	10	mA
I_{CC2}	V_{CC} Current (Active) CMOS Inputs	$\overline{CE}$ = GND, f = 1 MHz Inputs = V_{CC} or GND, I/O = 0 mA		1	5	mA
I_{CCSB1}	V_{CC} Current (Standby) TTL Inputs	$\overline{CE} = V_{IH}$		0.1	1	mA
I_{CCSB2}	V_{CC} Current (Standby) CMOS Inputs	$\overline{CE} = V_{CC}$		0.01	100	μA
V_{IL}	Input Low Voltage		−0.1		0.8	V
V_{IH}	Input High Voltage		2.0		V_{CC} +1	V
V_{OL1}	Output Low Voltage	I_{OL} = 2.1 mA			0.45	V
V_{OH1}	Output High Voltage	I_{OH} = −400 μA	2.4			V
V_{OL2}	Output Low Voltage	I_{OL} = 0 μA			0.1	V
V_{OH2}	Output High Voltage	I_{OH} = 0 μA	V_{CC} −0.1			V

AC Electrical Characteristics

Symbol	Parameter	Conditions	NMC27C32B 200, E200		NMC27C32B 350, E350		Units
			Min	Max	Min	Max	
t_{ACC}	Address to Output Delay	$\overline{CE} = \overline{OE} = V_{IL}$		200		350	ns
t_{CE}	$\overline{CE}$ to Output Delay	$\overline{OE} = V_{IL}$		200		350	ns
t_{OE}	$\overline{OE}$ to Output Delay	$\overline{CE} = V_{IL}$		60		150	ns
t_{DF}	$\overline{OE}$ High to Output Float	$\overline{CE} = V_{IL}$	0	60	0	130	ns
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$, Whichever Occurred First	$\overline{CE} = \overline{OE} = V_{IL}$	0		0		ns

3

Capacitance $T_A = +25°C$, f = 1 MHz (Note 3)

Symbol	Parameter	Conditions	Typ	Max	Units
C_{IN}	Input Capacitance	$V_{IN} = 0V$	4	6	pF
C_{OUT}	Output Capacitance	$V_{OUT} = 0V$	8	12	pF

AC Test Conditions

Output Load 1 TTL Gate and $C_L = 100$ pF (Note 9)

Input Rise and Fall Times ≤20 ns

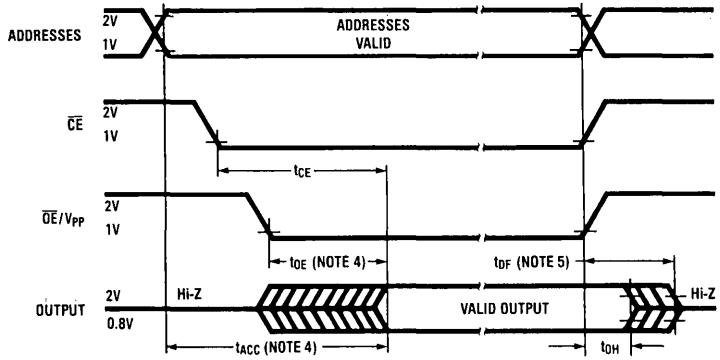
Input Pulse Levels 0.45V to 2.4V

Timing Measurement Reference Level

Inputs 1V and 2V

Outputs 0.8V and 2V

AC Waveforms



TL/D/8827-3

Note 1: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Note 2: Typical values are for $T_A = +25°C$ and nominal supply voltages.

Note 3: This parameter is only sampled and is not 100% tested.

Note 4: $\overline{OE}$ may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of $\overline{CE}$ without inpacting t_{ACC}.

Note 5: The t_{DF} compare level is determined as follows:

High to TRI-STATE, the measured V_{OH1} (DC) $-0.10V$

Low to TRI-STATE, the measured V_{OL1} (DC) $+0.10V$

Note 6: TRI-STATE may be attained using $\overline{OE}$ or $\overline{CE}$.

Note 7: The power switching characteristics of EPROMs require careful device decoupling. It is recommended that at least a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND.

Note 8: The outputs must be restricted to $V_{CC} + 0.3V$ to avoid latch-up and device damage.

Note 9: 1 TTL Gate: $I_{OL} = 1.6$ mA, $I_{OH} = -400$ μA.

C_L: 100 pF includes fixture capacitance.

Programming Characteristics $T_A = +25°C \pm 5°C$, $V_{CC} = 6V \pm 0.25V$ $V_{PP} = 12.2 - 13.3V$ (Notes 2, 3, & 4)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{AS}	Address Set-Up Time		2			μs
t_{OES}	$\overline{OE}$ Set-Up Time		2			μs
t_{DS}	Data Set-Up Time		2			μs
t_{AH}	Address Hold Time		0			μs
t_{DH}	Data Hold Time		2			μs
t_{VR}	V_{PP} Recovery Time		2			μs
t_{OEH}	$\overline{OE}$ Hold Time		2			μs
t_{DV}	Data Valid From $\overline{CE}$	$\overline{CE} = \overline{OE} = V_{IL}$			1	μs
t_{PRT}	$\overline{OE}$ Pulse Rise Time During Programming		50			ns
t_{DF}	Output Enable to Output Float Delay	$\overline{CE} = V_{IL}$	0		130	ns
t_{PW}	Program Pulse Width		0.5	0.5	10	ms
I_{PP}	V_{PP} Supply Current During Programming Pulse	$\overline{CE} = V_{IL}$			30	mA
I_{CC}	V_{CC} Supply Current				10	mA

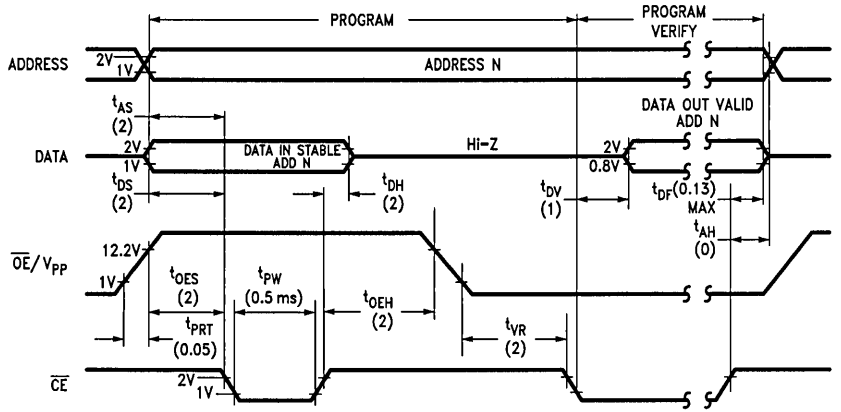
AC Test Conditions

V_{CC}	6V ±0.25V
V_{PP}	12.2 − 13.3V
Input Rise and Fall Times	≤20 ns
Input Pulse Levels	0.45V to 2.4V

Timing Measurement Reference Level

Inputs	1V and 2V
Outputs	0.8V and 2V

Programming Waveforms (Note 3)



TL/D/8827–4

Note: All times shown in parentheses are minimum and in μs unless otherwise specified.

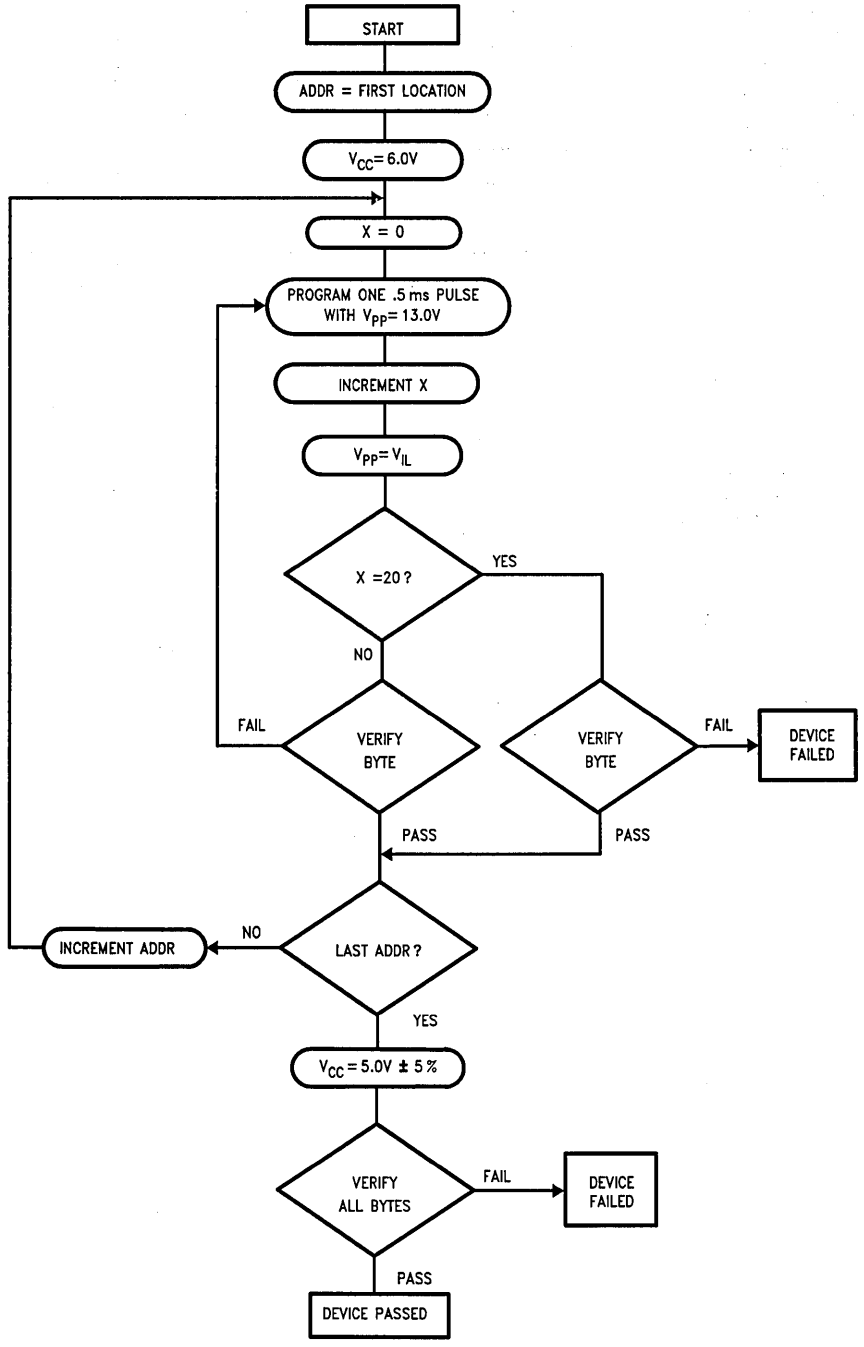
Note 1: National's standard product warranty applies only to devices programmed to specifications described herein.

Note 2: V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}. The NMC27C32B must not be inserted into or removed from a board with voltage applied to V_{PP} or V_{CC}.

Note 3: The maximum allowable voltage which may be applied to the V_{PP} pin during programming is 14V. Care must be taken when switching the V_{PP} supply to prevent overshoot exceeding this 14V maximum specification. At least a 0.1 μF capacitor is required across V_{PP}, V_{CC} to GND to suppress spurious voltage transients which may damage the device.

Note 4: Programming and program verify are tested at nominal power supply voltages.

Interactive Programming Flow Chart



TL/D/8827–5

Functional Description

DEVICE OPERATION

The five modes of operation of the NMC27C32B are listed in Table 1. A single 5V power supply is required in the read mode. All inputs are TTL levels except for $\overline{OE}/V_{PP}$ during programming. In the program mode the $\overline{OE}/V_{PP}$ input is pulsed from a TTL low level to 13V.

Read Mode

The NMC27C32B has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data to the output pins, independent of device selection. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC}-t_{OE}$.

Standby Mode

The NMC27C32B has a standby mode which reduces the active power dissipation by 99%, from 26.3 mW to 0.53 mW. The NMC27C32B is placed in the standby mode by applying a CMOS high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tying

Because EPROMs are usually used in larger memory arrays, National has provided a 2-line control function that accommodates this use of multiple memory connection. The 2-line control function allows for:

a) The lowest possible memory power dissipation, and

b) complete assurance that output bus contention will not occur.

To most efficiently use these two control lines, it is recommended that $\overline{CE}$ (pin 18) be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 20) be made a common connection to all devices in the array and connected to the READ line from the system control bus. This assures that all deselected memory devices are in their low power standby modes and that the output pins are active only when data is desired from a particular memory device.

Programming

CAUTION: Exceeding 14V or pin 1 (V_{PP}) will damage the NMC27C32B.

Initially, and after each erasure, all bits of the NMC27C32B are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The NMC27C32B is in the programming mode when $\overline{OE}/V_{PP}$ is at 13V. It is required that at least a 0.1 μF capacitor be placed across $\overline{OE}/V_{PP}$, V_{CC} and ground to suppress spurious voltage transients which may damage the device. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, an active low, TTL program pulse is applied to the $\overline{CE}$ input. A program pulse must be applied at each address location to be programmed. Any location may be programmed at any time—either individually, sequentially, or at random. The NMC27C32B is designed to be programmed in either of two ways: single pulse programming, where each address is programmed with a 10 ms pulse; or interactive programming, where each address is programmed with a series of 0.5 ms pulses until it verifies (up to a maximum of 20 pulses or 10 ms). Only the interactive programming method has been tested. The NMC27C32B must not be programmed with a DC signal applied to the $\overline{CE}$ input.

Programming multiple NMC27C32Bs in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled NMC27C32B may be connected together when they are programmed with the same data. A low level TTL pulse applied to the $\overline{CE}$ input programs the paralleled NMC27C32B.

TABLE I. Mode Selection

Pins / Mode	$\overline{CE}$ (18)	$\overline{OE}/V_{PP}$ (20)	V_{CC} (24)	Outputs (9-11, 13-17)
Read	V_{IL}	V_{IL}	5	D_{OUT}
Standby	V_{IH}	Don't Care	5	Hi-Z
Program	V_{IL}	13	6	D_{IN}
Program Verify	V_{IL}	V_{IL}	6	D_{OUT}
Program Inhibit	V_{IH}	13	6	Hi-Z

Functional Description (Continued)

Program Inhibit

Programming multiple NMC27C32B in parallel with different data is also easily accomplished. Except for $\overline{CE}$ all like inputs (including $\overline{OE}$) of the parallel NMC27C32B may be common. A TTL low level program pulse applied to an NMC27C32B's $\overline{CE}$ input with $\overline{OE}/V_{PP}$ at 13.0V will program that NMC27C32B. A TTL high level $\overline{CE}$ input inhibits the other NMC27C32B from being programmed.

Program Verify

A verify should be performed on the programmed bits to determine whether they were correctly programmed. The verify is accomplished with $\overline{OE}/V_{PP}$ and $\overline{CE}$ at V_{IL}. Data should be verified t_{DV} after the falling edge of $\overline{CE}$.

MANUFACTURER'S IDENTIFICATION CODE

The NMC27C32B has a manufacturer's identification code to aid in programming. The code, shown in Table II, is two bytes wide and is stored in a ROM configuration on the chip. It identifies the manufacturer and the device type. The code for the NMC27C32B is, "8F61", where "8F" designates that it is made by National Semiconductor, and "61" designates a 32k part.

The code is accessed by applying 12V ±0.5V to address pin A9. Addresses A1–A8, A10–A11, $\overline{CE}$, and $\overline{OE}$ are held at V_{IL}. Address A0 is held at V_{IL} for the manufacturer's code, and at V_{IH} for the device code. The code is read out on the 8 data pins. Proper code access is only guaranteed at 25°C ±5°C.

The primary purpose of the manufacturer's identification code is automatic programming control. When the device is inserted in an EPROM programmer socket, the programmer reads the code and then automatically calls up the specific programming algorithm for the part. This automatic programming control is only possible with programmers which have the capability of reading the code.

ERASURE CHARACTERISTICS

The erasure characteristics of the NMC27C32B are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000Å–4000Å range. Data shows that constant exposure to room-level fluorescent lighting could erase the typical NMC27C32B in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the NMC27C32B is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the NMC27C32B's window to prevent unintentional erasure. Covering the window will also prevent temporary functional failure due to the generation of photo currents.

The recommended erasure procedure for the NMC27C32B is exposure to short wave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity x exposure time) for erasure should be a minimum of 15W-sec/cm^2.

The NMC27C32B should be placed within 1 inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure. Table III shows the minimum NMC27C32B erasure time for various light intensities.

An erasure system should be calibrated periodically. The distance from lamp to unit should be maintained at one inch. The erasure time increases as the square of the distance. (If distance is doubled the erasure time increases by a factor of 4.) Lamps lose intensity as they age. When a lamp is changed, the distance has changed or the lamp has aged, the system should be checked to make certain full erasure is occurring. Incomplete erasure will cause symptoms that can be misleading. Programmers, components, and even system designs have been erroneously suspected when incomplete erasure was the problem.

SYSTEM CONSIDERATION

The power switching characteristics of EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by voltage transitions on input pins. The magnitude of these transient current peaks is dependent on the output capacitance loading of the device. The associated V_{CC} transient voltage peaks can be suppressed by properly selected decoupling capacitors. It is recommended that at least a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance. In addition, at least a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for each eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage drop caused by the inductive effects of the PC board traces.

TABLE II. Manufacturer's Identification Code

Pins	A_0 (8)	O_7 (17)	O_6 (16)	O_5 (15)	O_4 (14)	O_3 (13)	O_2 (11)	O_1 (10)	O_0 (9)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	1	1	1	8F
Device Code	V_{IH}	0	1	1	0	0	0	0	1	61

TABLE III. Minimum NMC27C32B Erasure Time

Light Intensity (Micro-Watts/cm^2)	Erasure Time (Minutes)
15,000	20
10,000	25
5,000	50

National Semiconductor Corporation

PRELIMINARY

microCMOS

NMC27C64
65,536-Bit (8k x 8) UV Erasable CMOS PROM

General Description

The NMC27C64 is a high-speed 64k UV erasable and electrically reprogrammable CMOS EPROM, ideally suited for applications where fast turnaround, pattern experimentation and low power consumption are important requirements.

The NMC27C64 is designed to operate with a single +5V power supply with ±10% tolerance. The CMOS design allows the part to operate over Extended and Military Temperature Ranges.

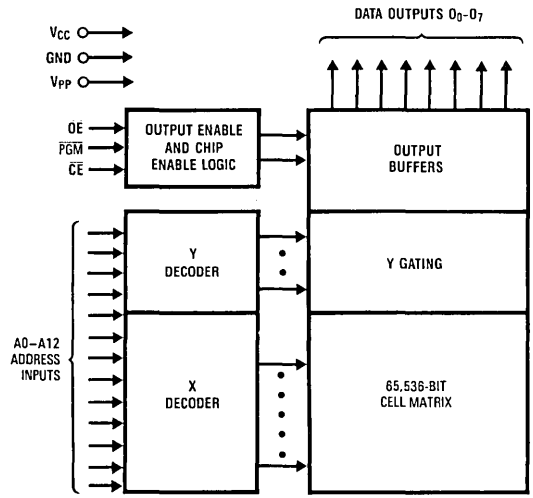
The NMC27C64 is packaged in a 28-pin dual-in-line package with transparent lid. The transparent lid allows the user to expose the chip to ultraviolet light to erase the bit pattern. A new pattern can then be written electrically into the device by following the programming procedure.

This EPROM is fabricated with National's proprietary, time proven microCMOS double-poly silicon gate technology which combines high performance and high density with low power consumption and excellent reliability.

Features

- Access time down to 150 ns, microCMOS technology
- Low CMOS power consumption
  - — Active Power: 55 mW max
  - — Standby Power: 0.55 mW max
- Performance compatible to NSC800™ CMOS microprocessor
- Single 5V power supply
- Extended temperature range (NMC27C64E), −40°C to 85°C, and military temperature range (NMC27C64M), −55°C to 125°C, available
- Pin compatible with NMOS 64k EPROMS
- Fast and reliable programming
- Static operation—no clocks required
- TTL, CMOS compatible inputs/outputs
- TRI-STATE® output

Block Diagram



TL/D/8634-1

Pin Names

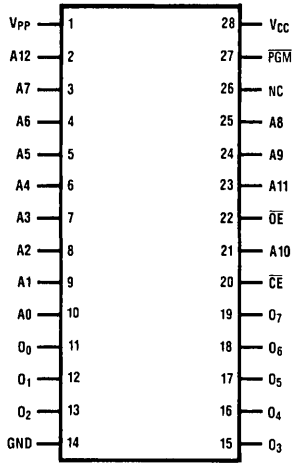
A0–A12	Addresses
$\overline{CE}$	Chip Enable
$\overline{OE}$	Output Enable
O_0–O_7	Outputs
$\overline{PGM}$	Program
NC	No Connect

3

Connection Diagram

27C512 27512	27C256 27256	27C128 27128	27C32 2732	27C16 2716
A15	V_{PP}	V_{PP}		
A12	A12	A12		
A7	A7	A7	A7	A7
A6	A6	A6	A6	A6
A5	A5	A5	A5	A5
A4	A4	A4	A4	A4
A3	A3	A3	A3	A3
A2	A2	A2	A2	A2
A1	A1	A1	A1	A1
A0	A0	A0	A0	A0
O_0	O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2	O_2
GND	GND	GND	GND	GND

Dual-In-Line Package



TL/D/8634–2

27C16 2716	27C32 2732	27C128 27128	27C256 27256	27C512 27512
		V_{CC}	V_{CC}	V_{CC}
		$\overline{PGM}$	A14	A14
V_{CC}	V_{CC}	A13	A13	A13
A8	A8	A8	A8	A8
A9	A9	A9	A9	A9
V_{PP}	A11	A11	A11	A11
$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}/V_{PP}$
A10	A10	A10	A10	A10
$\overline{CE}/\overline{PGM}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}/\overline{PGM}$	$\overline{CE}$
O_7	O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3	O_3

Note: National's socket compatible EPROM pin configurations are shown in the blocks adjacent to the NMC27C64 pins.

Order Number NMC27C64Q
See NS Package Number J28A–Q

Commercial Temp Range (0°C to +70°C)

V_{CC} = 5V ±10%

Parameter/Order Number	Access Time (ns)
NMC27C64Q150	150
NMC27C64Q200	200
NMC27C64Q250	250
NMC27C64Q300	300

Extended Temp Range (−40°C to +85°C)

V_{CC} = 5V ±10%

Parameter/Order Number	Access Time (ns)
NMC27C64QE200	200

Military Temp Range (−55°C to +125°C)

V_{CC} = 5V ±10%

Parameter/Order Number	Access Time (ns)
NMC27C64QM200	200
NMC27C64QM250	250

Absolute Maximum Ratings (Note 1)

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Temperature Under Bias	−10°C to +80°C
Storage Temperature	−65°C to +125°C
All Input Voltages with Respect to Ground	+6.5V to −0.3V
All Output Voltages with Respect to Ground	V_{CC} +0.3V to GND −0.3V
V_{PP} Supply Voltage with Respect to Ground During Programming	+14.0V to −0.3V
Power Dissipation	1.0W
Lead Temp. (Soldering, 10 sec.)	300°C

Operating Conditions (Note 7)

Temperature Range	
NMC27C64Q150, 200, 250, 300	0°C to +70°C
NMC27C64QE200	−40°C to +85°C
NMC27C64QM200, M250	−55°C to +125°C
V_{CC} Power Supply	+5V ±10%

READ OPERATION

DC Operating Characteristics

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
I_{LI}	Input Load Current	$V_{IN} = V_{CC}$ or GND			10	μA
I_{LO}	Output Leakage Current	$V_{OUT} = V_{CC}$ or GND, $\overline{CE} = V_{IH}$			10	μA
I_{CC1} (Note 10)	V_{CC} Current (Active) TTL Inputs	$\overline{CE} = V_{IL}$, f = 5 MHz Inputs = 2.4V or 0.45V, I/O = 0 mA		5	20	mA
I_{CC2} (Note 10)	V_{CC} Current (Active) CMOS Inputs	$\overline{CE}$ = GND, f = 5 MHz Inputs = V_{CC} or GND, I/O = 0 mA		3	10	mA
I_{CCSB1}	V_{CC} Current (Standby) TTL Inputs	$\overline{CE} = V_{IH}$		0.1	1	mA
I_{CCSB2}	V_{CC} Current (Standby) CMOS Inputs	$\overline{CE} = V_{CC}$		0.5	100	μA
V_{IL}	Input Low Voltage		−0.1		0.8	V
V_{IH}	Input High Voltage		2.0		V_{CC} +1	V
V_{OL1}	Output Low Voltage	I_{OL} = 2.1 mA			0.45	V
V_{OH1}	Output High Voltage	I_{OH} = −400 μA	2.4			V
V_{OL2}	Output Low Voltage	I_{OL} = 0 μA			0.1	V
V_{OH2}	Output High Voltage	I_{OH} = 0 μA	V_{CC} −0.1			V

AC Electrical Characteristics

Symbol	Parameter	Conditions	NMC27C64 150		NMC27C64 200 E200 M200		NMC27C64 250 M250		NMC27C64 300		Units
			Min	Max	Min	Max	Min	Max	Min	Max	
t_{ACC}	Address to Output Delay	$\overline{CE} = \overline{OE} = V_{IL}$ $\overline{PGM} = V_{IH}$		150		200		250		300	ns
t_{CE}	$\overline{CE}$ to Output Delay	$\overline{OE} = V_{IL}$ $\overline{PGM} = V_{IH}$		150		200		250		300	ns
t_{OE}	$\overline{OE}$ to Output Delay	$\overline{CE} = V_{IL}$ $\overline{PGM} = V_{IH}$		60		60		70		150	ns
t_{DF}	$\overline{OE}$ High to Output Float	$\overline{CE} = V_{IL}$ $\overline{PGM} = V_{IH}$	0	60	0	60	0	60	0	130	ns
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$, Whichever Occurred First	$\overline{CE} = \overline{OE} = V_{IL}$ $\overline{PGM} = V_{IH}$	0		0		0		0		ns

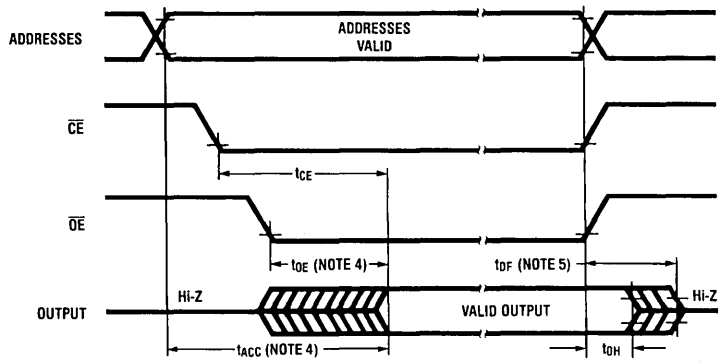
Capacitance $T_A = +25°C$, f = 1 MHz (Note 3)

Symbol	Parameter	Conditions	Typ	Max	Units
C_{IN}	Input Capacitance	$V_{IN} = 0V$	4	6	pF
C_{OUT}	Output Capacitance	$V_{OUT} = 0V$	8	12	pF

AC Test Conditions

Output Load	1 TTL Gate and C_L = 100 pF (Note 9)	Timing Measurement Reference Level	
Input Rise and Fall Times	≤20 ns	Inputs	1V and 2V
Input Pulse Levels	0.45V to 2.4V	Outputs	0.8V and 2V

AC Waveforms



TL/D/8634-3

Note 1: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Note 2: Typical values are for $T_A = +25°C$ and nominal supply voltages.

Note 3: This parameter is only sampled and is not 100% tested.

Note 4: $\overline{OE}$ may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of $\overline{CE}$ without impacting t_{ACC}.

Note 5: The t_{DF} compare level is determined as follows:

High to TRI-STATE, the measured V_{OH1} (DC) −0.10V

Low to TRI-STATE, the measured V_{OL1} (DC) +0.10V

Note 6: TRI-STATE may be attained using $\overline{OE}$ or $\overline{CE}$.

Note 7: The power switching characteristics of EPROMs require careful device decoupling. It is recommended that at least a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND.

Note 8: The outputs must be restricted to V_{CC} +0.3V to avoid latch-up and device damage.

Note 9: 1 TTL Gate: I_{OL} = 1.6 mA, I_{OH} = −400 μA.

CL: 100 pF includes fixture capacitance.

Note 10: V_{PP} may be connected to V_{CC} except during programming.

Programming Characteristics $T_A = +25°C \pm 5°C$, $V_{CC} = 6V \pm 0.25V$ $V_{PP} = 12.2-13.3V$ (Notes 2, 3, & 4)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{AS}	Address Set-Up Time		2			μs
t_{OES}	$\overline{OE}$ Set-Up Time		2			μs
t_{CES}	$\overline{CE}$ Set-Up Time		2			μs
t_{DS}	Data Set-Up Time		2			μs
t_{AH}	Address Hold Time		0			μs
t_{DH}	Data Hold Time		2			μs
t_{DF}	Output Enable to Output Float Delay	$\overline{CE} = V_{IL}$	0		130	ns
t_{OE}	Output Enable to Output Delay	$\overline{CE} = V_{IL}$			150	ns
t_{PW}	Program Pulse Width		0.5	0.5	10	ms
I_{PP}	V_{PP} Supply Current During Programming Pulse	$\overline{CE} = V_{IL}$			30	mA
I_{CC}	V_{CC} Supply Current				10	mA

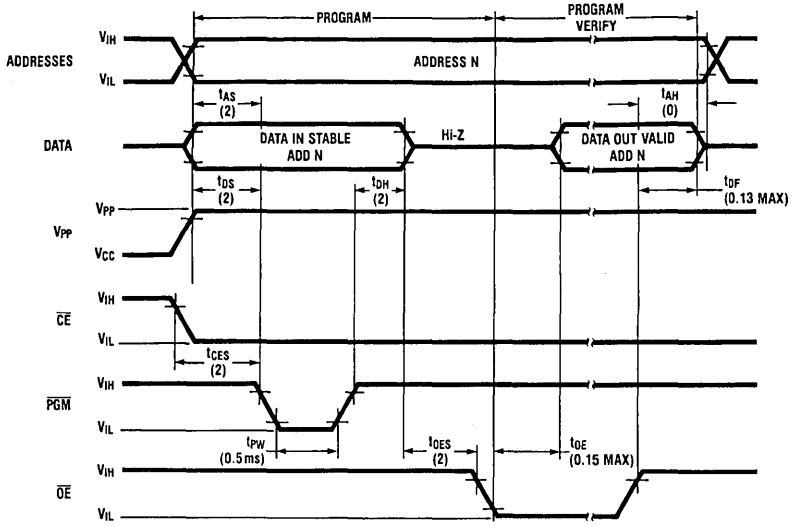
AC Test Conditions

V_{CC}	6V ±0.25V
V_{PP}	12.2−13.3V
Input Rise and Fall Times	≤20 ns
Input Pulse Levels	0.45V to 2.4V

Timing Measurement Reference Level

Inputs	1V and 2V
Outputs	0.8V and 2V

Programming Waveforms (Note 3)



TL/D/8634-4

Note: All times shown in parentheses are minimum and in μs unless otherwise specified.

The input timing reference level is 1V for a V_{IL} and 2V for a V_{IH}.

t_{OE} and t_{DF} are characteristics of the device but must be accommodated by the programmer.

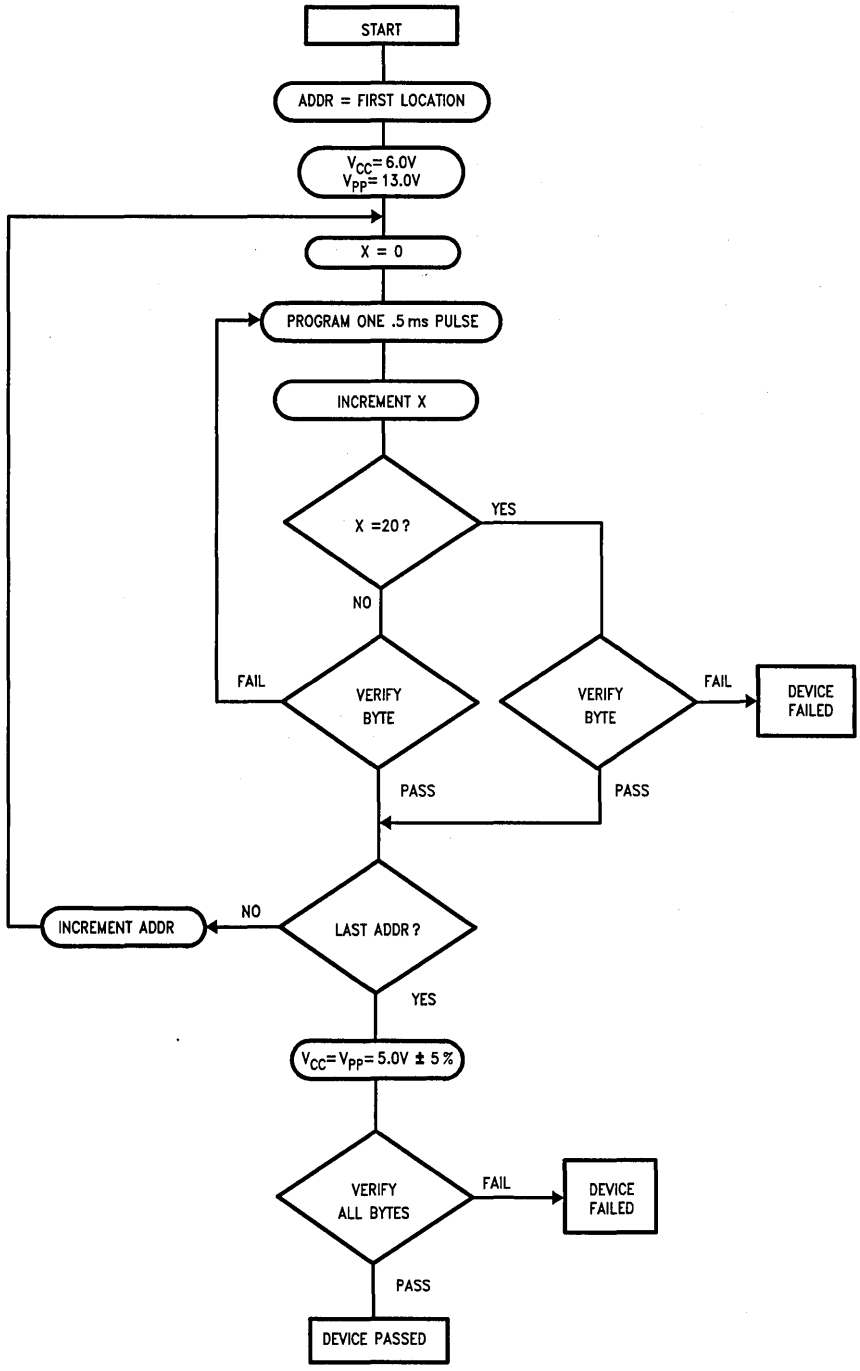
Note 1: National's standard product warranty applies only to devices programmed to specifications described herein.

Note 2: V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}. The NMC27C64 must not be inserted into or removed from a board with voltage applied to V_{PP} or V_{CC}.

Note 3: The maximum allowable voltage which may be applied to the V_{PP} pin during programming is 14V. Care must be taken when switching the V_{PP} supply to prevent overshoot exceeding this 14V maximum specification. At least a 0.1 μF capacitor is required across V_{PP}, V_{CC} to GND to suppress spurious voltage transients which may damage the device.

Note 4: Programming and program verify are tested at nominal power supply voltages.

INTERACTIVE PROGRAMMING FLOW CHART



TL/D/8634–5

Functional Description

DEVICE OPERATION

The five modes of operation of the NMC27C64 are listed in Table I. It should be noted that all inputs for the five modes are at TTL levels. The power supplies required are V_{CC} and V_{PP}. The V_{PP} power supply must be at 13.0V during the three programming modes, and must be at 5V in the other two modes. The V_{CC} power supply must be at 6V during the three programming modes, and at 5V in the other two modes.

Read Mode

The NMC27C64 has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data to the output pins, independent of device selection. The programming pin ($\overline{PGM}$) should be at V_{IH} except during programming. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs t_{OE} after the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC}-t_{OE}$.

Standby Mode

The NMC27C64 has a standby mode which reduces the active power dissipation by 99%, from 55 mW to 0.55 mW. The NMC27C64 is placed in the standby mode by applying a CMOS high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tying

Because NMC27C64s are usually used in larger memory arrays, National has provided a 2-line control function that accommodates this use of multiple memory connections. The 2-line control function allows for:

a) the lowest possible memory power dissipation, and

b) complete assurance that output bus contention will not occur.

To most efficiently use these two control lines, it is recommended that $\overline{CE}$ (pin 20) be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 22) be made a common connection to all devices in the array and connected to the READ line from the system control bus. This assures that all deselected memory devices are in their low power standby modes and that the output pins are active only when data is desired from a particular memory device.

Programming

CAUTION: Exceeding 14V on pin 1 (V_{PP}) will damage the NMC27C64.

Initially, and after each erasure, all bits of the NMC27C64 are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The NMC27C64 is in the programming mode when the V_{PP} power supply is at 13.0V and $\overline{OE}$ is at V_{IH}. It is required that at least a 0.1 μF capacitor be placed across V_{PP}, V_{CC} to ground to suppress spurious voltage transients which may damage the device. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

For programming, $\overline{CE}$ should be kept TTL low at all times while V_{PP} is kept at 13.0V

When the address and data are stable, an active low, TTL program pulse is applied to the $\overline{PGM}$ input. A program pulse must be applied at each address location to be programmed. You can program any location at any time—either individually, sequentially, or at random. The NMC27C64 is designed to be programmed in either of two ways: single pulse programming, where each address is programmed with a 10 ms pulse; or interactive programming, where each address is programmed with a series of 0.5 ms pulses until it verifies (up to a maximum of 20 pulses or 10 ms). Only the interactive programming method has been tested. The NMC27C64 must not be programmed with a DC signal applied to the $\overline{PGM}$ input.

Programming multiple NMC27C64s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled NMC27C64s may be connected together when they are programmed with the same data. A low level TTL pulse applied to the $\overline{PGM}$ input programs the paralleled NMC27C64s.

TABLE I. Mode Selection

Mode \ Pins	$\overline{CE}$ (20)	$\overline{OE}$ (22)	$\overline{PGM}$ (27)	V_{PP} (1)	V_{CC} (28)	Outputs (11–13, 15–19)
Read	V_{IL}	V_{IL}	V_{IH}	5	5	D_{OUT}
Standby	V_{IH}	X	X	5	5	Hi-Z
Program	V_{IL}	V_{IH}	Pulsed V_{IH} to V_{IL}	13.0	6	D_{IN}
Program Verify	V_{IL}	V_{IL}	V_{IH}	13.0	6	D_{OUT}
Program Inhibit	V_{IH}	X	X	13.0	6	Hi-Z

X can be either V_{IL} or V_{IH}

Functional Description (Continued)

Program Inhibit

Programming multiple NMC27C64s in parallel with different data is also easily accomplished. Except for $\overline{CE}$ all like inputs (including $\overline{OE}$) of the parallel NMC27C64 may be common. A TTL low level program pulse applied to an NMC27C64's $\overline{CE}$ input with V_{PP} at 13.0V will program that NMC27C64. A TTL high level $\overline{CE}$ input inhibits the other NMC27C64 from being programmed.

Program Verify

A verify should be performed on the programmed bits to determine whether they were correctly programmed. The verify may be performed with V_{PP} at 13.0V. Except during programming and program verify, V_{PP} must be at V_{CC}.

MANUFACTURER'S IDENTIFICATION CODE

The NMC27C64 has a manufacturer's identification code to aid in programming. The code, shown in Table II, is two bytes wide and is stored in a ROM configuration on the chip. It identifies the manufacturer and the device type. The code for the NMC27C64 is "8FC2", where "8F" designates that it is made by National Semiconductor, and "C2" designates a 64k part.

The code is accessed by applying 12V ± 0.5V to address pin A9. Addresses A1–A8, A10–A12, $\overline{CE}$, and $\overline{OE}$ are held at V_{IL}. Address A0 is held at V_{IL} for the manufacturer's code, and at V_{IH} for the device code. The code is read out on the 8 data pins. Proper code access is only guaranteed at 25°C ± 5°C.

The primary purpose of the manufacturer's identification code is automatic programming control. When the device is inserted in a EPROM programmer socket, the programmer reads the code and then automatically calls up the specific programming algorithm for the part. This automatic programming control is only possible with programmers which have the capability of reading the code.

ERASURE CHARACTERISTICS

The erasure characteristics of the NMC27C64 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000Å–4000Å range. Data shows that constant exposure to room-level fluorescent lighting could erase the typical NMC27C64 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the NMC27C64 is to be exposed to these types of lighting conditions for extended periods of time, opaque lables should be placed over the NMC27C64's window to prevent unintentional erasure. Covering the window will also prevent temporary functional failure due to the generation of photo currents.

The recommended erasure procedure for the NMC27C64 is exposure to short wave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity x exposure time) for erasure should be a minimum of 15W-sec/cm^2.

The NMC27C64 should be placed within 1 inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure. Table III shows the minimum NMC27C64 erasure time for various light intensities.

An erasure system should be calibrated periodically. The distance from lamp to unit should be maintained at one inch. The erasure time increases as the square of the distance. (If distance is doubled the erasure time increases by a factor of 4.) Lamps lose intensity as they age. When a lamp is changed, the distance has changed or the lamp has aged, the system should be checked to make certain full erasure is occurring. Incomplete erasure will cause symptoms that can be misleading. Programmers, components, and even system designs have been erroneously suspected when incomplete erasure was the problem.

SYSTEM CONSIDERATION

The power switching characteristics of EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by voltage transitions on input pins. The magnitude of these transient current peaks is dependent on the output capacitance loading of the device. The associated V_{CC} transient voltage peaks can be suppressed by properly selected decoupling capacitors. It is recommended that at least a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance. In addition, at least a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for each eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage drop caused by the inductive effects of the PC board traces.

TABLE II. Manufacturer's Identification Code

Pins	A_0 (10)	O_7 (19)	O_6 (18)	O_5 (17)	O_4 (16)	O_3 (15)	O_2 (13)	O_1 (12)	O_0 (11)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	1	1	1	8F
Device Code	V_{IH}	1	1	0	0	0	0	1	0	C2

TABLE III. Minimum NMC27C64 Erasure Time

Light Intensity (Micro-Watts/cm^2)	Erasure Time (Minutes)
15,000	20
10,000	25
5,000	50

National Semiconductor Corporation

PRELIMINARY

microCMOS

NMC27C256
262,144-Bit (32k x 8) UV Erasable CMOS PROM

General Description

The NMC27C256 is a high-speed 256k UV erasable and electrically reprogrammable CMOS EPROM, ideally suited for applications where fast turnaround, pattern experimentation and low power consumption are important requirements.

The NMC27C256 is designed to operate with a single +5V power supply with ±5% or ±10% tolerance. The CMOS design allows the part to operate over Extended and Military Temperature Ranges.

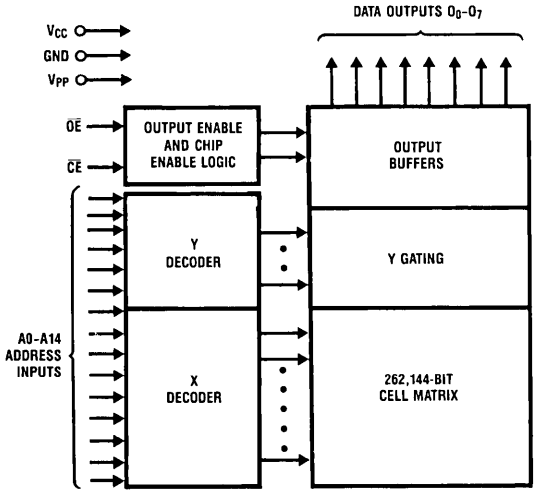
The NMC27C256 is packaged in a 28-pin dual in-line package with transparent lid. The transparent lid allows the user to expose the chip to ultraviolet light to erase the bit pattern. A new pattern can then be written electrically into the device by following the programming procedure.

This EPROM is fabricated with National's proprietary, time proven microCMOS double-poly silicon gate technology which combines high performance and high density with low power consumption and excellent reliability.

Features

- Access time down to 170 ns, microCMOS technology
- Low CMOS power consumption
  - — Active Power: 55 mW max
  - — Standby Power: 0.55 mW max
- Performance compatible to NSC800™ CMOS microprocessor
- Single 5V power supply
- Extended temperature range (NMC27C256E), −40°C to 85°C, and military temperature range (NMC27C256M), −55°C to 125°C, available
- Pin compatible with NMOS 256k EPROMS
- Fast and reliable programming (0.5 ms for most bytes)
- Static operation—no clocks required
- TTL, CMOS compatible inputs/outputs
- TRI-STATE® output

Block Diagram



TL/D/7512-1

Pin Names

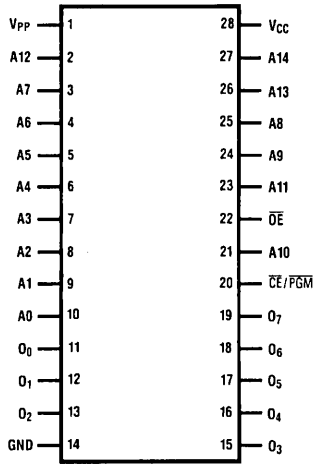
A0–A14	Addresses
$\overline{CE}$	Chip Enable
$\overline{OE}$	Output Enable
O_0–O_7	Outputs
$\overline{PGM}$	Program
NC	No Connect

3

Connection Diagram

27C512 27512	27C128 27128	27C64 2764	27C32 2732	27C16 2716
A15	V_{PP}	V_{PP}		
A12	A12	A12		
A7	A7	A7	A7	A7
A6	A6	A6	A6	A6
A5	A5	A5	A5	A5
A4	A4	A4	A4	A4
A3	A3	A3	A3	A3
A2	A2	A2	A2	A2
A1	A1	A1	A1	A1
A0	A0	A0	A0	A0
O_0	O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2	O_2
GND	GND	GND	GND	GND

Dual-In-Line Package



TL/D/7512-2

27C16 2716	27C32 2732	27C64 2764	27C128 27128	27C512 27512
		V_{CC}	V_{CC}	V_{CC}
		$\overline{PGM}$	$\overline{PGM}$	A14
V_{CC}	V_{CC}	NC	A13	A13
A8	A8	A8	A8	A8
A9	A9	A9	A9	A9
V_{PP}	A11	A11	A11	A11
$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}/V_{PP}$
A10	A10	A10	A10	A10
$\overline{CE}/\overline{PGM}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3	O_3

Note: National's socket compatible EPROM pin configurations are shown in the blocks adjacent to the NMC27C256 pins.

Order Number NMC27C256Q
See NS Package Number J28A-Q

Commercial Temp Range (0°C to +70°C)

V_{CC} = 5V ±5%

Parameter/Order Number	Access Time
NMC27C256Q17	170

V_{CC} = 5V±10%

Parameter/Order Number	Access Time
NMC27C256Q200	200
NMC27C256Q250	250
NMC27C256Q300	300

Extended Temp Range (−40°C to +85°C)

V_{CC} = 5V ±10%

Parameter/Order Number	Access Time
NMC27C256QE250	250

Military Temp Range (−55°C to +125°C)

V_{CC} = 5V±10%

Parameter/Order Number	Access Time
NMC27C256QM250	250
NMC27C256QM350	350

COMMERCIAL TEMPERATURE RANGE

Absolute Maximum Ratings (Note 1)

Temperature Under Bias	−10°C to +80°C
Storage Temperature	−65°C to +125°C
All Input Voltages with Respect to Ground	+6.5V to −0.3V
All Output Voltages with Respect to Ground	V_{CC} +0.3 to GND −0.3V
V_{PP} Supply Voltage with Respect to Ground During Programming	+14.0V to −0.3V
Power Dissipation	1.0W
Lead Temperature (Soldering, 10 sec.)	300°C

Operating Conditions (Note 7)

Temperature Range	0°C to +70°C
V_{CC} Power Supply	
NMC27C256Q17	5V ± 5%
NMC27C256Q200, 250, 300	5V ± 10%

READ OPERATION

DC Electrical Characteristics

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
I_{LI}	Input Load Current	$V_{IN} = V_{CC}$ or GND			10	μA
I_{LO}	Output Leakage Current	$V_{OUT} = V_{CC}$ or GND, $\overline{CE} = V_{IH}$			10	μA
I_{CC1} (Note 10)	V_{CC} Current (Active) TTL Inputs	$\overline{CE} = V_{IL}$, f = 5 MHz Inputs = V_{IH} or V_{IL}, I/O = 0 mA		5	20	mA
I_{CC2} (Note 10)	V_{CC} Current (Active) CMOS Inputs	$\overline{CE}$ = GND, f = 5 MHz Inputs = V_{CC} or GND, I/O = 0 mA		3	10	mA
I_{CCSB1}	V_{CC} Current (Standby) TTL Inputs	$\overline{CE} = V_{IH}$		0.1	1	mA
I_{CCSB2}	V_{CC} Current (Standby) CMOS Inputs	$\overline{CE} = V_{CC}$		0.5	100	μA
V_{IL}	Input Low Voltage		−0.1		0.8	V
V_{IH}	Input High Voltage		2.0		V_{CC} +1	V
V_{OL1}	Output Low Voltage	I_{OL} = 2.1 mA			0.45	V
V_{OH1}	Output High Voltage	I_{OH} = −400 μA	2.4			V
V_{OL2}	Output Low Voltage	I_{OL} = 0 μA			0.1	V
V_{OH2}	Output High Voltage	I_{OH} = 0 μA	V_{CC} −0.1			V

AC Electrical Characteristics (Note 11)

Symbol	Parameter	Conditions	NMC27C256 17		NMC27C256 200		NMC27C256 250		NMC27C256 300		Units
			Min	Max	Min	Max	Min	Max	Min	Max	
t_{ACC}	Address to Output Delay	$\overline{CE} = \overline{OE} = V_{IL}$		170		200		250		300	ns
t_{CE}	$\overline{CE}$ to Output Delay	$\overline{OE} = V_{IL}$		170		200		250		300	ns
t_{OE}	$\overline{OE}$ to Output Delay	$\overline{CE} = V_{IL}$		75		75		100		120	ns
t_{DF}	$\overline{OE}$ High to Output Float	$\overline{CE} = V_{IL}$	0	60	0	60	0	60		105	ns
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$, Whichever Occurred First	$\overline{CE} = \overline{OE} = V_{IL}$	0		0		0		0		ns

MILITARY AND EXTENDED TEMPERATURE RANGE

Absolute Maximum Ratings (Note 1)

Temperature Under Bias	Operating Temp Range
Storage Temperature	−65°C to +125°C
All Input Voltages with Respect to Ground	+6.5V to −0.3V
All Output Voltages with Respect to Ground	V_{CC} +0.3V to GND −0.3V
V_{PP} Supply Voltage with Respect to Ground During Programming	+14.0V to −0.3V
Power Dissipation	1.0W
Lead Temperature (Soldering, 10 sec.)	300°C

Operating Conditions (Note 7)

Temperature Range	
NMC27C256QE250	−40°C to +85°C
NMC27C256QM250, M350	−55°C to +125°C
V_{CC} Power Supply	5V ± 10%

READ OPERATION

DC Operating Characteristics

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
I_{LI}	Input Load Current	$V_{IN} = V_{CC}$ or GND			10	μA
I_{LO}	Output Leakage Current	$V_{OUT} = V_{CC}$ or GND, $\overline{CE} = V_{IH}$			10	μA
I_{CC1} (Note 10)	V_{CC} Current (Active) TTL Inputs	$\overline{CE} = V_{IL}$, f = 5 MHz Inputs = V_{IH} or V_{IL} I/O = 0 mA		5	20	mA
I_{CC2} (Note 10)	V_{CC} Current (Active) CMOS Inputs	$\overline{CE}$ = GND, f = 5 MHZ Inputs = V_{CC} or GND I/O = 0 mA		3	10	mA
I_{CCSB1}	V_{CC} Current (Standby) TTL Inputs	$\overline{CE} = V_{IH}$		0.1	1	mA
I_{CCSB2}	V_{CC} Current (Standby) CMOS Inputs	$\overline{CE} = V_{CC}$		0.5	100	μA
V_{IL}	Input Low Voltage		−0.1		0.8	V
V_{IH}	Input High Voltage		2.0		V_{CC} + 1	V
V_{OL1}	Output Low Voltage	I_{OL} = 2.1 mA			0.45	V
V_{OH1}	Output High Voltage	I_{OH} = −400 μA	2.4			V
V_{OL2}	Output Low Voltage	I_{OH} = 0 μA			0.1	V
V_{OH2}	Output High Voltage	I_{OH} = 0 μA	V_{CC} −0.1			V

AC Electrical Characteristics

Symbol	Parameter	Conditions	NMC27C256 E250 M250		NMC27C256 M350		Units
			Min	Max	Min	Max	
t_{ACC}	Address to Output Delay	$\overline{CE} = \overline{OE} = V_{IL}$		250		350	ns
t_{CE}	$\overline{CE}$ to Output Delay	$\overline{OE} = V_{IL}$		250		350	ns
t_{OE}	$\overline{OE}$ to Output Delay	$\overline{CE} = V_{IL}$		100		120	ns
t_{DF}	$\overline{OE}$ High to Output Float	$\overline{CE} = V_{IL}$	0	60	0	105	ns
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$, Whichever Occurred First	$\overline{CE} = \overline{OE} = V_{IL}$	0		0		ns

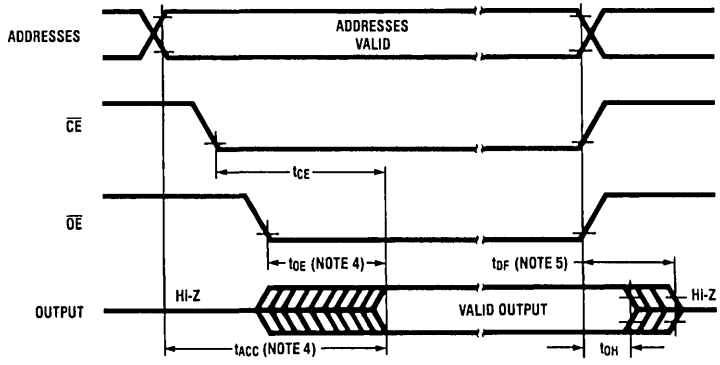
Capacitance T_A = +25°C, f = 1 MHz (Note 3)

Symbol	Parameter	Conditions	Typ	Max	Units
C_{IN}	Input Capacitance	V_{IN} = 0V	4	6	pF
C_{OUT}	Output Capacitance	V_{OUT} = 0V	8	12	pF

AC Test Conditions

Output Load	1 TTL Gate and C_L = 100 pF (Note 9)	Timing Measurement Reference Level	
Input Rise and Fall Times	≤20 ns	Inputs	1V and 2V
Input Pulse Levels	0.45V to 2.4V	Outputs	0.8V and 2V

AC Waveforms



TL/D/7512-3

Note 1: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Note 2: Typical values are for T_A = +25°C and nominal supply voltages.

Note 3: This parameter is only sampled and is not 100% tested.

Note 4: $\overline{OE}$ may be delayed up to $t_{ACC}-t_{OE}$ after the falling edge of $\overline{CE}$ without impacting t_{ACC}.

Note 5: The t_{DF} compare level is determined as follows:

High to TRI-STATE, the measured V_{OH1} (DC) −0.10V

Low to TRI-STATE, the measured V_{OL1} (DC) +0.10V

Note 6: TRI-STATE may be attained using $\overline{OE}$ or $\overline{CE}$.

Note 7: The power switching characteristics of EPROMs require careful device decoupling. It is recommended that at least a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND.

Note 8: The outputs must be restricted to V_{CC} +0.3V to avoid latch-up and device damage.

Note 9: 1 TTL Gate: I_{OL} = 1.6 mA, I_{OH} = −400 μA.

CL: 100 pF includes fixture capacitance.

Note 10: V_{PP} may be connected to V_{CC} except during programming.

Note 11: All parameters in nanoseconds.

Programming Characteristics

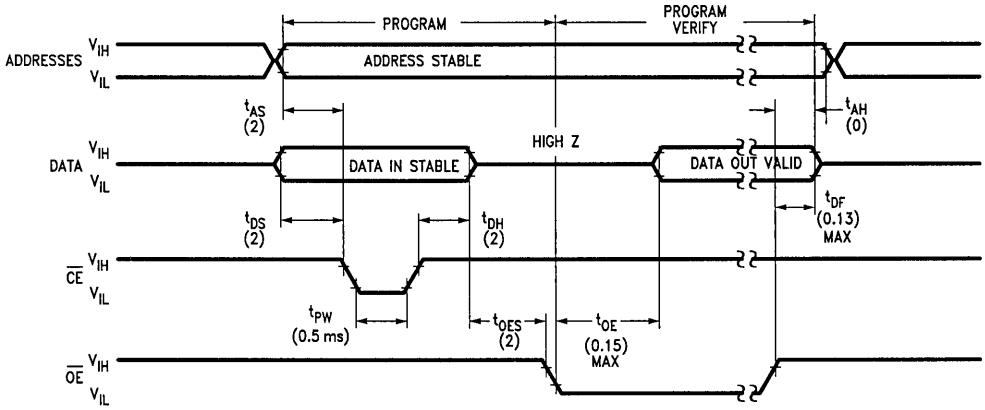
T_A = +25°C ±5°C, V_{CC} = 6V ±0.25V, V_{PP} = 12.2V to 13.3V (Notes 2, 3, & 4)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{AS}	Address Set-Up Time		2			μs
t_{OES}	$\overline{OE}$ Set-Up Time		2			μs
t_{DS}	Data Set-Up Time		2			μs
t_{AH}	Address Hold Time		0			μs
t_{DH}	Data Hold Time		2			μs
t_{DF}	Output Enable to Output Float Delay	$\overline{CE}/\overline{PGM} = V_{IL}$	0		130	ns
t_{OE}	Output Enable to Output Delay	$\overline{CE}/\overline{PGM} = V_{IL}$			150	ns
t_{PW}	Program Pulse Width		0.5	0.5	10	ms
I_{PP}	V_{PP} Supply Current During Programming Pulse	$\overline{CE} = V_{IL}$			30	mA
I_{CC}	V_{CC} Supply Current				10	mA

AC Test Conditions

V_{CC}	6V ±0.25V	Timing Measurement Reference Level	
V_{PP}	12.2V to 13.3V	Inputs	1V and 2V
Input Rise and Fall Times	≤20 ns	Outputs	0.8V and 2V
Input Pulse Levels	0.45V to 2.4V		

Programming Waveforms (Note 3)



TL/D/7512–4

Note: All times shown in parentheses are minimum and in μs unless otherwise specified.

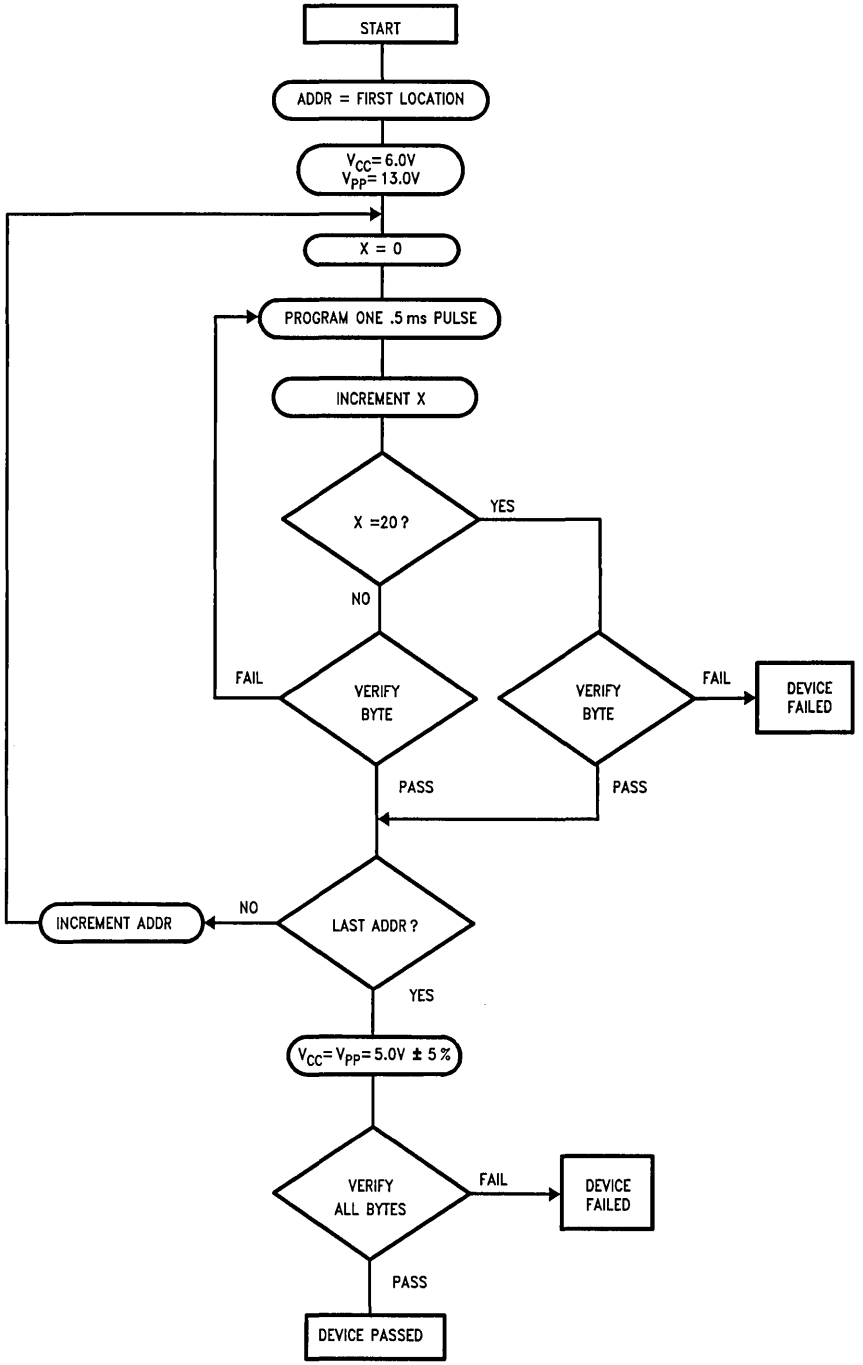
Note 1: National's standard product warranty applies only to devices programmed to specifications described herein.

Note 2: V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}. The NMC27C256 must not be inserted into or removed from a board with voltage applied to V_{PP} or V_{CC}.

Note 3: The maximum allowable voltage which may be applied to the V_{PP} pin during programming is 14V. Care must be taken when switching the V_{PP} supply to prevent overshoot exceeding this 14V maximum specification. At least a 0.1 μF capacitor is required across V_{PP}, V_{CC} to GND to suppress spurious voltage transients which may damage the device.

Note 4: Programming and program verify are tested at nominal power supply voltages.

INTERACTIVE PROGRAMMING FLOW CHART



TL/D/7512–5

Functional Description

DEVICE OPERATION

The five modes of operation of the NMC27C256 are listed in Table I. It should be noted that all inputs for the five modes are at TTL levels. The power supplies required are V_{CC} and V_{PP}. The V_{PP} power supply must be at 13.0V during the three programming modes, and must be at 5V in the other two modes. The V_{CC} power supply must be at 6V during the three programming modes, and at 5V in the other two modes.

Read Mode

The NMC27C256 has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data to the output pins, independent of device selection. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs t_{OE} after the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC}-t_{OE}$.

Standby Mode

The NMC27C256 has a standby mode which reduces the active power dissipation by 99%, from 55 mW to 0.55 mW. The NMC27C256 is placed in the standby mode by applying a CMOS high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tying

Because NMC27C256s are usually used in larger memory arrays, National has provided a 2-line control function that accommodates this use of multiple memory connections. The 2-line control function allows for:

a) the lowest possible memory power dissipation, and

b) complete assurance that output bus contention will not occur.

To most efficiently use these two control lines, it is recommended that $\overline{CE}$ (pin 20) be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 22) be made a common connection to all devices in the array and connected to the READ line from the system control bus. This assures that all deselected memory devices are in their low power standby modes and that the output pins are active only when data is desired from a particular memory device.

Programming

CAUTION: Exceeding 14V on pin 1 (V_{PP}) will damage the NMC27C256.

Initially, and after each erasure, all bits of the NMC27C256 are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The NMC27C256 is in the programming mode when the V_{PP} power supply is at 13.0V and $\overline{OE}$ is at V_{IH}. It is required that at least a 0.1 μF capacitor be placed across V_{PP}, V_{CC} to ground to suppress spurious voltage transients which may damage the device. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, an active low, TTL program pulse is applied to the $\overline{CE}/\overline{PGM}$ input. A program pulse must be applied at each address location to be programmed. You can program any location at any time—either individually, sequentially, or at random. The NMC27C256 is designed to be programmed in either of two ways: single pulse programming, where each address is programmed with a 10 ms pulse; or interactive programming, where each address is programmed with a series of 0.5 ms pulses until it verifies (up to a maximum of 20 pulses or 10 ms). Only the interactive programming method has been tested. The NMC27C256 must not be programmed with a DC signal applied to the $\overline{CE}/\overline{PGM}$ input.

Programming multiple NMC27C256s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled NMC27C256s may be connected together when they are programmed with the same data. A low level TTL pulse applied to the $\overline{CE}/\overline{PGM}$ input programs the paralleled NMC27C256s.

TABLE I. Mode Selection

Mode \ Pins	$\overline{CE}/\overline{PGM}$ (20)	$\overline{OE}$ (22)	V_{PP} (1)	V_{CC} (28)	Outputs (11–13, 15–19)
Read	V_{IL}	V_{IL}	V_{CC}	5	D_{OUT}
Standby	V_{IH}	Don't Care	V_{CC}	5	Hi-Z
Program	Pulsed V_{IH} to V_{IL}	V_{IH}	13.0	6	D_{IN}
Program Verify	V_{IH}	V_{IL}	13.0	6	D_{OUT}
Program Inhibit	V_{IH}	V_{IH}	13.0	6	Hi-Z

Functional Description (Continued)

Program Inhibit

Programming multiple NMC27C256s in parallel with different data is also easily accomplished. Except for $\overline{CE}$ all like inputs (including $\overline{OE}$) of the parallel NMC27C256s may be common. A TTL low level program pulse applied to an NMC27C256's $\overline{CE}/\overline{PGM}$ input with V_{PP} at 13.0V will program that NMC27C256. A TTL high level $\overline{CE}$ input inhibits the other NMC27C256s from being programmed.

Program Verify

A verify should be performed on the programmed bits to determine whether they were correctly programmed. The verify may be performed with V_{PP} at 13.0V. Except during programming and program verify, V_{PP} must be at V_{CC}.

ERASURE CHARACTERISTICS

The erasure characteristics of the NMC27C256 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000Å–4000Å range. Data shows that constant exposure to room-level fluorescent lighting could erase the typical NMC27C256 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the NMC27C256 is to be exposed to these types of lighting conditions for extended periods of time, opaque lables should be placed over the NMC27C256's window to prevent unintentional erasure. Covering the window will also prevent temporary functional failure due to the generation of photo currents.

The recommended erasure procedure for the NMC27C256 is exposure to short wave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity x exposure time) for erasure should be a minimum of 15W-sec/cm^2.

The NMC27C256 should be placed within 1 inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure. Table II shows the minimum NMC27C256 erasure time for various light intensities.

An erasure system should be calibrated periodically. The distance from lamp to unit should be maintained at one inch. The erasure time increases as the square of the distance. (If distance is doubled the erasure time increases by a factor of 4.) Lamps lose intensity as they age. When a lamp is changed, the distance has changed or the lamp has aged, the system should be checked to make certain full erasure is occurring. Incomplete erasure will cause symptoms that can be misleading. Programmers, components, and even system designs have been erroneously suspected when incomplete erasure was the problem.

SYSTEM CONSIDERATION

The power switching characteristics of EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by voltage transitions on input pins. The magnitude of these transient current peaks is dependent on the output capacitance loading of the device. The associated V_{CC} transient voltage peaks can be suppressed by properly selected decoupling capacitors. It is recommended that at least a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance. In addition, at least a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for each eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage drop caused by the inductive effects of the PC board traces.

TABLE II. Minimum NMC27C256 Erasure Time

Light Intensity (Micro-Watts/cm^2)	Erasure Time (Minutes)
15,000	20
10,000	25
5,000	50

National Semiconductor Corporation

PRELIMINARY

microCMOS

NMC27C512
524,288-Bit (64k x 8) UV Erasable CMOS PROM

General Description

The NMC27C512 is a high-speed 512k UV erasable and electrically reprogrammable CMOS EPROM, ideally suited for applications where fast turnaround, pattern experimentation and low power consumption are important requirements.

The NMC27C512 is designed to operate with a single +5V power supply with ±10% tolerance. The CMOS design allows the part to operate over Extended and Military Temperature Ranges.

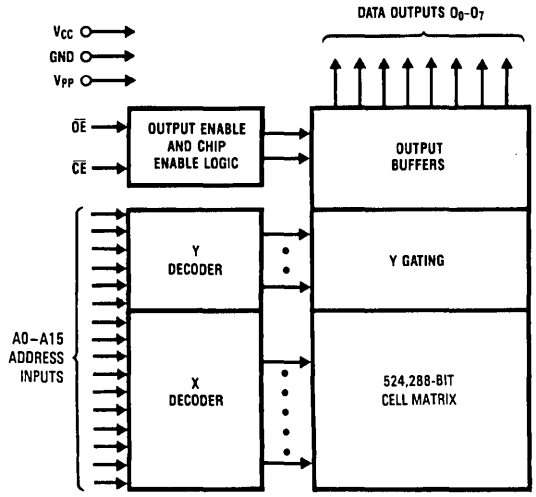
The NMC27C512 is packaged in a 28-pin dual in-line package with transparent lid. The transparent lid allows the user to expose the chip to ultraviolet light to erase the bit pattern. A new pattern can then be written electrically into the device by following the programming procedure.

This EPROM is fabricated with National's proprietary, time proven microCMOS double-poly silicon gate technology which combines high performance and high density with low power consumption and excellent reliability.

Features

- Access time down to 200 ns, microCMOS technology
- Low CMOS power consumption
  - — Active Power: 55 mW max
  - — Standby Power: 0.55 mW max
- Performance compatible to NSC800™ CMOS microprocessor
- Single 5V power supply
- Extended temperature range (NMC27C512E), −40°C to 85°C, and military temperature range (NMC27C512M), −55°C to 125°C, available
- Pin compatible with NMOS 512k EPROMS
- Fast and reliable programming (0.5 ms for most bytes)
- Static operation—no clocks required
- TTL, CMOS compatible inputs/outputs
- TRI-STATE® output

Block Diagram



TL/D/8754-1

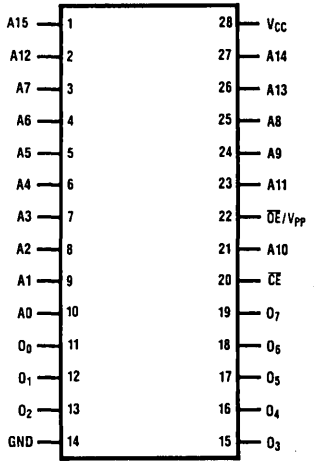
Pin Names

A0–A15	Addresses
$\overline{CE}$	Chip Enable
$\overline{OE}$	Output Enable
O_0–O_7	Outputs
$\overline{PGM}$	Program
NC	No Connect

Connection Diagram

27C256 27256	27C128 27128	27C64 2764	27C32 2732	27C16 2716
V_{PP}	V_{PP}	V_{PP}		
A12	A12	A12		
A7	A7	A7	A7	A7
A6	A6	A6	A6	A6
A5	A5	A5	A5	A5
A4	A4	A4	A4	A4
A3	A3	A3	A3	A3
A2	A2	A2	A2	A2
A1	A1	A1	A1	A1
A0	A0	A0	A0	A0
O_0	O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2	O_2
GND	GND	GND	GND	GND

Dual-In-Line Package



TL/D/8754–2

27C16 2716	27C32 2732	27C64 2764	27C128 27128	27C256 27256
		V_{CC}	V_{CC}	V_{CC}
		$\overline{PGM}$	$\overline{PGM}$	A14
V_{CC}	V_{CC}	NC	A13	A13
A8	A8	A8	A8	A8
A9	A9	A9	A9	A9
V_{PP}	A11	A11	A11	A11
$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$
A10	A10	A10	A10	A10
$\overline{CE}$/PGM	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$/$\overline{PGM}$
O_7	O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3	O_3

Note: National's socket compatible EPROM pin configurations are shown in the blocks adjacent to the NMC27C512 pins.

See NS Package Number J28A–Q

Commercial Temp Range (0°C to +70°C)

V_{CC} = 5V ±5%

Parameter/Order Number	Access Time
NMC27C512Q20	200

V_{CC} = 5V ±10%

Parameter/Order Number	Access Time
NMC27C512Q250	250
NMC27C512Q300	300
NMC27C512Q350	350

Extended Temp Range (−40°C to +85°C)

V_{CC} = 5V ±10%

Parameter/Order Number	Access Time
NMC27C512QE250	250

Military Temp Range (−55°C to +125°C)

V_{CC} = 5V ±10%

Parameter/Order Number	Access Time
NMC27C512QM350	350

Absolute Maximum Ratings (Note 1)

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Temperature Under Bias	Operating Temp Range
Storage Temperature	−65°C to +125°C
All Input Voltages with Respect to Ground	+6.5V to −0.3V
All Output Voltages with Respect to Ground	V_{CC} +0.3 to GND −0.3V
V_{PP} and A9 Supply Voltage with Respect to Ground	+14.0V to −0.3V
Power Dissipation	1.0W
Lead Temperature (Soldering, 10 sec.)	300°C

Operating Conditions (Note 7)

Temperature Range	
NMC27C512Q20,250,300,350	0°C to 70°C
NMC27C512QE250	−40°C to +85°C
NMC27C512QM350	−55°C to +125°C
V_{CC} Power Supply	5V ±10%
except NMC27C572Q20	5V ±5%

ESD rating is to be determined.

READ OPERATION

DC Electrical Characteristics

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
I_{LI}	Input Load Current	$V_{IN} = V_{CC}$ or GND			10	μA
I_{LO}	Output Leakage Current	$V_{OUT} = V_{CC}$ or GND, $\overline{CE} = V_{IH}$			10	μA
I_{CC1}	V_{CC} Current (Active) TTL Inputs	$\overline{CE} = V_{IL}$, f = 5 MHz Inputs = V_{IH} or V_{IL} I/O = 0 mA		5	20	mA
I_{CC2}	V_{CC} Current (Active) CMOS Inputs	$\overline{CE}$ = GND, f = 5 MHz Inputs = V_{CC} or GND, I/O = 0 mA		3	10	mA
I_{CCSB1}	V_{CC} Current (Standby) TTL Inputs	$\overline{CE} = V_{IH}$		0.1	1	mA
I_{CCSB2}	V_{CC} Current (Standby) CMOS Inputs	$\overline{CE} = V_{CC}$		0.5	100	μA
V_{IL}	Input Low Voltage		−0.1		0.8	V
V_{IH}	Input High Voltage		2.0		V_{CC}+1	V
V_{OL1}	Output Low Voltage	I_{OL} = 2.1 mA			0.45	V
V_{OH1}	Output High Voltage	I_{OH} = −400 μA	2.4			V
V_{OL2}	Output Low Voltage	I_{OL} = 0 μA			0.1	V
V_{OH2}	Output High Voltage	I_{OH} = 0 μA	V_{CC} −0.1			V

AC Electrical Characteristics

Symbol	Parameter	Conditions	NMC27C512 20		NMC27C512 250, E250		NMC27C512 300		NMC27C512 350 M350		Units
			Min	Max	Min	Max	Min	Max	Min	Max	
t_{ACC}	Address to Output Delay	$\overline{CE} = \overline{OE} = V_{IL}$		200		250		300		350	ns
t_{CE}	$\overline{CE}$ to Output Delay	$\overline{OE} = V_{IL}$		200		250		300		350	ns
t_{OE}	$\overline{OE}$ to Output Delay	$\overline{CE} = V_{IL}$		75		100		120		120	ns
t_{DF}	$\overline{OE}$ High to Output Float	$\overline{CE} = V_{IL}$	0	60	0	60		105		105	ns
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$, Whichever Occurred First	$\overline{CE} = \overline{OE} = V_{IL}$	0		0		0		0		ns

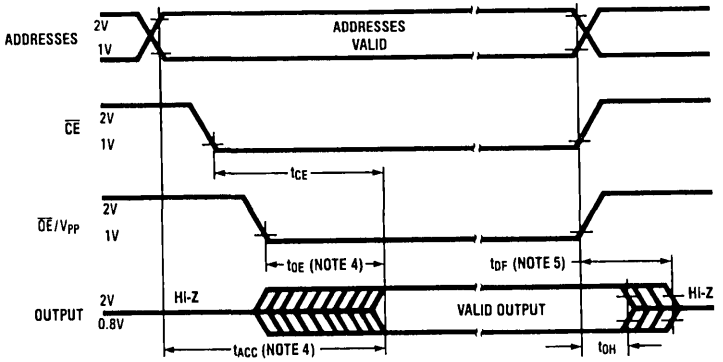
Capacitance $T_A = +25°C$, f = 1 MHz (Note 3)

Symbol	Parameter	Conditions	Typ	Max	Units
C_{IN}	Input Capacitance	$V_{IN} = 0V$	4	6	pF
C_{OUT}	Output Capacitance	$V_{OUT} = 0V$	8	12	pF

AC Test Conditions

Output Load	1 TTL Gate and C_L = 100 pF (Note 9)	Timing Measurement Reference Level	
Input Rise and Fall Times	≤20 ns	Inputs	1V and 2V
Input Pulse Levels	0.45V to 2.4V	Outputs	0.8V and 2V

AC Waveforms



TL/D/8754-3

Note 1: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Note 2: Typical values are for $T_A = +25°C$ and nominal supply voltages.

Note 3: This parameter is only sampled and is not 100% tested.

Note 4: $\overline{OE}$ may be delayed $t_{ACC} - t_{OE}$ after the falling edge of $\overline{CE}$ without impacting t_{ACC}.

Note 5: The t_{DF} compare level is determined as follows:

High to TRI-STATE, the measured V_{OH1} (DC) −0.10V

Low to TRI-STATE, the measured V_{OL1} (DC) +0.10V

Note 6: TRI-STATE may be attained using $\overline{OE}$ or $\overline{CE}$.

Note 7: The power switching characteristics of EPROMs require careful device decoupling. It is recommended that at least a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND.

Note 8: The outputs must be restricted to V_{CC} +0.3V to avoid latch-up and device damage.

Note 9: 1 TTL Gate: I_{OL} = 1.6 mA, I_{OH} = −400 μA.

CL: 100 pF includes fixture capacitance.

Programming Characteristics

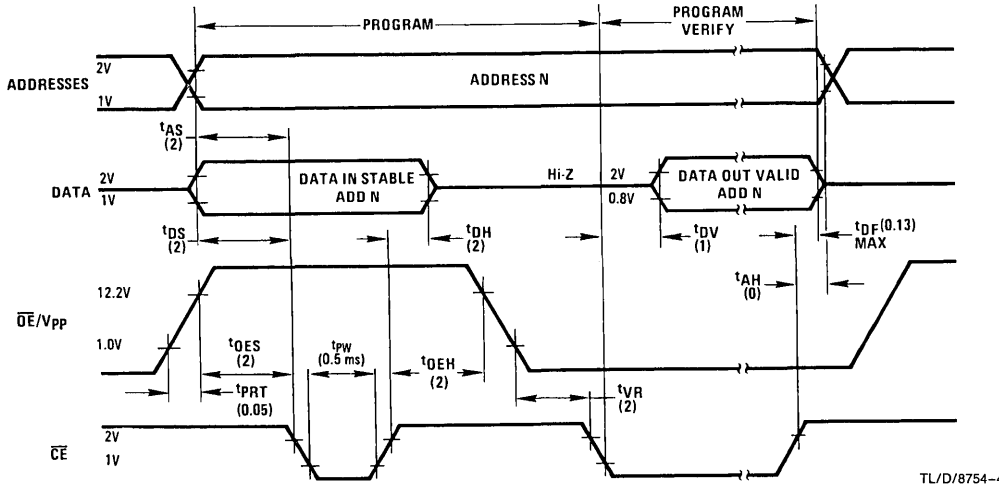
T_A = +25°C ±5°C, V_{CC} = 6V ±0.25V, V_{PP} = 12.2V to 13.3V (Notes 2, 3, & 4)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{AS}	Address Set-Up Time		2			μs
t_{OES}	$\overline{OE}$ Set-Up Time		2			μs
t_{DS}	Data Set-Up Time		2			μs
t_{AH}	Address Hold Time		0			μs
t_{DH}	Data Hold Time		2			μs
t_{DF}	Output Enable to Output Float Delay	$\overline{CE}$ = V_{IL}	0		130	ns
t_{PW}	Program Pulse Width		0.5	0.5	10	ms
I_{PP}	V_{PP} Supply Current During Programming Pulse	$\overline{CE}$ = V_{IL} $\overline{OE}$ = V_{PP}			30	mA
I_{CC}	V_{CC} Supply Current				10	mA
t_{OEH}	$\overline{OE}$ Hold Time		2			μs
t_{DV}	Data Valid from $\overline{CE}$	$\overline{CE}$ = V_{IL}, $\overline{OE}$ = V_{IL}			1	μs
t_{PRT}	$\overline{OE}$ Pulse Rise Time During Programming		50			ns
t_{VR}	V_{PP} Recovery Time		2			μs

AC Test Conditions

V_{CC}	6V ±0.25V	Timing Measurement Reference Level	
V_{PP}	12.2V to 13.3V	Inputs	1V and 2V
Input Rise and Fall Times	≤20 ns	Outputs	0.8V and 2V
Input Pulse Levels	0.45V to 2.4V		

Programming Waveforms (Note 3)



Note: All times shown in parentheses are minimum and in μs unless otherwise specified.

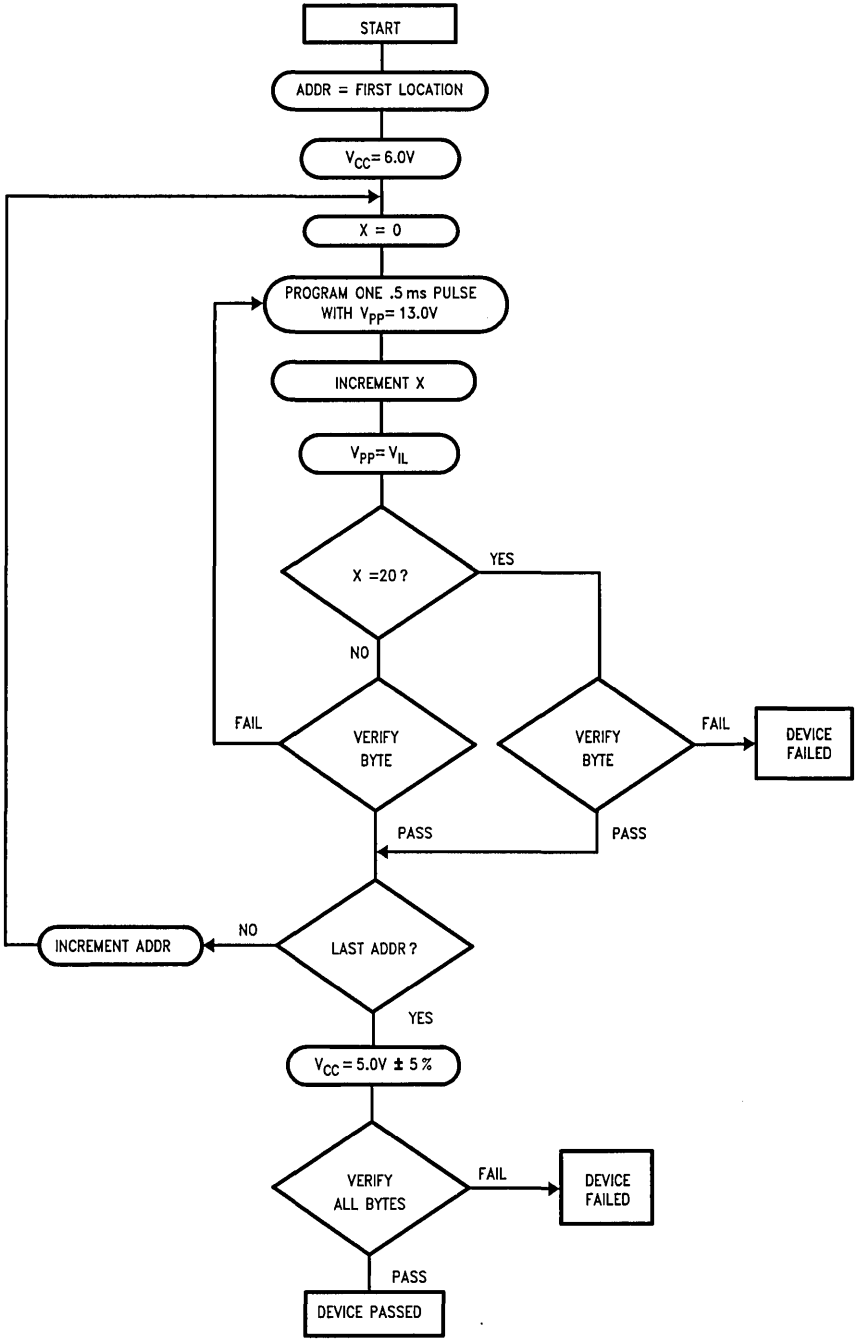
Note 1: National's standard product warranty applies only to devices programmed to specifications described herein.

Note 2: V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}. The NMC27C512 must not be inserted into or removed from a board with voltage applied to V_{PP} or V_{CC}.

Note 3: The maximum allowable voltage which may be applied to the V_{PP} pin during programming is 14V. Care must be taken when switching the V_{PP} supply to prevent overshoot exceeding this 14V maximum specification. At least a 0.1 μF capacitor is required across V_{PP}, V_{CC} to GND to suppress spurious voltage transients which may damage the device.

Note 4: Programming and program verify are tested at nominal power supply voltages.

INTERACTIVE PROGRAMMING FLOW CHART



TL/D/8754–5

Functional Description

DEVICE OPERATION

The five modes of operation of the NMC27C512 are listed in Table I. It should be noted that all inputs for the five modes may be at TTL levels. The power supplies required are V_{CC} and V_{PP}. The V_{CC} power supply must be at 6V during the three programming modes, and at 5V in the other two modes. The $\overline{OE}/V_{PP}$ pin must be at 13V in the programming mode, and at V_{IL} in the read and verify mode.

Read Mode

The NMC27C512 has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data to the output pins, independent of device selection. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs t_{OE} after the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC}-t_{OE}$.

Standby Mode

The NMC27C512 has a standby mode which reduces the active power dissipation by 99%, from 55 mW to 0.55 mW. The NMC27C512 is placed in the standby mode by applying a CMOS high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tying

Because NMC27C512s are usually used in larger memory arrays, National has provided a 2-line control function that accommodates this use of multiple memory connections. The 2-line control function allows for:

a) the lowest possible memory power dissipation, and

b) complete assurance that output bus contention will not occur.

To most efficiently use these two control lines, it is recommended that $\overline{CE}$ (pin 20) be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 22) be made a common connection to all devices in the array and connected to the READ line from the system control bus. This assures that all deselected memory devices are in their low power standby modes and that the output pins are active only when data is desired from a particular memory device.

Programming

CAUTION: Exceeding 14V on pin 22 (V_{PP}) will damage the NMC27C512.

Initially, and after each erasure, all bits of the NMC27C512 are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The NMC27C512 is in the programming mode when $\overline{OE}/V_{PP}$ is at 13.0V. It is required that at least a 0.1 μF capacitor be placed across $\overline{OE}/V_{PP}$, V_{CC} to ground to suppress spurious voltage transients which may damage the device. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, an active low, TTL program pulse is applied to the $\overline{CE}$ input. A program pulse must be applied at each address location that is to be programmed. Any location may be programmed at any time—either individually, sequentially, or at random. The NMC27C512 is designed to be programmed in either of two ways: single pulse programming, where each address is programmed with a 10 ms pulse; or interactive programming, where each address is programmed with a series of 0.5 ms pulses until it verifies (up to a maximum of 20 pulses or 10 ms). Only the interactive programming method has been tested. The NMC27C512 must not be programmed with a DC signal applied to the $\overline{CE}$ input.

Programming multiple NMC27C512s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled NMC27C512s may be connected together when they are programmed with the same data. A low level TTL pulse applied to the $\overline{CE}$ input programs the paralleled NMC27C512s.

TABLE I. Mode Selection

Pins / Mode	$\overline{CE}$ (20)	$\overline{OE}/V_{PP}$ (22)	V_{CC} (28)	Outputs (11–13, 15–19)
Read	V_{IL}	V_{IL}	5	D_{OUT}
Standby	V_{IH}	Don't Care	5	Hi-Z
Program	Pulsed V_{IH} to V_{IL}	V_{PP}	6	D_{IN}
Program Verify	V_{IL}	V_{IL}	6	D_{OUT}
Program Inhibit	V_{IH}	V_{PP}	6	Hi-Z

Functional Description (Continued)

Program Inhibit

Programming multiple NMC27C512s in parallel with different data is also easily accomplished. Except for $\overline{CE}$ all like inputs (including $\overline{OE}$) of the parallel NMC27C512s may be common. A TTL low level program pulse applied to an NMC27C512's $\overline{CE}$ input with $\overline{OE}/V_{PP}$ at 13.0V will program that NMC27C512. A TTL high level $\overline{CE}$ input inhibits the other NMC27C512 from being programmed.

Program Verify

A verify should be performed on the programmed bits to determine whether they were correctly programmed. The verify may be performed with $\overline{OE}/V_{PP}$ at V_{IL}.

MANUFACTURER'S IDENTIFICATION CODE

The NMC27C512 has a manufacturer's identification code to aid in programming. The code, shown in Table II, is two bytes wide and is stored in a ROM configuration on the chip. It identifies the manufacturer and the device type. The code for the NMC27C512 is "8F45", where "8F" designates that it is made by National Semiconductor, and "45" designates a 512k part.

The code is accessed by applying 12V ±0.5V to address pin A9. Addresses A1–A8, A10–A15, $\overline{CE}$, and $\overline{OE}$ are held at V_{IL}. Address A0 is held at V_{IL} for the manufacturer's code, and at V_{IH} for the device code. The code is read out on the 8 data pins. Proper code access is only guaranteed at 25°C ±5°C.

The primary purpose of the manufacturer's identification code is automatic programming control. When the device is inserted in an EPROM programmer socket, the programmer reads the code and then automatically calls up the specific programming algorithm for the part. This automatic programming control is only posible with programmers which have the capability of reading the code.

ERASURE CHARACTERISTICS

The erasure characteristics of the NMC27C512 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000Å–4000Å range. Data shows that constant exposure to room-level fluorescent lighting could erase the typical NMC27C512 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. Opaque lables should be placed over the NMC27C512's window to prevent unintentional erasure. Covering the window will also prevent temporary functional failure due to the generation of photo currents.

The recommended erasure procedure for the NMC27C512 is exposure to short wave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity x exposure time) for erasure should be a minimum of 15W-sec/cm^2.

The NMC27C512 should be placed within 1 inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure. Table III shows the minimum NMC27C512 erasure time for various light intensities.

An erasure system should be calibrated periodically. The distance from lamp to unit should be maintained at one inch. The erasure time increases as the square of the distance. (If distance is doubled the erasure time increases by a factor of 4.) Lamps lose intensity as they age. When a lamp is changed, the distance has changed or the lamp has aged, the system should be checked to make certain full erasure is occurring. Incomplete erasure will cause symptoms that can be misleading. Programmers, components, and even system designs have been erroneously suspected when incomplete erasure was the problem.

SYSTEM CONSIDERATION

The power switching characteristics of EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by voltage transitions on input pins. The magnitude of these transient current peaks is dependent on the output capacitance loading of the device. The associated V_{CC} transient voltage peaks can be suppressed by properly selected decoupling capacitors. It is recommended that at least a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance. In addition, at least a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for each eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage drop caused by the inductive effects of the PC board traces.

3

TABLE II. Manufacturer's Identification Code

Pins	A_0 (10)	O_7 (19)	O_6 (18)	O_5 (17)	O_4 (16)	O_3 (15)	O_2 (13)	O_1 (12)	O_0 (11)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	1	1	1	8F
Device Code	V_{IH}	0	1	0	0	0	1	0	1	45

TABLE III. Minimum NMC27C512 Erasure Time

Light Intensity (Micro-Watts/cm^2)	Erasure Time (Minutes)
15,000	20
10,000	25
5,000	50

National Semiconductor Corporation

PRELIMINARY

microCMOS

NMC27C1024
1,048,576-Bit (64k x 16) UV Erasable CMOS PROM

General Description

The NMC27C1024 is a high-speed 1024k UV erasable and electrically reprogrammable CMOS EPROM, ideally suited for applications where fast turnaround, pattern experimentation and low power consumption are important requirements.

The NMC27C1024 is designed to operate with a single +5V power supply with ±10% tolerance.

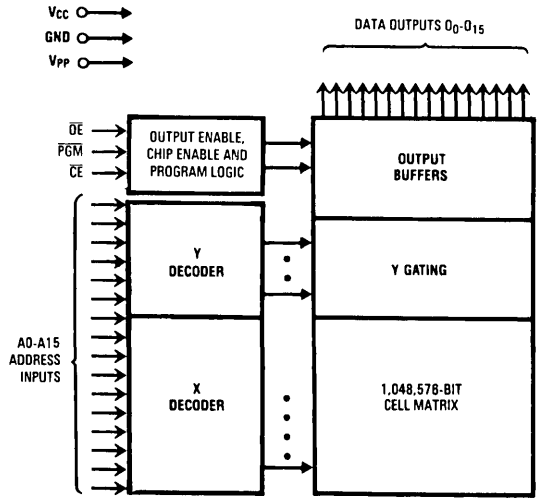
The NMC27C1024 is packaged in a 40-pin dual in-line package with transparent lid. The transparent lid allows the user to expose the chip to ultraviolet light to erase the bit pattern. A new pattern can then be written electrically into the device by following the programming procedure.

This EPROM is fabricated with National's proprietary, time proven microCMOS double-poly silicon gate technology which combines high performance and high density with low power consumption and excellent reliability.

Features

- Access time down to 90 ns, microCMOS technology
- Low CMOS power consumption
  - — Active Power: 275 mW max
  - — Standby Power: 550 μW max
- Performance compatible to 16-bit and 32-bit microprocessors
- Single 5V power supply
- Pin compatible with NMOS 1024k EPROMS
- Fast and reliable programming (0.5 ms for most bytes)
- Static operation—no clocks required
- TTL, CMOS compatible inputs/outputs
- TRI-STATE® output

Block Diagram



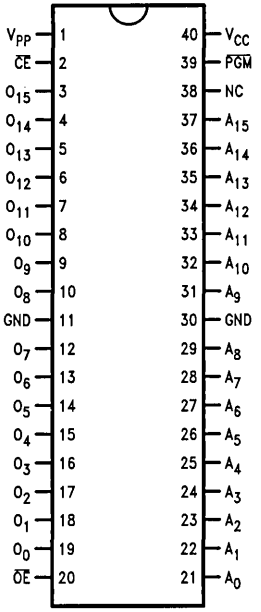
TL/D/8806-1

Pin Names

A0–A15	Addresses
$\overline{CE}$	Chip Enable
$\overline{OE}$	Output Enable
O_0–O_{15}	Outputs
$\overline{PGM}$	Program
NC	No Connect

Connection Diagram

Dual-In-Line Package



TL/D/8806–2

Commercial Temp Range (0°C to +70°C)

V_{CC} = 5V ±10%

Parameter/Order Number	Access Time (ns)
NMC27C1024Q90	90
NMC27C1024Q120	120
NMC27C1024Q150	150
NMC27C1024Q200	200

3

Absolute Maximum Ratings (Note 1)

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Temperature Under Bias	−10°C to +80°C
Storage Temperature	−65°C to +125°C
All Input Voltages except A9 Respect to Ground	+6.5V to −0.3V
All Output Voltages with Respect to Ground	V_{CC} +0.3 to GND −0.3V
V_{PP} Supply Voltage and A9 with Respect to Ground	+12.0V to −0.3V
Power Dissipation	1.0W
Lead Temperature (Soldering, 10 sec.)	300°C

ESD rating is to be determined.

Operating Conditions (Note 7)

Temperature Range	
NMC27C1024Q90, 120, 150, 200	0°C to +70°C
V_{CC} Power Supply	5V ±10%

READ OPERATION

DC Electrical Characteristics

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
I_{LI}	Input Load Current	$V_{IN} = V_{CC}$ or GND			10	μA
I_{LO}	Output Leakage Current	$V_{OUT} = V_{CC}$ or GND, $\overline{CE} = V_{IH}$			10	μA
I_{CC1} (Note 10)	V_{CC} Current (Active) TTL Inputs	$\overline{CE} = V_{IL}$, f = 5 MHz Inputs = V_{IH} or V_{IL} I/O = 0 mA		20	70	mA
I_{CC2} (Note 10)	V_{CC} Current (Active) CMOS Inputs	$\overline{CE}$ = GND, f = 5 MHz Inputs = V_{CC} or GND, I/O = 0 mA		15	50	mA
I_{CCSB1}	V_{CC} Current (Standby) TTL Inputs	$\overline{CE} = V_{IH}$		0.1	1	mA
I_{CCSB2}	V_{CC} Current (Standby) CMOS Inputs	$\overline{CE} = V_{CC}$		0.5	100	μA
V_{IL}	Input Low Voltage		−0.1		0.8	V
V_{IH}	Input High Voltage		2.0		V_{CC} + 1	V
V_{OL1}	Output Low Voltage	I_{OL} = 2.1 mA			0.45	V
V_{OH1}	Output High Voltage	I_{OH} = −400 μA	2.4			V
V_{OL2}	Output Low Voltage	I_{OL} = 0 μA			0.1	V
V_{OH2}	Output High Voltage	I_{OH} = 0 μA	V_{CC} −0.1			V

AC Electrical Characteristics

Symbol	Parameter	Conditions	NMC27C1024 90		NMC27C1024 120		NMC27C1024 150		NMC27C1024 200		Units
			Min	Max	Min	Max	Min	Max	Min	Max	
t_{ACC}	Address to Output Delay	$\overline{CE} = \overline{OE} = V_{IL}$		90		120		150		200	ns
t_{CE}	$\overline{CE}$ to Output Delay	$\overline{OE} = V_{IL}$		90		120		150		200	ns
t_{OE}	$\overline{OE}$ to Output Delay	$\overline{CE} = V_{IL}$		40		50		60		75	ns
t_{DF}	$\overline{OE}$ High to Output Float	$\overline{CE} = V_{IL}$	0	40	0	40	0	50	0	60	ns
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$, Whichever Occurred First	$\overline{CE} = \overline{OE} = V_{IL}$	0		0		0		0		ns

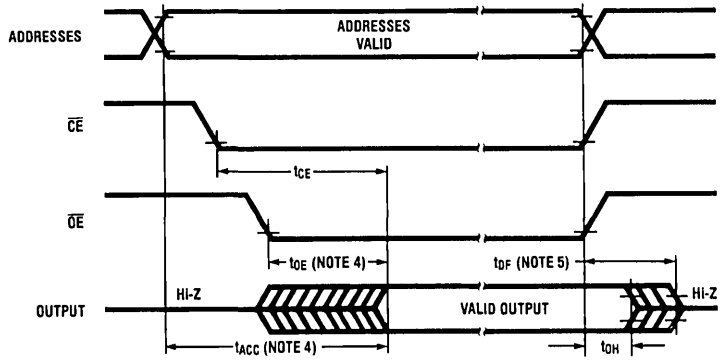
Capacitance $T_A = +25°C$, f = 1 MHz (Note 3)

Symbol	Parameter	Conditions	Typ	Max	Units
C_{IN}	Input Capacitance	$V_{IN} = 0V$	8	10	pF
C_{OUT}	Output Capacitance	$V_{OUT} = 0V$	10	15	pF

AC Test Conditions

Output Load	1 TTL Gate and C_L = 100 pF (Note 9)	Timing Measurement Reference Level	
Input Rise and Fall Times	≤ 5 ns	Inputs	1V and 2V
Input Pulse Levels	0.45V to 2.4V	Outputs	0.8V and 2V

AC Waveforms



TL/D/8806–3

Note 1: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Note 2: Typical values are for $T_A = +25°C$ and nominal supply voltages.

Note 3: This parameter is only sampled and is not 100% tested.

Note 4: $\overline{OE}$ may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of $\overline{CE}$ without impacting t_{ACC}.

Note 5: The t_{DF} compare level is determined as follows:

High to TRI-STATE, the measured V_{OH1} (DC) − 0.10V

Low to TRI-STATE, the measured V_{OL1} (DC) + 0.10V

Note 6: TRI-STATE may be attained using $\overline{OE}$ or $\overline{CE}$.

Note 7: The power switching characteristics of EPROMs require careful device decoupling. It is recommended that at least a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND.

Note 8: The outputs must be restricted to V_{CC} +0.3V to avoid latch-up and device damage.

Note 9: 1 TTL Gate: $I_{OL} = 1.6$ mA, $I_{OH} = -400$ μA.
CL: 100 pF includes fixture capacitance.

Note 10: V_{PP} may be connected to V_{CC} except during programming.

Programming Characteristics

T_A = +25°C ±5°C, V_{CC} = 6V ±0.25V, V_{PP} = 11.5 ±0.3V (Notes 2, 3, & 4)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{AS}	Address Set-Up Time		1			μs
t_{OES}	$\overline{OE}$ Set-Up Time		1			μs
t_{DS}	Data Set-Up Time		1			μs
t_{AH}	Address Hold Time		0			μs
t_{DH}	Data Hold Time		1			μs
t_{CES}	$\overline{CE}$ Set-Up Time		1			μs
t_{ACC}	Address to Output Delay				250	ns
t_{DF}	Output Enable to Output Float Delay	$\overline{CE}$ = V_{IL}	0		60	ns
t_{OE}	Output Enable to Output Delay	$\overline{CE}$ = V_{IL}			100	ns
t_{PW}	Program Pulse Width		0.5	0.5	10	ms
I_{PP}	V_{PP} Supply Current During Programming Pulse	$\overline{CE}$ = V_{IL}			30	mA
I_{CC}	V_{CC} Supply Current				10	mA

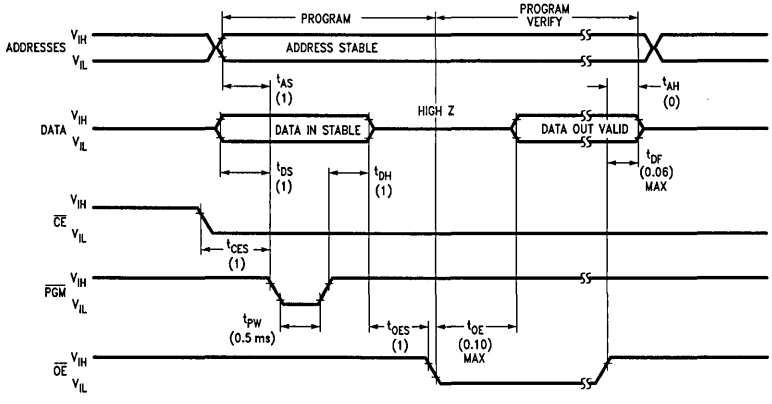
AC Test Conditions

V_{CC}	6V ±0.25V
V_{PP}	11.5V ±0.3V
Input Rise and Fall Times	≤ 20 ns
Input Pulse Levels	0.45V to 2.4V

Timing Measurement Reference Level

Inputs	1V and 2V
Outputs	0.8V and 2V

Programming Waveforms (Note 3)



TL/D/8806–4

Note: All times shown in parentheses are minimum and in μs unless otherwise specified.
T_{OE} and T_{DF} are characteristics of the device but must be accomodated by the programmer.

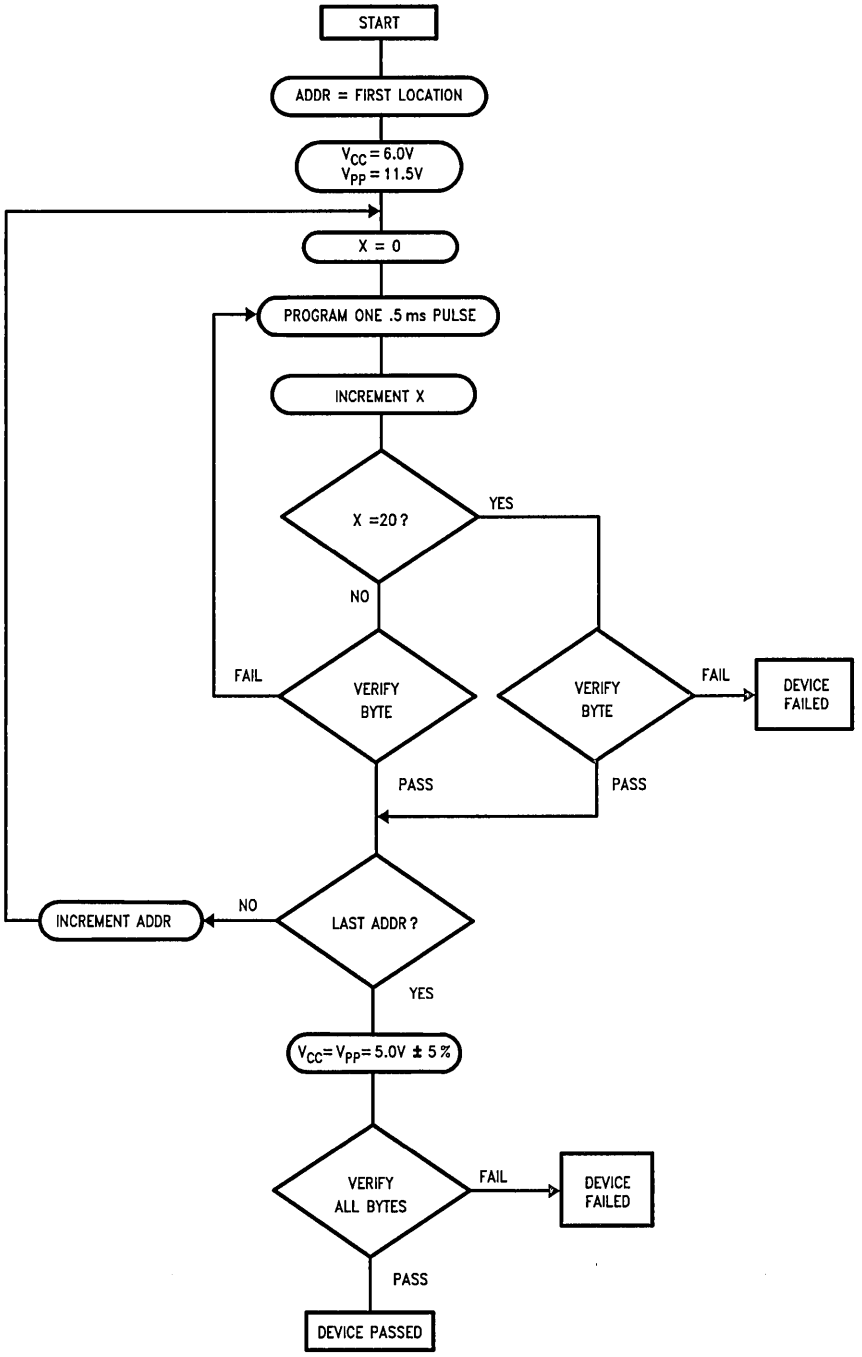
Note 1: National's standard product warranty applies only to devices programmed to specifications described herein.

Note 2: V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}. The NMC27C1024 must not be inserted into or removed from a board with voltage applied to V_{PP} or V_{CC}.

Note 3: The maximum allowable voltage which may be applied to the V_{PP} pin during programming is 12V. Care must be taken when switching the V_{PP} supply to prevent overshoot exceeding this 12V maximum specification. At least a 0.1 μF capacitor is required across V_{PP}, V_{CC} to GND to suppress suprious voltage transients which may damage the device.

Note 4: Programming and program verify are tested at nominal power supply voltages.

Interactive Programming Flow Chart



TL/D/8806–5

Functional Description

DEVICE OPERATION

The five modes of operation of the NMC27C1024 are listed in Table I. It should be noted that all inputs for the five modes are at TTL levels. The power supplies required are V_{CC} and V_{PP}. The V_{PP} power supply must be at 11.5V during the three programming modes, and must be at 5V in the other two modes. The V_{CC} power supply must be at 6V during the three programming modes, and at 5V in the other two modes.

Read Mode

The NMC27C1024 has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data to the output pins, independent of device selection. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs t_{OE} after the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC}-t_{OE}$.

Standby Mode

The NMC27C1024 has a standby mode which reduces the active power dissipation by over 99.7%, from 275 mW to 0.55 mW. The NMC27C1024 is placed in the standby mode by applying a CMOS high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tying

Because NMC27C1024s are usually used in larger memory arrays, National has provided a 2-line control function that accommodates this use of multiple memory connections. The 2-line control function allows for:

a) the lowest possible memory power dissipation, and

b) complete assurance that output bus contention will not occur.

To most efficiently use these two control lines, it is recommended that $\overline{CE}$ (pin 2) be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 20) be made a common connection to all devices in the array and connected to the READ line from the system control bus. This assures that all deselected memory devices are in their low power standby modes and that the output pins are active only when data is desired from a particular memory device.

Programming

CAUTION: Exceeding 12V on the V_{PP} or A9 pin will damage the NMC27C1024.

Initially, and after each erasure, all bits of the NMC27C1024 are in the "1" state. Data is introduced by selectively programming "0's" into the desired bit locations. Although only "0's" will be programmed, both "1's" and "0's" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The NMC27C1024 is in the programming mode when the V_{PP} power supply is at 11.5V and $\overline{OE}$ is at V_{IH}. It is required that at least a 0.1 μF capacitor be placed across V_{PP}, V_{CC} to ground to suppress spurious voltage transients which may damage the device. The data to be programmed is applied 16 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, an active low, TTL program pulse is applied to the PGM input. A program pulse must be applied at each address location to be programmed. Any location may be programmed at any time—either individually, sequentially, or at random. The NMC27C1024 is designed to be programmed in either of two ways: single pulse programming, where each address is programmed with a 10 ms pulse; or interactive programming, where each address is programmed with a series of 0.5 ms pulses until it verifies (up to a maximum of 20 pulses or 10 ms). Only the interactive programming method has been tested. The NMC27C1024 must not be programmed with a DC signal applied to the $\overline{PGM}$ input.

Programming multiple NMC27C1024s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the parallel NMC27C1024s may be connected together when they are programmed with the same data. A low level TTL pulse applied to the $\overline{PGM}$ input programs the paralleled NMC27C1024s.

TABLE I. Mode Selection

Pins / Mode	$\overline{CE}$ (2)	$\overline{OE}$ (20)	$\overline{PGM}$ (39)	V_{PP} (1)	V_{CC} (40)	Outputs (3–10, 12–19)
Read	V_{IL}	V_{IL}	V_{IH}	V_{CC}	5	D_{OUT}
Standby	V_{IH}	Don't Care	Don't Care	V_{CC}	5	Hi-Z
Program	V_{IL}	V_{IH}	⊓	11.5	6	D_{IN}
Program Verify	V_{IL}	V_{IL}	V_{IH}	11.5	6	D_{OUT}
Program Inhibit	V_{IH}	Don't Care	Don't Care	11.5	6	Hi-Z

Functional Description (Continued)

Program Inhibit

Programming multiple NMC27C1024s in parallel with different data is also easily accomplished. Except for $\overline{CE}$ all like inputs (including $\overline{OE}$) of the parallel NMC27C1024s may be common. A TTL low level program pulse applied to an NMC27C1024's $\overline{CE}$ input with V_{PP} at 11.5V will program that NMC27C1024. A TTL high level $\overline{CE}$ input inhibits the other NMC27C1024s from being programmed.

Program Verify

A verify should be performed on the programmed bits to determine whether they were correctly programmed. The verify may be performed with V_{PP} at 11.5V. Except during programming and program verify, V_{PP} must be at V_{CC}.

Manufacturer's Identification Code

The NMC27C1024 has a manufacturer's identification code to aid in programming. When the device is inserted in an EPROM programmer socket, the programmer reads the code and then automatically calls up the specific programming algorithm for the part. This automatic programming control is only possible with programmers which have the capability of reading the code.

The Manufacturer's Identification code, shown in Table II, specifically identifies the manufacturer and the device type. The code for the NMC27C1024 is "8FD6", where "8F" designates that it is made by National Semiconductor, and "D6" designates a 1 Meg part.

The code is accessed by applying 11.4V to 12.0V to address pin A9. Addresses A1–A8, A10–A15, and all control pins are held at V_{IL}. Address pin A0 is held at V_{IL} for the manufacturer's code, and held at V_{IH} for the device code. The code is read on the lower eight data pins, O_0–O_7. Proper code access is only guaranteed at 25°C ±5°C.

ERASURE CHARACTERISTICS

The erasure characteristics of the NMC27C1024 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000Å–4000Å range. Data shows that constant exposure to room-level fluorescent lighting could erase the typical NMC27C1024 in approximately 3 years, while it would take approximatley 1 week to cause erasure when exposed to direct sunlight. Opaque labels should be placed over the NMC27C1024s window to prevent unintentional erasure. Covering the window will also prevent temporary functional failure due to the generation of photo currents.

The recommended erasure procedure for the NMC27C1024 is exposure to short wave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity x exposure time) for erasure should be a minimum of 15W-sec/cm^2.

The NMC27C1024 should be placed within 1 inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure. Table III shows the minimum NMC27C1024 erasure time for various light intensities.

An erasure system should be calibrated periodically. The distance from lamp to unit should be maintained at one inch. The erasure time increases as the square of the distance. (If distance is doubled the erasure time increases by a factor of 4.) Lamps lose intensity as they age. When a lamp is changed, the distance has changed, or the lamp has aged, the system should be checked to make certain full erasure is occurring. Incomplete erasure will cause symptoms that can be misleading. Programmers, components, and even system designs have been erroneously suspected when incomplete erasure was the problem.

SYSTEM CONSIDERATION

The power switching characteristics of EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by voltage transitions on input pins. The magnitude of these transient current peaks is dependent on the output capacitance loading of the device. The associated V_{CC} transient voltage peaks can be suppressed by properly selected decoupling capacitors. It is recommended that at least a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance. In addition, at least a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for each eight devices. The bulk, capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage drop caused by the inductive effects of the PC board traces.

3

TABLE II. Manufacturer's Identification Code

Pins	A_0 (21)	O_7 (12)	O_6 (13)	O_5 (14)	O_4 (15)	O_3 (16)	O_2 (17)	O_1 (18)	O_0 (19)	Hex Data
Manufacturer Code	V_{IL}	1	0	0	0	1	1	1	1	8F
Device Code	V_{IH}	1	1	0	1	0	1	1	0	D6

TABLE III. Minimum NMC27C1024 Erasure Time

Light Intensity (Micro-Watts/cm^2)	Erasure Time (Minutes)
15,000	20
10,000	25
5,000	50

National Semiconductor Corporation

DM54LS373/DM74LS373/DM54LS374/DM74LS374 TRI-STATE® Octal D-Type Transparent Latches and Edge-Triggered Flip-Flops

General Description

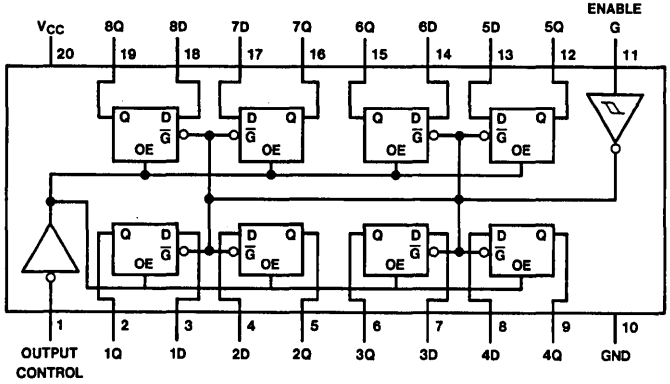
These 8-bit registers feature totem-pole TRI-STATE outputs designed specifically for driving highly-capacitive or relatively low-impedance loads. The high-impedance state and increased high-logic level drive provide these registers with the capability of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pull-up components. They are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers. (Continued)

Features

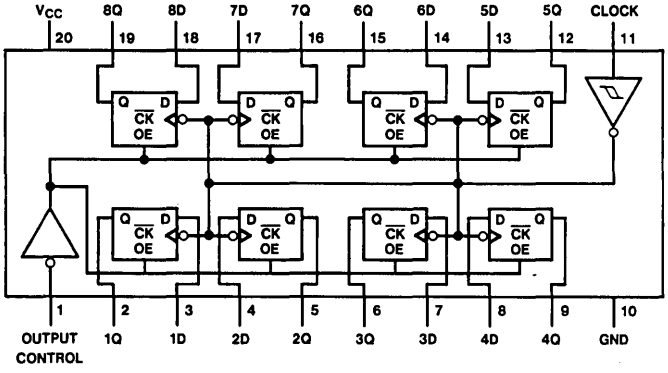
- Choice of 8 latches or 8 D-type flip-flops in a single package
- TRI-STATE bus-driving outputs
- Full parallel-access for loading
- Buffered control inputs
- Clock/Enable input has hysteresis to improve noise rejection
- P-N-P inputs reduce D-C loading on data lines

Connection Diagrams

Dual-In-Line Packages



TL/F/6431–1



TL/F/6431–2

Order Number DM54LS373J, DM74LS373N, DM54LS374J or DM74LS374N
See NS Package Number J20A or N20A

Absolute Maximum Ratings (See Note)

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Supply Voltage	7V
Input Voltage	7V
Storage Temperature Range	−65°C to +150°C
Operating Free Air Temperature Range	
DM54LS	−55°C to +125°C
DM74LS	0°C to +70°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter		DM54LS373			DM74LS373			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Votage		2			2			V
V_{IL}	Low Level Input Voltage				0.7			0.8	V
I_{OH}	High Level Output Current				−1			−2.6	mA
I_{OL}	Low Level Output Current				12			24	mA
t_W	Pulse Width (Note 2)	Enable High	15			15			ns
		Enable Low	15			15			
t_{SU}	Data Setup Time (Notes 1 & 2)		5↓			5↓			ns
t_H	Data Hold Time (Notes 1 & 2)		20↓			20↓			ns
T_A	Free Air Operating Temperature		−55		125	0		70	°C

Note 1: The symbol (↓) indicates the falling edge of the clock pulse is used for reference.

Note 2: T_A = 25°C and V_{CC} = 5V.

'LS373 Electrical Characteristics

over recommended operating free air temperature (unless otherwise noted)

Symbol	Parameter	Conditions		Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max, V_{IH} = Min	DM54	2.4	3.4		V
			DM74	2.4	3.1		
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IL} = Max, V_{IH} = Min	DM54		0.25	0.4	V
			DM74		0.35	0.5	
		I_{OL} = 12 mA, V_{CC} = Min	DM74			0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V				−0.4	mA
I_{OZH}	Off-State Output Current with High Level Output Voltage Applied	V_{CC} = Max, V_O = 2.7V, V_{IH} = Min, V_{IL} = Max				20	μA
I_{OZL}	Off-State Output Current with Low Level Output Voltage Applied	V_{CC} = Max, V_O = 0.4V, V_{IH} = Min, V_{IL} = Max				−20	μA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	DM54	−20		−100	mA
			DM74	−20		−100	
I_{CC}	Supply Current	V_{CC} = Max			24	40	mA

'LS373 Switching Characteristics at V_{CC} = 5V and T_A = 25°C

(See Section 1 for Test Waveforms and Output Load)

Symbol	Parameter	From (Input) To (Output)	R_L = 667Ω				Units
			C_L = 45 pF		C_L = 150 pF		
			Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output	Data to Q		18		26	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Data to Q		18		27	ns
t_{PLH}	Propagation Delay Time Low to High Level Output	Enable to Q		30		38	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	Enable to Q		30		36	ns
t_{PZH}	Output Enable Time to High Level Output	Output Control to Any Q		28		36	ns
t_{PZL}	Output Enable Time to Low Level Output	Output Control to Any Q		36		50	ns
t_{PHZ}	Output Disable Time from High Level Output (Note 3)	Output Control to Any Q		20			ns
t_{PLZ}	Output Disable Time from Low Level Output (Note 3)	Output Control to Any Q		25			ns

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Note 3: C_L = 5 pF.

Recommended Operating Conditions

Symbol	Parameter		DM54LS374			DM74LS374			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage		2			2			V
V_{IL}	Low Level Input Voltage				0.7			0.8	V
I_{OH}	High Level Output Current				−1			−2.6	mA
I_{OL}	Low Level Output Current				12			24	mA
f_{CLK}	Clock Frequency (Note 2)		0		35	0		35	MHz
f_{CLK}	Clock Frequency (Note 3)		0		20	0		20	MHz
t_W	Pulse Width (Note 4)	Clock High	15			15			ns
		Clock Low	15			15			
t_{SU}	Data Setup Time (Notes 1 & 4)		20↑			20↑			ns
t_H	Data Hold Time (Notes 1 & 4)		0↑			0↑			ns
T_A	Free Air Operating Temperature		−55		125	0		70	°C

Note 1: The symbol (↑) indicates the rising edge of the clock pulse is used for reference.

Note 2: C_L = 45 pF, R_L = 667Ω, T_A = 25°C and V_{CC} = 5V.

Note 3: C_L = 150 pF, R_L = 667Ω, T_A = 25°C and V_{CC} = 5V.

Note 4: T_A = 25°C and V_{CC} = 5V.

'LS374 Electrical Characteristics

over recommended operating free air temperature (unless otherwise noted)

Symbol	Parameter	Conditions		Min	Typ (Note 1)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA				−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max, V_{IH} = Min	DM54	2.4	3.4		V
			DM74	2.4	3.1		
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IL} = Max, V_{IH} = Min	DM54		0.25	0.4	V
			DM74		0.35	0.5	
		I_{OL} = 12 mA, V_{CC} = Min	DM74		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V				−0.4	mA
I_{OZH}	Off-State Output Current with High Level Output Voltage Applied	V_{CC} = Max, V_O = 2.7V, V_{IH} = Min, V_{IL} = Max				20	μA
I_{OZL}	Off-State Output Current with Low Level Output Voltage Applied	V_{CC} = Max, V_O = 0.4V, V_{IH} = Min, V_{IL} = Max				−20	μA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 2)	DM54	−20		−100	mA
			DM74	−20		−100	
I_{CC}	Supply Current	V_{CC} = Max			27	45	mA

'LS374 Switching Characteristics at V_{CC} = 5V and T_A = 25°C

(See Section 1 for Test Waveforms and Output Load)

Symbol	Parameter	R_L = 667Ω				Units
		C_L = 45 pF		C_L = 150 pF		
		Min	Max	Min	Max	
f_{MAX}	Maximum Clock Frequency	35		20		MHz
t_{PLH}	Propagation Delay Time Low to High Level Output		28		32	ns
t_{PHL}	Propagation Delay Time High to Low Level Output		28		38	ns
t_{PZH}	Output Enable Time to High Level Output		28		44	ns
t_{PZL}	Output Enable Time to Low Level Output		28		44	ns
t_{PHZ}	Output Disable Time from High Level Output (Note 3)		20			ns
t_{PLZ}	Output Disable Time from Low Level Output (Note 3)		25			ns

Note 1: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 2: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Note 3: C_L = 5 pF.

General Description (Continued)

The eight latches of the DM54/74LS373 are transparent D-type latches meaning that while the enable (G) is high the Q outputs will follow the data (D) inputs. When the enable is taken low the output will be latched at the level of the data that was set up.

The eight flip-flops of the DM54/74LS374 are edge-triggered D-type flip flops. On the positive transition of the clock, the Q outputs will be set to the logic states that were set up at the D inputs.

Schmitt-trigger buffered inputs at the enable/clock lines simplify system design as AC and DC noise rejection is improved by typically 400 mV due to the input hysteresis. A buffered output control input can be used to place the eight outputs in either a normal logic state (high or low logic levels) or a high-impedance state. In the high-impedance state the outputs neither load nor drive the bus lines significantly.

The output control does not affect the internal operation of the latches or flip-flops. That is, the old data can be retained or new data can be entered even while the outputs are off.

Function Tables

DM54/74LS373

Output Control	Enable G	D	Output
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

DM54/74LS374

Output Control	Clock	D	Output
L	↑	H	H
L	↑	L	L
L	L	X	Q_0
H	X	X	Z

H = High Level (Steady State), L = Low Level (Steady State), X = Don't Care

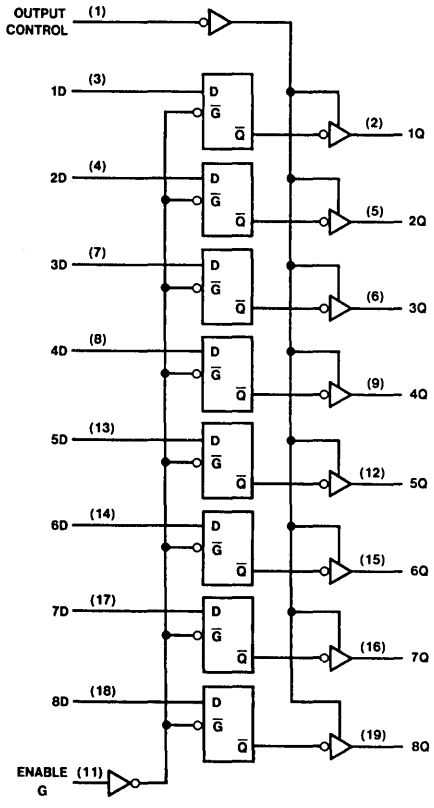
↑ = Transition from low-to-high level, Z = High Impedance State

Q_0 = The level of the output before steady-state input conditions were established.

Logic Diagrams

DM54/74LS373

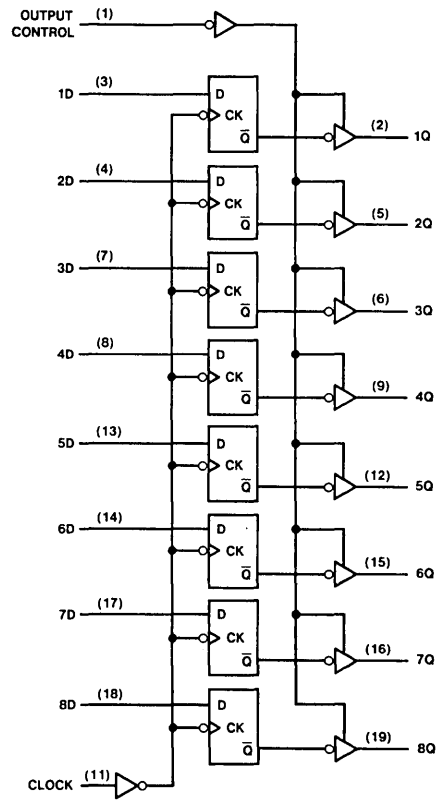
Transparent Latches



TL/F/6431–3

DM54/74LS374

Positive-Edge-Triggered Flip-Flops



TL/F/6431–4

National Semiconductor Corporation

microCMOS

MM54HC373/MM74HC373 TRI-STATE® Octal D-Type Latch

General Description

These high speed octal D-type latches utilize microCMOS Technology, 3.5 micron silicon gate P-well CMOS. They possess the high noise immunity and low power consumption of standard CMOS integrated circuits, as well as the ability to drive 15 LS-TTL loads. Due to the large output drive capability and the TRI-STATE feature, these devices are ideally suited for interfacing with bus lines in a bus organized system.

When the LATCH ENABLE input is high, the Q outputs will follow the D inputs. When the LATCH ENABLE goes low, data at the D inputs will be retained at the outputs until LATCH ENABLE returns high again. When a high logic level is applied to the OUTPUT CONTROL input, all outputs go to a high impedance state, regardless of what signals are present at the other inputs and the state of the storage elements.

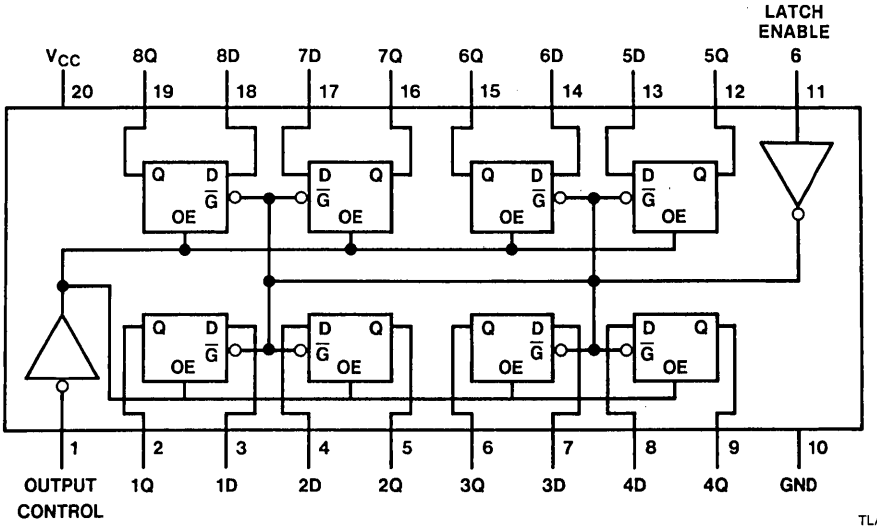
The 54HC/74HC logic family is speed, function, and pin-out compatible with the standard 54LS/74LS logic family. All inputs are protected from damage due to static discharge by internal diode clamps to V_{CC} and ground.

Features

- Typical propagation delay: 18 ns
- Wide operating voltage range: 2 to 6 volts
- Low input current: 1 μA maximum
- Low quiescent current: 80 μA maximum (74 Series)
- Output drive capability: 15 LS-TTL loads

Connection Diagram

Dual-In-Line Package



TL/F/5335–1

Top View

Order Number MM54HC373J or MM74HC373J,N
See NS Package J20A or N20A

Truth Table

Output Control	Latch Enable	Data	373 Output
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

H = high level, L = low level
Q_0 = level of output before steady-state input conditions were established.
Z = high impedance

Absolute Maximum Ratings (Notes 1 & 2)

Supply Voltage (V_{CC})	−0.5 to +7.0V
DC Input Voltage (V_{IN})	−1.5 to V_{CC} +1.5V
DC Output Voltage (V_{OUT})	−0.5 to V_{CC} +0.5V
Clamp Diode Current (I_{IK}, I_{OK})	±20 mA
DC Output Current, per pin (I_{OUT})	±35 mA
DC V_{CC} or GND Current, per pin (I_{CC})	±70 mA
Storage Temperature Range (T_{STG})	−65°C to +150°C
Power Dissipation (P_D)	500 mW
Lead Temp. (T_L) (Soldering 10 seconds)	260°C

Operating Conditions

	Min	Max	Units
Supply Voltage (V_{CC})	2	6	V
DC Input or Output Voltage (V_{IN},V_{OUT})	0	V_{CC}	V
Operating Temp. Range (T_A)			
MM74HC	−40	+85	°C
MM54HC	−55	+125	°C
Input Rise or Fall Times (t_r, t_f) V_{CC} = 2.0V		1000	ns
V_{CC} = 4.5V		500	ns
V_{CC} = 6.0V		400	ns

DC Electrical Characteristics

Symbol	Parameter	Conditions	V_{CC}	T_A = 25°C	74HC T_A = −40 to 85°C	54HC T_A = −55 to 125°C	Units
				Typ	Guaranteed Limits		
V_{IH}	Minimum High Level Input Voltage		2.0V		1.5	1.5	1.5	V
			4.5V		3.15	3.15	3.15	V
			6.0V		4.2	4.2	4.2	V
V_{IL}	Maximum Low Level Input Voltage		2.0V		0.3	0.3	0.3	V
			4.5V		0.9	0.9	0.9	V
			6.0V		1.2	1.2	1.2	V
V_{OH}	Minimum High Level Output Voltage	V_{IN} = V_{IH} or V_{IL} $\|I_{OUT}\| \leq 20\ \mu A$	2.0V	2.0	1.9	1.9	1.9	V
			4.5V	4.5	4.4	4.4	4.4	V
			6.0V	6.0	5.9	5.9	5.9	V
		V_{IN} = V_{IH} or V_{IL} $\|I_{OUT}\| \leq 6.0$ mA	4.5V	4.2	3.98	3.84	3.7	V
		$\|I_{OUT}\| \leq 7.8$ mA	6.0V	5.7	5.48	5.34	5.2	V
V_{OL}	Maximum Low Level Output Voltage	V_{IN} = V_{IH} or V_{IL} $\|I_{OUT}\| \leq 20\ \mu A$	2.0V	0	0.1	0.1	0.1	V
			4.5V	0	0.1	0.1	0.1	V
			6.0V	0	0.1	0.1	0.1	V
		V_{IN} = V_{IH} or V_{IL} $\|I_{OUT}\| \leq 6.0$ mA	4.5V	0.2	0.26	0.33	0.4	V
		$\|I_{OUT}\| \leq 7.8$ mA	6.0V	0.2	0.26	0.33	0.4	V
I_{IN}	Maximum Input Current	V_{IN} = V_{CC} or GND	6.0V		±0.1	±1.0	±1.0	μA
I_{OZ}	Maximum TRI-STATE Output Leakage Current	V_{IN} = V_{IH} or V_{IL}, OC = V_{IH} V_{OUT} = V_{CC} or GND	6.0V		±0.5	±5	±10	μA
I_{CC}	Maximum Quiescent Supply Current	V_{IN} = V_{CC} or GND I_{OUT} = 0 μA	6.0V		8.0	80	160	μA

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: Power Dissipation temperature derating — plastic "N" package: −12 mW/°C from 65°C to 85°C; ceramic "J" package: −12 mW/°C from 100°C to 125°C.

Note 4: For a power supply of 5V ±10% the worst case output voltages (V_{OH}, and V_{OL}) occur for HC at 4.5V. Thus the 4.5V values should be used when designing with this supply. Worst case V_{IH} and V_{IL} occur at V_{CC} = 5.5V and 4.5V respectively. (The V_{IH} value at 5.5V is 3.85V.) The worst case leakage current (I_{IN}, I_{CC}, and I_{OZ}) occur for CMOS at the higher voltage and so the 6.0V values should be used.

AC Electrical Characteristics $V_{CC}=5V$, $T_A=25°C$, $t_r=t_f=6$ ns

Symbol	Parameter	Conditions	Typ	Guaranteed Limit	Units
t_{PHL}, t_{PLH}	Maximum Propagation Delay, Data to Q	$C_L=45$ pF	18	25	ns
t_{PHL}, t_{PLH}	Maximum Propagation Delay, Clock to Q	$C_L=45$ pF	21	30	ns
t_{PZH}, t_{PZL}	Maximum Output Enable Time	$R_L=1$ kΩ $C_L=45$ pF	20	28	ns
t_{PHZ}, t_{PLZ}	Maximum Output Disable Time	$R_L=1$ kΩ $C_L=5$ pF	18	25	ns
t_S	Minimum Set Up Time			5	ns
t_H	Minimum Hold Time			10	ns
t_W	Minimum Pulse Width		9	16	ns

AC Electrical Characteristics $V_{CC}=2.0$–$6.0V$, $C_L=50$ pF, $t_r=t_f=6$ ns (unless otherwise specified)

Symbol	Parameter	Conditions	V_{CC}	$T_A=25°C$		74HC $T_A=-40$ to $85°C$	54HC $T_A=-55$ to $125°C$	Units
				Typ		Guaranteed Limits		
t_{PHL}, t_{PLH}	Maximum Propagation Delay, Data to Q	$C_L=50$ pF	2.0V	50	150	188	225	ns
		$C_L=150$ pF	2.0V	80	200	250	300	ns
		$C_L=50$ pF	4.5V	22	30	37	45	ns
		$C_L=150$ pF	4.5V	30	40	50	60	ns
		$C_L=50$ pF	6.0V	19	26	31	39	ns
		$C_L=150$ pF	6.0V	26	35	44	53	ns
t_{PHL}, t_{PLH}	Maximum Propagation Delay, Clock to Q	$C_L=50$ pF	2.0V	63	175	220	263	ns
		$C_L=150$ pF	2.0V	110	225	280	338	ns
		$C_L=50$ pF	4.5V	25	35	44	52	ns
		$C_L=150$ pF	4.5V	35	45	56	68	ns
		$C_L=50$ pF	6.0V	21	30	37	45	ns
		$C_L=150$ pF	6.0V	28	39	49	59	ns
t_{PZH}, t_{PZL}	Maximum Output Enable Time	$R_L=1$ kΩ						
		$C_L=50$ pF	2.0V	50	150	188	225	ns
		$C_L=150$ pF	2.0V	80	200	250	300	ns
		$C_L=50$ pF	4.5V	21	30	37	45	ns
		$C_L=150$ pF	4.5V	30	40	50	60	ns
		$C_L=50$ pF	6.0V	19	26	31	39	ns
		$C_L=150$ pF	6.0V	26	35	44	53	ns
t_{PHZ}, t_{PLZ}	Maximum Output Disable Time	$R_L=1$ kΩ $C_L=50$ pF	2.0V	50	150	188	225	ns
			4.5V	21	30	37	45	ns
			6.0V	19	26	31	39	ns
t_S	Minimum Set Up Time		2.0V	5	25	31	38	ns
			4.5V	2	5	6	8	ns
			6.0V	2	5	6	8	ns
t_H	Minimum Hold Time		2.0V	20	50	60	75	ns
			4.5V	6	10	13	20	ns
			6.0V	6	10	13	20	ns
t_W	Minimum Pulse Width		2.0V	30	80	100	120	ns
			4.5V	10	16	20	24	ns
			6.0V	9	14	18	20	ns
t_{THL}, t_{TLH}	Maximum Output Rise and Fall Time, Clock	$C_L=50$ pF	2.0V	25	60	75	90	ns
			4.5V	7	12	15	18	ns
			6.0V	6	10	13	15	ns
C_{PD}	Power Dissipation Capacitance (Note 5)	(per latch) OC = V_{CC} OC = GND		30 50				pF pF
C_{IN}	Maximum Input Capacitance			5	10	10	10	pF
C_{OUT}	Maximum Output Capacitance			15	20	20	20	pF

Note 5: C_{PD} determines the no load dynamic power consumption, $P_D=C_{PD}\,V_{CC}^2\,f+I_{CC}\,V_{CC}$, and the no load dynamic current consumption, $I_S=C_{PD}\,V_{CC}\,f+I_{CC}$.

3

National Semiconductor Corporation

PRELIMINARY

MM57409 Super Number Cruncher

General Description

The Super Number Cruncher (SNC) is an MOS/LSI arithmetic processor (actually, a pre-programmed member of National's single-chip microcontroller COPS™ family) intended for use in number processing applications. Scientific calculator functions, conditional output capability, internal number storage, and input/output instructions have been combined in this single chip device. Programming is done in calculator keyboard level language which simplifies software development. Data or instructions can be synchronous or asynchronous, I/O digit count, I/O notation mode, and error control are user programmable; a sense input and flag outputs are available for single bit control; and instructions and lines are available for I/O expansion.

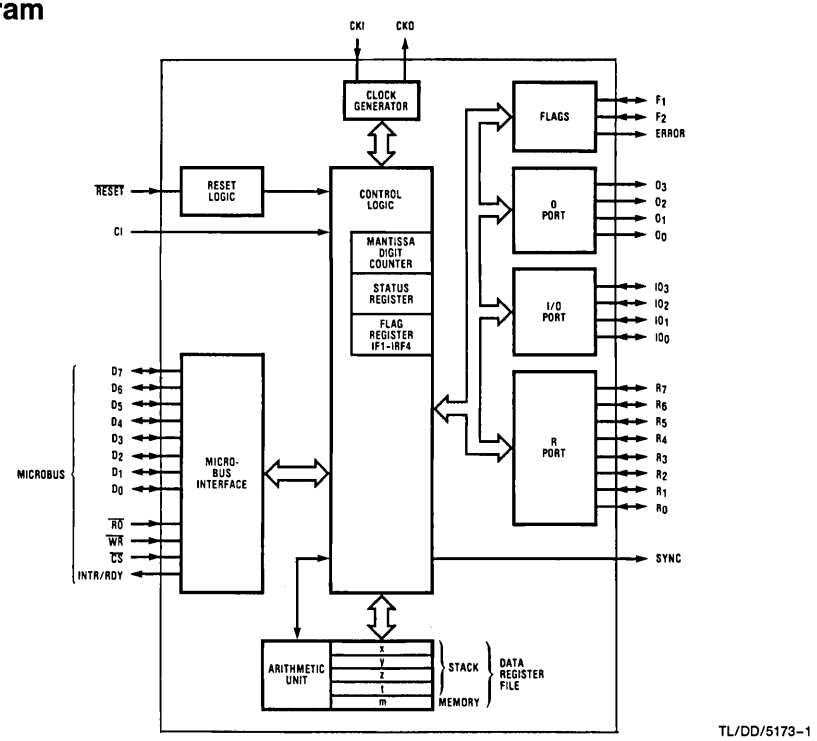
Applications

- Instruments
- Microprocessor/minicomputer peripheral
- Test Equipment
  Process controllers

Features

- Scientific calculator instructions (RPN)
  - Up to 12-digit mantissa, 2-digit exponent
  - 4-register stack, one memory register
  - Trigonometric functions, logarithmic functions, Y^x, e^x, π
  - Error flag generation and recovery
- Flexible input/output
  - Multidigit OUT instruction with foating-point or scientific notations
  - Programmable mantissa digit count for OUT instruction
  - Sense input and flag outputs
  - Eight high-impedance I/O lines (TRI-STATE®), six I/O lines, and four output lines for I/O expansion.
- Interface simplicity
  - On-chip clock OSC
  - Generates all I/O control signals
  - MICROBUS™ interface

Block Diagram



TL/DD/5173-1

FIGURE 1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Voltage at Any Pin Relative to GND	−0.5V to +7V
Ambient Operating Temperature	0°C to +70°C
Ambient Storage Temperature	−65°C to +50°C
Lead Temperature (Soldering, 10 sec.)	300°C
Power Dissipation	0.75W at 25°C
	0.4W at 70°C
Total Source Current	80 mA
Total Sink Current	75 mA

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not insured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics $0°C \le T_A \le 70°C$; $4.5V \le V_{CC} \le 6.3V$ unless otherwise noted.

Parameter	Conditions	Min	Max	Units
Operating Voltage (V_{CC})	(Note 1)	4.5	6.3	V
Power Supply Ripple	(Peak to Peak)		0.4	V
Operating Supply Current	All inputs and outputs open			
	$T_A = 0°C$		41	mA
	$T_A = 25°C$		35	
	$T_A = 70°C$		27	
Input Voltage Levels				
CKI				
Logic High (V_{IH})	$V_{CC} = 5V \pm 5\%$	2.0		
Logic Low (V_{IL})	$V_{CC} = 5V \pm 5\%$	−0.3	0.4	
Logic High (V_{IH})	V_{CC} = Max	2.5		
$\overline{RESET}$				
Logic High		0.7 V_{CC}		V
Logic Low		−0.3	0.6	
All Other Inputs				
Logic High	V_{CC} = Max	2.5		
Logic High	$V_{CC} = 5V \pm 5\%$	2.0		
Logic Low		−0.3	0.8	
Input Capacitance			7.0	pF
Input Leakage ($\overline{RD}$, $\overline{CS}$, $\overline{WR}$)		−1.0	1.0	μA
Output Voltage Levels				
TTL Operation				
Logic High (V_{OH})	$I_{OH} = -100\ \mu A$	2.4		
Logic Low (V_{OL})	$I_{OL} = 1.6$ mA		0.4	V
CMOS Operation				
Logic High (V_{OH})	$I_{OH} = -10\ \mu A$	$V_{CC} - 0.4$		
Logic Low (V_{OL})	$I_{OL} = 10\ \mu A$		0.2	
Output Current Levels				
Output Source Current	$V_{CC} = 4.5V$			
	$V_{OH} = 2.4V$	−100	−650	μA
TRI-STATE Output Leakage				
Current (R, D Lines)		−2.5	2.5	μA
CKO Output				
Output Source Current (I_{OH})	$V_{OH} = 2.0V$	−0.2		mA
Output Sink Current (I_{OL})	$V_{OL} = 0.4V$	0.4		mA
Input Current Levels				
Input Load Source Current	$V_{IH} = 2.0V$	14	230	μA
(CI, $\overline{RESET}$)	V_{CC} 4.5V			
Total Sink Current Allowed				
All I/O Combined			75	mA
Each D, R Port			20	mA
Each, O, I/O, F1, F2			10	mA
SYNC Line			2.5	mA
Total Source Current Allowed				
All I/O Combined			80	
Each D Pin			5.0	mA
All Other Output Pins			1.6	

AC Electrical Characteristics $0°C \le T_A \le 70°C$; $4.5V \le V_{CC} \le 6.3V$, unless otherwise noted.

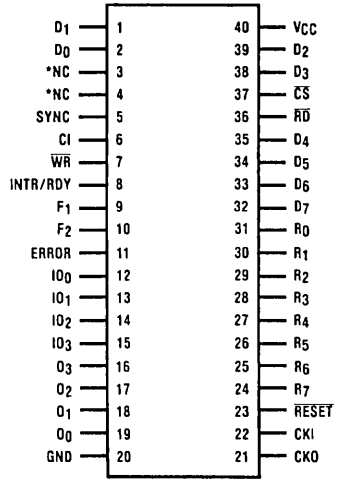
Parameter	Conditions	Min	Max	Units
Microcycle Time (tm)		4.0	10	μs
CKI Frequency (f_I)		1.6	4.0	MHz
Duty Cycle (Note 2)	f_I = 4.0 MHz	30	60	%
Rise Time			60	ns
Fall Time			40	ns
Inputs *(Figure 3)*				
t_{SETUP}		1.7		μs
t_{HOLD}		300		ns
Output Propagation Delay	C_L = 50 PF			
(Figure 3)	V_{OUT} = 1.5V			
CKO t_{PD1}, t_{PD0}			0.17	μs
SYNC t_{PD1}, t_{PD0}	R_L = 2.4 kΩ		1.0	μs
All Other Outputs	R_L = 5.0 kΩ		1.4	μs
MICROBUS Timing	C_L = 100 pF V_{CC} = 5V ±5%			
Read Operation *(Figure 4)*				
$\overline{CS}$ Stable Before $\overline{RD}$—t_{CSR}		65		
$\overline{CS}$ Hold Time for $\overline{RD}$—t_{RCS}		20		
$\overline{RD}$ Pulse Width—t_{RR}		400		
Data Delay from $\overline{RD}$—t_{RD}			375	
$\overline{RD}$ to Data Floating—t_{DF}			250	ns
Write Operation *(Figure 5)*				
$\overline{CS}$ Stable Before $\overline{WR}$—t_{CSW}		65		
$\overline{CS}$ Hold Time for $\overline{WR}$—t_{WCS}		20		
$\overline{WR}$ Pulse Width—t_{WW}		400		
Data Setup Time—t_{DW}		320		
Data Hold Time—t_{WD}		100		
INTD/RDY Transition Time from $\overline{WR}$—t_{WI}			700	

Note 1: V_{CC} voltage change must be less than 0.5V in a 1 ms period to maintain proper operation.

Note 2: Duty Cycle = $t_{W1}/(t_{W1} + t_{W0})$.

Connection Diagram

Dual-In-Line



Top View TL/DD/5173-2

*NC means no external connection allowed on these pins.

FIGURE 2

Molded Dual-In-Line Package (N)
Order Number MM57409N
See NS Package Number N40A

Pin Name	Description
D_0–D_7	8-Bit Data Bus (bidirectional)
IO_0–IO_3	4-Bit I/O Lines (bidirectional)
O_0–O_3	4-Bit Output Lines
R_0–R_7	8-Bit Bidirectional I/O Lines with TRI-STATE
ERROR	Error Flag Line
V_{CC}, GND	Power Supply
F1, F2	Two Bidirectional Flags
INTR/RDY, $\overline{WR}$ $\overline{RD}$, $\overline{CS}$	MICROBUS Handshake Signals
$\overline{RESET}$	System Reset
SYNC	Microcycle Clock
CKI, CKO	System Oscillator
CI	Test/Conditional Input

Functional Description

The MM57409 Super Number Cruncher (SNC) is intended for microprocessor number processing applications as a microcomputer peripheral device. The block diagram of the SNC is shown in *Figure 1*. Data paths are illustrated in simplified form to depict how the various logic elements communicate with each other. Positive logic is used, i.e., when a bit is set, it is a logic "1" (greater than 2V) and when a bit is reset, it is a logic "0" (less than 0.8V).

INTERNAL LOGIC

The data register file, consisting of a 4-level stack and one memory location, is the source and destination register for all mathematic operations. The organization of an x, y, z, or t level is shown in *Figure 6*. The structure of the M (memory) level is similar, lacking only the guard/link digit. The mantissa and the exponent each have a sign, which may be read or changed by SNC instructions. The guard/link digit in each of the x, y, z, and t levels of the stack is used for the execution of the arithmetic instructions. Regardless of any operating mode, all data internal to the SNC is stored in scientific notation format with the mantissa rounded to 12 digits.

The arithmetic unit performs all mathematic operations of the MM57409, storing its results in the data register file.

The R port can be used as an 8-bit high-impedance I/O port. If the R port is to be used, the first instruction the SNC must receive after a power up or reset is the RIO instruction. The state of the R port will be 03 hex. A host processor can then instruct the SNC to output an 8-bit value to the R port or read the state of the R lines. Before reading the input state of the R lines, the lines *must be* put into a high-impedance mode. The R lines are then high-impedance inputs and must be externally driven high or low.

An external processor may also cause a 4-bit value to be output to either the O or I/O ports. The input state of the I/O port or the present state of the output lines of the O port may be read from the SNC. If the I/O lines are to be used as inputs, the value F must first be written to the port using the I/O instruction.

Two flags are provided on the SNC—F1 and F2. These flags may be set high or pulsed high through the use of appropriate SNC instructions. When pulsed, the pulse width of F1 is four microcycles and the pulse width of F2 is three microcycles. (A microcycle is the external clock input divided by 16—see Oscillator section for further information.) These flags may also be tested by test/conditional output instructions if they are first set high. An external test input (CI) is also provided for conditional output control and is tested by the TCI instruction.

There are four general purpose internal flags (IF1–IF4) in the SNC that can be reset, or tested using the appropriate instructions.

OPERATING MODES

The SNC has several user controllable operating modes. Any combination of these modes may be selected. These modes are:

1. Angular mode—The SNC can be instructed to accept and return data in either degrees or radians.
2. Input/Output mode—The SNC will accept and output numerical data in either scientific notation (signed exponent and mantissa) or floating-point (decimal point position, signed mantissa). Note that only the input or output data can be in floating-point or scientific notation—the data internal to the SNC is always in scientific notation.
3. Rounding—The SNC can be instructed to round the output to the Mantissa Digit Count (MDC). This rounding may also be disabled. The rounding mode causes rounding on the output data only—the data internal to the SNC is still in scientific notation format with the mantissa rounded to 12 digits.

Note: If both the floating-point I/O mode and the rounding mode are selected, the output data is rounded to the MDC before it is converted to the floating-point mode. This means that fractional numbers ($|x| < 1$) may not appear to be rounded.

4. Mantissa Digit Counter (MDC)—The number of digits the SNC expects in the mantissa during the multiple digit OUT operation may be set anywhere between 1 and 12.

Functional Description (Continued)

STATUS REGISTER

The SNC contains an 8-bit status register *(Figure 7)* which the host can read to determine the exact status of the SNC. The various status information is described below.

Bit 7: Scientific Notation/Floating-Point I/O mode
(MSB) = 1 indicates floating-point I/O
= 0 indicates scientific notation I/O

Bit 6: Rounding
= 1 indicates no rounding to MDC is performed
= 0 indicates rounding to MDC is performed

Bit 5: Not Used

Bit 4: Angular Mode
= 1 indicates angles are expressed in radians
= 0 indicates angles are expressed in degrees

Bit 3: Status of x stack level as fraction
= 1 indicates the absolute value of x is a non-zero fraction
= 0 indicates the absolute value of x is ≥ 1

Bit 2: Polarity of x register
= 1 indicates $x < 0$
= 0 indicates $x > 0$ or $x = 0$

Bit 1: x equivalence to 0
= 1 indicates $x = 0$
= 0 indicates $x \neq$ zero

Bit 0: Error
= 1 indicates an error has occurred
= 0 indicates no error

OSCILLATOR

An oscillator between 1.6 MHz and 4 MHz is required. This may be supplied from an external oscillator that is input to CKI or through the use of the external crystal network as shown in *Figure 8*. In either case, the frequency at CKI is divided by 16 to provide the basic timing reference (microcycle) for the SNC. This signal is available at the SYNC output. A single period of the SYNC signal (measured from rising edge to rising edge) corresponds to one microcycle. The microcycle will be between 5 and 10 μs, depending on the CKI frequency.

INITIALIZATION

The SNC is reset upon power up or upon application of a low going pulse to the $\overline{\text{RESET}}$ input. The reset pulse must be a minimum of three microcycles (three SYNC pulses) in duration in order to reliably reset the device. If the power supply rise time is greater than 1 ms, the circuit of *Figure 9* must be used.

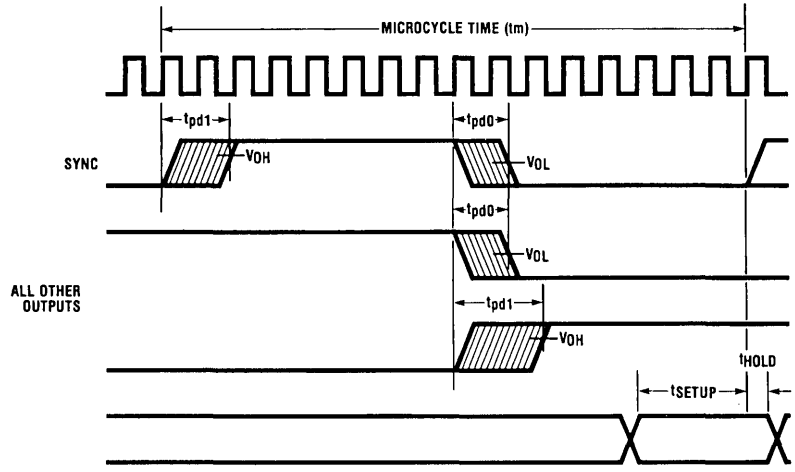
In the reset state, R and the D ports are in a high impedance state; SYNC is the CKI inut frequency divided by 16; and O, I/O, F1, F2, ERROR, and INTR/RDY are reset to 0. The DMC is set to 10, the angular mode is set to degrees, the input/output mode is set to scientific notation, and any data output will be rounded to the MDC.

ERROR CONDITIONS

The error flag and error output are set upon detection of any of the error conditions listed in Table I. The resultant status of the SNC after an error condition is also shown in Table I. The error flag and error output are cleared only upon execution of any one of the following:

1. an ECLR (error clear) instruction
2. an MCLR (master clear) instruction
3. a hardware system reset

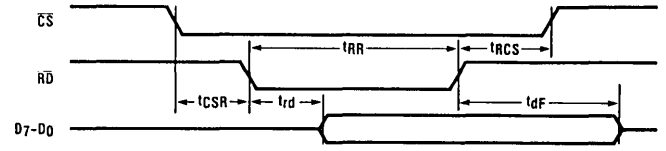
Timing Diagram



TL/DD/5173-3

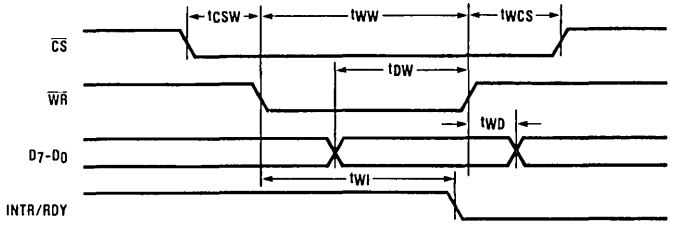
FIGURE 3. Basic Timing

Timing Diagrams (Continued)



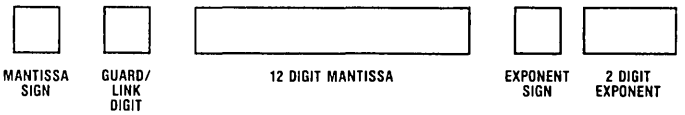
TL/DD/5173-4

FIGURE 4. MICROBUS Read Operation Timing



TL/DD/5173-5

FIGURE 5. MICROBUS Write Operation Timing

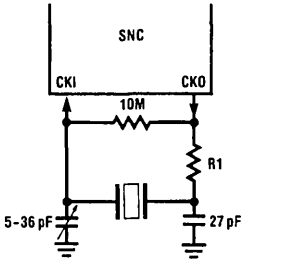


TL/DD/5173-6

FIGURE 6. Stack Level x, y, z or t Structure

Bit		
7	0 =	Scientific Notation I/O Mode
	1 =	Floating Point I/O Mode
6	0 =	Round the Output to the MDC
	1 =	Do Not Round the Output to the MDC
5		Not Used
4	0 =	Angular Data is Expressed in Degrees
	1 =	Angular Data is Expressed in Radians
3	0 =	$\|x\| \geq 1$ or $= 0$
	1 =	$\|x\|$ is a Non-Zero Fraction
2	0 =	$x \geq 0$
	1 =	$x < 0$
1	0 =	$x \neq 0$
	1 =	$x = 0$
0	0 =	No Error Has Occurred
	1 =	Error Has Occurred

FIGURE 7. SNC Status Register

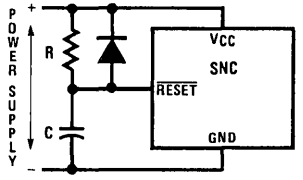


TL/DD/5173-7

FIGURE 8. Crystal Oscillator Configuration

Crystal Oscillator

Crystal Value	R_1
4 MHz	1k
3.58 MHz	1k
2.097 MHz	2k



TL/DD/5173-8

RC ≥ Power Supply Rise Time

FIGURE 9. Power-Up Reset Circuit

Functional Description (Continued)

MICROBUS COMMUNICATION

The SNC communicates via an 8-bit bidirectional bus (D_0–D_7). The 8-bit instruction opcodes are sent to the SNC on this bus and data is transferred between the SNC and its host on this bus. In addition to the bus, a chip select ($\overline{CS}$), read strobe ($\overline{RD}$), write strobe ($\overline{WR}$), and an interrupt/ready signal (INTR/RDY) complete the interface. The SNC will not respond to any signal on the bus or any of the control signals unless it has been selected (i.e., $\overline{CS}$ low) by the host. The SNY is ready to receive data or an instruction whenever the INTR/RDY is high. INTR/RDY high also indicates that the SNC has data available for the host. Pulsing the $\overline{WR}$ line (write strobe) transfers the information on the bus into the SNC. The write strobe going low forces INTR/RDY to go low. INTR/RDY will not return to the high state until the SNC has completed the operation required by the previous write. Pulsing the $\overline{RD}$ line (read strobe) will cause an internal 8-bit register of the SNC to be transferred to the external bus. INTR/RDY does not go low. Therefore, a read in this manner does not affect the readiness of the SNC. The SNC is in an idle condition when INTR/RDY is high. It is waiting for the next write and it will not proceed or escape from the idle loop until the device is selected and the $\overline{WR}$ is pulsed low. This characteristic makes the interface simple and allows the device to work with a host running at any speed.

Note: Do not pulse $\overline{WR}$ with the $\overline{CS}$ low when INTR/RDY is low. This can, in some cases, create errors in the SNC.

INPUT/OUTPUT CONFIGURATIONS

The SNC input lines have the following configurations, illustrated in *Figure 10:*

1. $\overline{RD}$, $\overline{WR}$, $\overline{CS}$—High-impedance *(Figure 10a)*
2. CI, $\overline{RESET}$—Internal load device *(Figure 10b)*

The output lines have the following configurations, illustrated in *Figure 11:*

1. D_7–D_0—TRI-STATE *(Figure 11a)*
2. R_7–R_0—Push-pull with TRI-STATE *(Figure 11b)*
3. O_3–O_0, I/O_0, INTR/RDY, F1, F2, ERROR—Standard *(Figure 11c)*
4. SYNC—Push-pull *(Figure 11d)*

Number Entry Mode Description

If the SNC is not in the number entry mode, the instructions AIN1, DP, BP, PI, or the numbers 0 through 9 will initiate number entry. This means that the stack is pushed (z → t, y → z, x → y), the x register is cleared, and the number entry mode established. If a number was entered to initiate the entry mode, that digit will go into the x register; or, if EE was entered prior to the digit, that digit will go into the x exponent. The subsequent entry of numbers, DP, EE, or PI does not again initiate number entry. Up to 12 mantissa digits may be entered, MSD entered first. Any digit entered after the 12th mantissa digit will simply be ignored. Up to two exponents digits may be entered, MSD entered first. If the user enters more than two exponent digits, only the last two entered will be accepted.

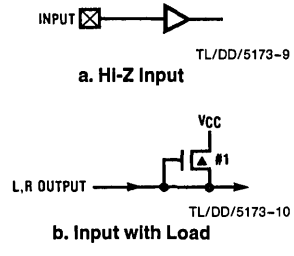
If number entry is initiated by the EE instruction, the x mantissa is loaded with a "1". Subsequent digits will go into the x exponent.

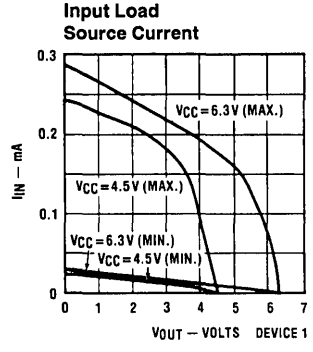
The CS instruction does not initiate number entry. It normally toggles the sign of the x mantissa. If, however, CS is entered after an EE instruction, the sign of the x exponent will be toggled if the exponent is a non-zero number. CS may be mixed with the various digit entry instructions without interference.

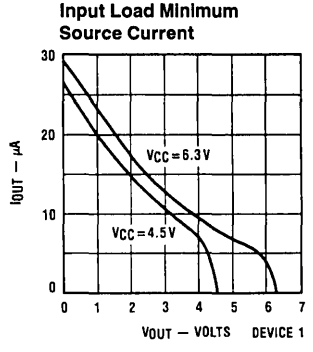
Only mantissa digits may be entered through the use of the AIN1 instruction. This instruction causes the SNC to enter data into the x mantissa, starting at the MSD and ending at the LSD. One digit is entered for each AIN1 instruction. This instruction is actually a 2-byte instruction. The first byte is the instruction's opcode—0E hex. When INTR/RDY goes high, the SNC will respond with a digit address (0N, where N is a number from 0 to 11, MSD → LSD). The second byte writes the address N and the data for that address back to the SNC.

The following example illustrates the first time an AIN1 is entered:

WRITE:	0E hex	(AIN1 opcode)
READ:	00 hex	(SNC says write data to mantissa address 0)
WRITE:	09 hex	(9 → mantissa digit 0)



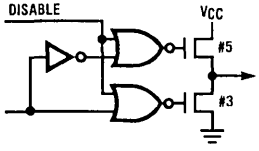




TL/DD/5173-11

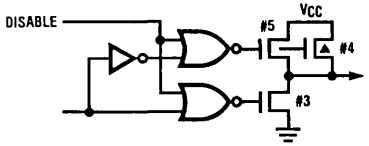
FIGURE 10. SNC Input Characteristics

Input/Output Configuration (Continued)



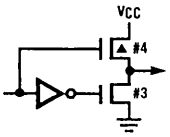
TL/DD/5173–12

a. TRI-STATE Output



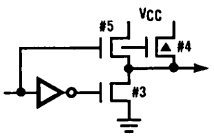
TL/DD/5173–13

b. Push-Pull Outputs with TRI-STATE



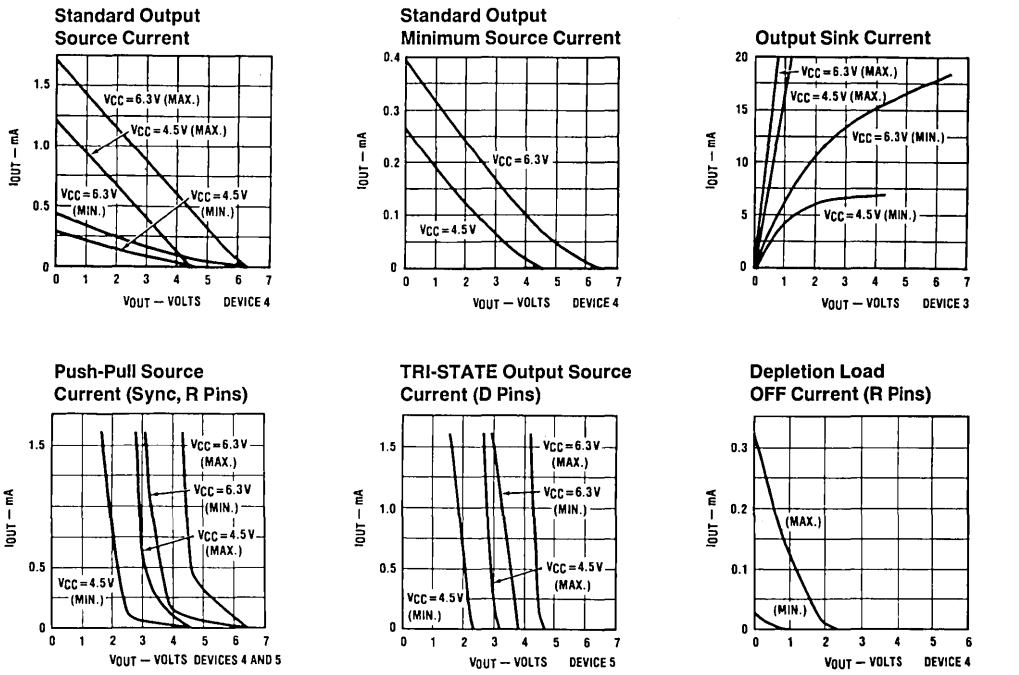
TL/DD/5173–14

c. Standard Output



TL/DD/5173–15

d. Push-Pull Output



TL/DD/5173–16

FIGURE 11. SNC Output Characteristics

The next example illustrates when AIN1 is entered immediately after the previous example:

WRITE	0E hex	(AIN1 opcode)
AD:	01 hex	(SNC says write data to mantissa address 1)
WRITE:	15 hex	(5 → mantissa digit 1)

One exception to the number entry initiation is that the stack is not pushed if the instruction prior to an entered digit was EN (enter)—EN pushed the stack. However, the x register is still cleared and the entered data put in x.

The number entry mode is terminated by any instruction except DP, EE, CS, PI, AIN1, NOP1, or any number.

Data Input Description

The AIN2 instruction is a 2-byte, single digit asynchronous input instruction. This instruction does not initiate number entry mode and the x level of the stack is not cleared. The first byte is the instruction's opcode: 91 hex. The second byte is of the form nD where n is the digit address and D is the BCD digit. Since n is specified by the host, AIN2 can be used to write to any digit in the x register. Digits may only be entered in scientific notation format. This example will write a 7 to the MSD of the x mantissa. Refer to Table IV.

WRITE:	91 hex	(AIN2 opcode)
WRITE:	47 hex	(4 is the address of the MSD mantissa digit, 7 is data)

Data Output Description

The OUT instruction is a multidigit output instruction that will output all digits of the x register. The host writes the OUT opcode (8F hex) and the SNC will respond with the address of the data and the data in the form nD where n is the address and D is the data. The host should then write the same information back to the SNC to indicate that it received the data. This procedure continues until all data has been read by the host. The data addresses are shown in Table III for scientific notation mode and in Table II for floating-point mode.

Table V contains data formats for other output instructions.

Instruction Set Notes

Two of the TEST/CONDITIONAL OUTPUT instructions—IMNZ and DMNZ—deserve special comment. These instructions increment or decrement the memory mantissa and if the new value of the mantissa is = 0, then R will have the value contained in the operand field. The increment/decrement portion affects the entire 12-digit mantissa, starting at digit 12 (LSD), regardless of the MDC and the decimal point. So, if the host wished to decrement the memory twice and then change R, the x register would be cleared, AIN2 would be used to write a 2 to digit 12, and the memory and x would be exchanged. The host could then expect the SNC to load R upon the second execution of the DMNZ instruction.

If the host wished to observe the true results of the LSH (left shift x mantissa) and RSH (right shift x mantissa), rounding the output to the MDC should be disabled. Otherwise the shift might be obscured due to rounding.

Execution times for all SNC instructions are contained in Table VI. Typical instruction times are given for math and memory operation instructions, and worst case times given for all else. These times were found with the CKI input frequency equal to 4 MHz (4 μs microcycle time) and are measured from the rising edge of the $\overline{WR}$ signal to the rising edge of the INTR/RDY line.

TABLE I. SNC Error Conditions

	Error Condition	SNC Status
1.	ln x or log x when x is ≤ 0	x, y, z, t, M unchanged
2.	x+y, x−y, x*y, or y/x when result is ≤ 10^{100} or < 10^{-99}	Previous x → y; x is invalid data
3.	M+x, M−x, M*x, or M/x which result is ≥ 10^{100} or < 10^{-99}	y, z, t, M unchanged; x is invalid data
4.	Tan 90°, 270°, 450°, etc.	y, z, t, M unchanged; x is invalid data
5.	Sin x, Cos x, or Tan x when the absolute value of x is ≤ 9000° (157.08 radians)	y, z, t, M unchanged; x is invalid data
6.	arcsin x or across x when the absolute value of x is > 1 or ≤ 10^{-50}	y, z, t, M unchanged; x is invalid data
7.	square root of x when x < 0	x, y, z, t, M unchanged
8.	y/x when x = 0	x, y swapped; z, t, M unchanged
9.	1/x when x = 0	1 → x, y, z, t, M unchanged
10.	M/x when x = 0	M to x; y, z, t, M unchanged
11.	y^x when y ≤ 0	x, y swapped; z, t, M unchanged
12.	Floating-point OUT instruction when the number of mantissa digits to the left of the decimal point is > 12	x, y, z, t, M unchanged
13.	Attempt to enter a number ≥ 10^{100} or < 10^{-99}	Error occurs on termination of number entry mode. The stack push at initiation of number entry will occur normally. x contains invalid data.

TABLE II. OUT Instruction—Floating-Point

D_7–D_4	DPX	D_3	D_2	D_1	D_0
2		Sm	0	0	Se
3		Decimal Point Position (DP POS).			
4	11	Most significant mantissa digit. On the Out instruction, this digit will be nonzero unless $\|x\| < 1$, in which case it will be zero and decimal point position will be 11.			
5	10	Second most significant mantissa digit.			
.	.	.			
.	.	.			
.	.	.			
MDC+3	2−MDC	Least significant mantissa digit.			

TABLE III. OUT Instruction—Scientific Notation

D_7–D_4	D_3	D_2	D_1	D_0
0	Most significant exponent digit.			
1	Least significant exponent digit.			
2	Sm	0	0	Se
3	Not used.			
4	Most significant mantissa digit. Decimal point follows this digit.			
5	Second most significant mantissa digit.			
.	.			
.	.			
.	.			
MDC+3	Least significant mantissa digit.			

TABLE IV. AIN2 Instruction

D_7–D_4	D_3	D_2	D_1	D_0
0	Most significant exponent digit.			
1	Least significant exponent digit.			
2	Sm	0	0	Se
3	Not used.			
4	Most significant mantissa digit.			
5	Second most significant mantissa digit.			
.	.			
.	.			
.	.			
14	Second least significant mantissa digit.			
15	Least significant mantissa digit.			

Sm = Sign of mantissa, 0 = positive, 1 = negative.

Se = Sign of exponent, 0 = positive, 1 = negative.

MDC = Mantissa digit count.

DP POS = Decimal point position indicator is a value in the range from 11 down to 12−MDC, which indicates a digit, as given by the DPX column in the table, after which the decimal point is located.

TABLE V. Output Instruction Data Formats

Output Instruction	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
OUT1	(LSD+1) Digit 1					(LSD) Digit 0		
OUT2	(LSD+3) Digit 3					(LSD+2) Digit 2		
OUT3	(LSD+5) Digit 5					(LSD+4) Digit 4		
OUT4	(LSD+7) Digit 7					(LSD+6) Digit 6		
OUT5	(LSD+9) Digit 9					(LSD+8) Digit 8		
OUT6	(MSD) Digit 11					(MSD−1) Digit 10		
OUTSGN	Sign of Mantissa	0	0	Sign of Exponent		Link/Guard Digit		
OUTEXP	MSD EXPONENT					LSD Exponent		
OUTST	I/P Mode	Round. Mode	Not Used	Angles Mode	x As Fraction	Sign of x	x Comp. to 0	Error Status
OUTIO	IO_3	IO_2	IO_1	IO_0	1	0	0	1
OUTO	O_3	O_2	O_1	O_0	1	1	0	1
OUTR	R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0
OUTFL	O	F_2	F_1	0	IF_4	IF_3	IF_2	IF_1
OUTMDC	MDC_{MSB}	MDC_{MSB-1}	MDC_{MSB-2}	MDC_{LSB}	1	1	0	0

TABLE VI. Instruction Execution Times

All times are measured with 4 MHz at CKI and are measured from the rising edge of $\overline{WR}$ to the rising edge of INTR/RDY.

Instruction	Worst Case Execution Time (ms)	Instruction		Worst Case Execution Time (ms)
0–9	1.6	TJC	1st Byte	3.0
DP	1.5	a	2nd Byte	1.4
EE	1.5	TX2	1st Byte	3.0
CS		a	2nd Byte	1.4
PI	1.8	TXN	1st Byte	3.0
AIN1 1st Byte	0.3	a	2nd Byte	1.4
2nd Byte	0.9	TXF	1st Byte	3.0
NOP1	1.6	a	2nd Byte	1.4
NOP2	3.1	TERR	1st Byte	3.0
SMDC n	3.1	a	2nd Byte	1.4
IO n	3.1	TMNZ	1st Byte	4.0
O n	3.1	a	2nd Byte	1.4
LDR 1st Byte	3.1	TMZ	1st Byte	4.0
2nd Byte	1.5	a	2nd Byte	1.4
AIN2 1st Byte	3.1	TFI	1st Byte	1.1
2nd Byte	1.5	a	2nd Byte	0.5
OUT1–OUT6	3.0	TFL	1st Byte	1.1
OUTSGN	3.0	a	2nd Byte	0.5
OUTEXP	3.0	IMNZ	1st Byte	3.0
OUTST	3.0	a	2nd Byte	0.5
OUTIO	3.0	DMNZ	1st Byte	4.4
OUTO	3.0	a	2nd Byte	0.5
OUTR	3.0	TIF1–4	1st Byte	3.0
OUTFL	3.0	a	2nd Byte	1.5
OUTMDC	3.0			
OUT 1st Byte	3.0			

TABLE VI. Instruction Execution Times (Continued)

Instruction	Worst Case Execution Time (ms)	Instruction	Worst Case Execution Time (ms)
RAD	2.7	MCLR	5.1
DEG	2.7	EN	5.5
NRND	2.7	ROLL	8.4
RND	2.7	POP	8.6
FLP	2.7	SIN	820
SCI	2.7	COS	830
ECLR	2.7	TAN	540
SIF1–4	2.7	ARCSIN	840
RIF1–4	2.7	ARCCOS	700
ROFF	2.7	ARCTAN	410
RON	2.7	RTD	162
RIO	2.7	DTR	162
SF1	2.7	XXY	2.9
PF1	2.7	EX	440
SF2	2.7	IOX	100
PF2	2.7	SQ	18
Math/Memory Instruction	**Typical Execution Time (ms)**	SQRT	48
		LN	210
XXM	4.2	LOG	140
MS	5.7	I/X	66
MR	7.0	YX	400
M+	13.2	+	20
M−	13.2	−	20
M*	11.8	*	28
M/	60	/	66
CLRM	6.0	LSH	5.2
CLRX	3.0	RSH	2.5

Note 1: Add 0.3 ms to the execution time of any instruction which initiates number entry and is preceded by an enter instruction.

Note 2: Add 2.5 ms to the execution time of any instruction which initiates number entry and is not preceded by an enter instruction.

Note 3: Add 2.0 ms to the execution time of any instruction which terminates number entry mode.

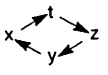
MM57409 Number Cruncher Instruction Set

Mnemonic	Operand	Hex Code	Description
BASIC NUMBER ENTRY INSTRUCTIONS			
0		00	Mantissa or exponent digits. If the previous code was EN (enter), then the digit is placed in x. If the previous code was not EN, the stack is pushed as follows: digit → x x → y y → z z → t
1		01	
2		02	
3		03	
4		04	
5		05	
6		06	
7		07	
8		08	
9		09	See Number Entry Mode Description
DP		0A	Decimal point. Digits that follow will be mantissa fraction.
EE		0B	Enter exponent. Digits that follow will be exponent digits. If this is the first data entry, a "1" will be loaded into the mantissa.
CS		0C	Change sign. The mantissa's sign is changed unless EE was the last number entry initiation, in which case the exponent's sign is changed.
PI		0D	3.14159265359 → x.
AIN1		0E	Single digit asynchronous input. See Number Entry Mode Description.
NOP1		0F	No operation.
NOP2		22	Terminate number entry, no other operation.

MM57409 Number Cruncher Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Description
MANTISSA DIGIT COUNT (MDC) CONTROL INSTRUCTIONS			
SMDC	n	7(n−1)	Set the MDC = n.n = 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C hex
I/O, O, AND PC PORT WRITE INSTRUCTIONS			
IO	n	A(n)	Write n to the 4-bit general I/O port. n = 0 through F hex.
O	n	B(n)	Write n to the 4-bit general O port. n = 0 through F hex.
LDR	a	92 a	Load the R port with "a", where "a", is 8-bit value from 00 hex through FF hex.
DATA INPUT INSTRUCTIONS			
AIN2		91	Asynchronous input 2. See Data Input Description.
DATA OUTPUT INSTRUCTIONS			
OUT1		80	Output x mantissa LSD and LSD+1.
OUT2		81	Output x mantissa LSD+2 and LSD+3.
OUT3		82	Output x mantissa LSD+4 and LSD+5.
OUT4		83	Output x mantissa LSD+6 and LSD+7.
OUT5		84	Output x mantissa LSD+8 and LSD+9.
OUT6		85	Output x mantissa MSD−1 and MSD.
OUTSGN		86	Output link/guard digit and mantissa/exponent sign digit.
OUTEXP		87	Output x exponent digit.
OUTST		88	Output 8-bit status register.
OUTIO		89	Output the state of the 4-bit general I/O port.
OUTO		8D	Output the state of the 4-bit general outut port.
OUTR		8A	Output the state of the 8-bit R port.
OUTFI		8B	Output F1, F2, and the four internal flags, IF1 through IF4.
OUTMDC		8C	Output the Mantissa Digit Count.
OUT		8F	Multiple digit output instruction. See Data Output Description.
TEST/CONDITIONAL OUTPUT INSTRUCTIONS			
TCI	a	10 a	If external test input CI = logic "1", load R with "a", where "a" is an 8-bit value from 00 hex through FF hex.
TXZ	a	11	If x = 0, load R with "a".
TXN	a	12 a	If x is negative, load R with "a".
TXF	a	13 a	If the absolute value of x is a fraction, load R with "a".
TERR	a	14	If the error flag is set, load R with "a".
TMNZ	a	16 a	If memory is not equal to 0, load R with "a".
TMZ	a	17 a	If memory equals 0, load R with "a".
TF1	a	18 a	If the external flag input 1 = logic "1", load R with "a".
TF2	a	1B a	If the external flag input 2 = logic "1", load R with "a".
IMNZ	a	19 a	Increment the mantissa contained in memory, and if the new value of the memory is not = 0, load R with "a".
DMNZ	a	1A a	Decrement the mantissa contained in memory, and if the new value of the memory is not = 0, load R with "a".
TIF	n, a	1(B+n) a	If the internal flag n = logic "1", load R with "a". n = 1,2,3,4.

MM57409 Number Cruncher Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Description
MATH INSTRUCTIONS			
CLRX		20	0 → x.
MCLR		2F	Master clear: clear all internal registers and outputs; 10 → MDC. Scientific notation mode; round to MDC on output; R port set to 03 hex and is enabled; I/O port unaffected.
EN		21	Enter and push stack. The same digit will be in x and y.
ROLL		23	Roll stack:  TL/DD/5173–17
POP		2E	Pop the stack: y → x z → y t → z 0 → t
SIN		24	sin(x) → x
COS		25	cos(x) → x
TAN		26	tan(x) → x
ARCSIN		27	arcsin(x) → x
ARCCOS		28	arccos(x) → x
ARCTAN		29	arctan(x) → x
RTD		2C	Convert the value in x from radians to degrees.
DTR		2D	Convert the value in x from degrees to radians.
XXY		30	Exchange x and y.
EX		31	e^x → x
10X		32	10^x → x
SQ		33	x^2 → x
SQRT		34	Square root(x) → x
LN		35	ln x → x
LOG		36	Log x → x
1/X		37	1/x → x
YX		38	y^x → x; 0 → t, t → z, z → y
+		39	y+x → x; 0 → t, t → z, z → y
—		3A	y−x → x; 0 → t, t → z, z → y
*		3B	y*x → x; 0 → t, t → z, z → y
/		3C	y/x → x; 0 → t, t → z, z → y
LSH		3E	Left shift x mantissa, DP unchanged, MSD in guard/link digit.
RSH		3F	Right shift x mantissa, DP unchanged, link/guard digit → MSD.
MEMORY INSTRUCTIONS			
XXM		40	Exchange x and memory.
MS		41	Store x in memory.
MR		42	Memory → x → y → z → t.
M+		43	Memory+x → memory.
M−		44	Memory−x → memory.
M*		45	Memory*x → memory.
M/		46	Memory divided by x → memory.
CLRM		47	Clear memory; 0 → memory.

MM57409 Number Cruncher Instruction Set (Continued)

Mnemonic	Operand	Hex Code	Description
MODE AND FLAG INSTRUCTIONS			
RAD		50	Set radian angular mode.
DEG		51	Set degrees angular mode.
NRND		54	Round to MDC on output disabled.
RND		55	Round to MDC on output enabled.
FLP		56	Set floating-point I/O mode.
SCI		57	Set scientific notation I/O mode.
ECLR		2B	Clear error flag.
SIF1		58	Set internal flag 1.
SIF2		5A	Set internal flag 2.
SIF3		5C	Set internal flag 3.
SIF4		5E	Set internal flag 4.
RIF1		59	Reset internal flag 1.
RIF2		5B	Reset internal flag 2.
RIF3		5D	Reset internal flag 3.
RIF4		5F	Reset internal flag 4.
OUTPUT CONTROL INSTRUCTIONS			
ROFF		60	TRI-STATE the R port.
RON		61	Enable the R port.
RIO		52	R port is enabled as high-impedance I/O.
SF1		67	Set external flag 1 high.
PF1		68	Pulse external flag 1 high. If F1 is already high, then it is reset.
SF2		69	Set external flag 2 high.
PF2		6A	Pulse external flag 2 high. If F2 is already high, then it is reset.

Section 4

COPS Applications

Section 4 Contents

Easy Logarithms for COP400

National Semiconductor Corp.
COP Brief 2

Logarithms have long been a convenient tool for the simplification of multiplication, division, and root extraction. Many assembly language programmers avoid the use of logarithms because of supposed complexity in their application to binary computers. Logarithms conjure up visions of time consuming iterations during the solution of a long series. The problem is far simpler than imagined and its solution yields, for the applications programmer, the classical benefits of logarithms:

1) Multiplication can be performed by a single addition.

2) Division can be performed by a single subtraction.

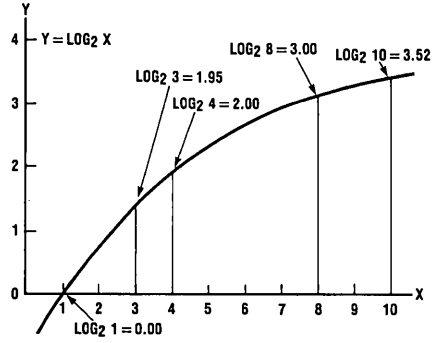
3) Raising a number to a power involves a single multiply.

4) Extracting a root involves a single divide.

When applied to binary computer operation logarithms yield two further important advantages. First, a broad range of values can be handled without resorting to floating point techniques (other than implied by the characteristic). Second, it is possible to establish the significance of an answer during the body of a calculation, again, without resorting to floating point techniques.

Implementation of $base_{10}$ logarithms in a binary system is cumbersome and unnecessary since logarithmic functions can be implemented in a number system of any base. The techniques presented here deal only with logarithms to the $base_2$.

A logarithm consists of two parts: an integer characteristic and a fractional mantissa.



TL/DD/6942–1

	CHARACTERISTIC	MANTISSA
LOG_2 3 =	1	0.95
LOG_2 4 =	2	0.00
LOG_2 8 =	3	0.00
LOG_2 10 =	3	0.52

FIGURE 1. The Logarithmic Function and Some Example Values

In *Figure 1* some points on the logarithmic curve are identified and evaluated to the $base_2$. Notice that the characteristic in each case represents the highest even power of 2 contained in the value of X. This is readily seen when binary notation is used.

X_{10}	X_2					Log_2 X Characteristic	Log_2 X Where X = Even Power of 2
	2^4	2^3	2^2	2^1	2^0		
3	0	0	0	1▲	1	1	
4	0	0	1▲	0	0	2	010.0000
8	0	1▲	0	0	0	3	011.0000
10	0	1▲	0	1	0	3	

FIGURE 2. Identification of the Characteristic

In *Figure 2* each point evaluated in *Figure 1* has been repeated using binary notation. An arrow subscript indicates the highest even power of 2 appearing in each value of X. Notice that in X = 3 the highest even power of 2 is 2^1. Thus the characteristic of the log_2 3 is 1. Where X = 10 the characteristic of the log_2 10 is 3.

To find the log_2 X is very easy where X is an even power of 2. We simply shift the value of X left until a carry bit emerges from the high order position of the register. This procedure is illustrated in *Figure 3*. This characteristic is found by counting the number of shifts required and subtracting the result from the number of bits in the register. In practice it is easier to being with the number of bits and count down once prior to each shift.

Counter for Characteristic	Value of X in Binary		
1000	0000	1000	Initial
0111	0001	0000	First Shift
0110	0010	0000	Second Shift
0101	0100	0000	Third Shift
0100	1000	0000	Fourth Shift
0011	0000	0000	Fifth Shift
Characteristic	**Mantissa**		**Final**
011.0000	0000		Log_2 X = 3.00

FIGURE 3. Conversion to $Base_2$ Logarithm by Base Shift

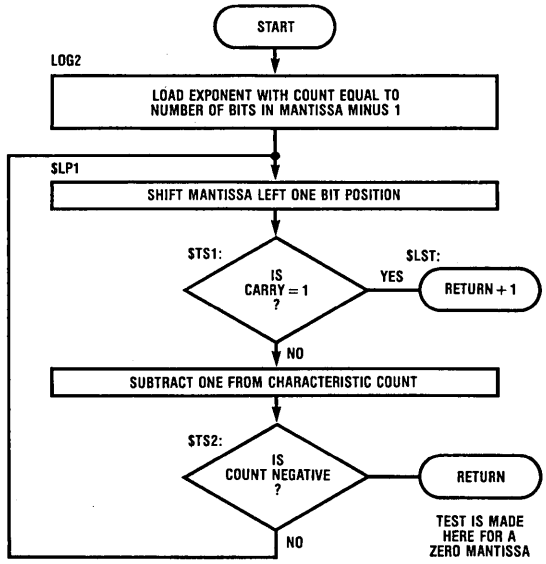
Examination of the final value obtained in *Figure 3* reveals no bits in the mantissa. The value 3 in the characteristic, however, indicates that a bit did exist in the 2^3 position of the original number and would have to be restored in order to reconstruct the original value (antilog).

The log of any even power of 2 can be found in this way:

Decimal	Binary	Log_2
128	10000000	0111.00000000
64	01000000	0110.00000000
32	00100000	0101.00000000
4	00000100	0010.00000000
2	00000010	0001.00000000
1	00000001	0000.00000000

FIGURE 4. Base$_2$ Logarithms of Even Powers of 2

A simple flow chart, and program, can be devised for generating the values found in the table and, as will be apparent, a straight line approximation for values that are not even powers of 2. The method, as already illustrated in *Figure 3,* involves only shifting a binary number left until the most significant bit moves into the carry position. The characteristic is formed by counting. Since a carry on each successive shift will yield a decreasing power of 2, we must start the characteristic count with the number of bits in the binary value (x) and count down one each shift.



TL/DD/6942–2

FIGURE 5. Log Flowchart

```
COP CROSS ASSEMBLER     PAGE:  1
LOGS

 1                      ; TITLE LOGS                ; BINARY LOGARITHMS
 2
 3          01A4                    . CHIP 420
 4
 5                                  ; ·····→ CONVERT TO LOGARITHM ←·····;
 6                      ;
 7                      ;                          RAM ASSIGNMENT
 8                      ;
 9                      ; DIGIT:    15  14  13  12  11  10  09  08  07  06  05  04  03  02  01  00
10                      ;
11                      ; REG 0     |   |   |   |   |   |   |   | CH| HM| LM|   |   |   |TEMP   |
12                      ;           +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
13                      ; REG 1     |   |   |   | CH| HM| LM|   |   |   |   |   |   |   |TEMP   |
14                      ;           +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
15                      ; REG 2     |   |   |   |   |TEMP   |   |   |   |   |   |   |   | CH| HM| LM|
16                      ;           +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
17                      ; REG 3     |   |   |   |   |   |   |   |   |   |   |TEMP   |   | CH| HM| LM|
18                      ;           +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
19                      ;
20                                  . LOCAL
21                      ;
22                      ; CH, HM, LM REPRESENT ANY THREE SEQUENTIAL MEMORY DIGITS. THEY
23                      ; MAY BE DEFINED IN ANY REGISTER. THE SYMBOLIC NOTATION CH, HM,
24                      ; AND LM ARE USED FOR ADDRESSING TO ALLOW USER FLEXIBILITY.
25                      ; UPON ENTRY TO THE ROUTINE HM AND LM CONTAIN THE HI AND LO
26                      ; OF SOME VALUE X. THE MEMORY POINTER MUST CONTAIN THE ADDRESS
27                      ; OF THE CHARACTERISTIC (CH). THE CONTENTS OF THIS LOCATION ARE
28                      ; IGNORED AND ARE LOST DURING EXECUTION.
29                      ;
30                      ; UPON EXIT CH, HM, LM CONTAIN A STRAIGHT LINE APPROXIMATION OF
31                      ; THE LOG BASE 2 OF X. CH = CHARACTERISTIC  HM = HI ORDER MANTISSA
32                      ; LM = LO ORDER MANTISSA. AN 8 BIT MEMORY AREA (TEMP) IS USED IN
33                      ; THE REGISTER OPPOSITE DURING THE CORRECTION OF A STRAIGHT
34                      LINE APPROXIMATION OF A LOG OR AN ANTILOG.
35                      ;               A TEST IS MADE FOR X = 0. IF THE VALUE OF X
36                      ; IS NOT ZERO AN INSTRUCTION IS SKIPPED UPON RETURN
37                      ; TO THE CALLING ROUTINE.
38                      ;
39                      ;                                          — EXAMPLE —
40                      ;
41                      ; SUBROUTINE CALL                                  JSR  LOG2
42                      ; RETURN HERE IF X = 0 ···→                        JP   ZERO
43                      ; RETURN HERE IF X>0 ···→                           CONTINUE
44                      ;
45                      ;
46                      ;
47                      ;
48                      ;
49                      ;
50     000      00      LOG2:       CLRA
51     001      57                  AISC        07          ; TO REG LENGTH -1.
52     002      06                  X                       ; STORE IN MEMORY.

COP CROSS ASSEMBLER     PAGE:  2
LOGS

53     003      A4      $LP1:       JSRP        SDB2        ; SET ADDRESS POINTER
54                                                          ; BACK 2 DIGITS.
55     004      A9                  JSRP        SHLR        ; RESET CARRY AND SHIFT
56                                                          ; REG LEFT ONE BIT.
57     005      20      $TS1:       SKC                     ; IS CARRY = 1 YET?
58     006      C8                  JP          $NO         ; NO — KEEP GOING.
59     007      49      $LST:       RETSK                   ; YES — FINISHED!!
60     008      05      $NO:        LD                      ; NO — LOAD COUNT IN ACC.
61     009      5F                  AISC        -1          ; SUBTRACT ONE.
62     00A      48      $TS2:       RET                     ; MANTISSA IS A 0! RETURN
63     00B      06                  X                       ; STORE CHARACTERISTIC.
64     00C      C3                  JP          $LP1        ; DO IT AGAIN!
65
66
67
68
69
70                      ; 2 ROUTINES ARE CALLED FROM THE SUBROUTINE PAGE BY THIS
71                      ; PROGRAM: SDB2, SHLR.
72
```

TL/DD/6942–5

FIGURE 6

The program shown develops the $\log_2$ of any even power of 2 by shifting and testing as previously described. Examine what happens to a value of X that is not an even power of 2. In *Figure 7,* the number 25 is converted to a base 2 log.

$$25_{10} = 00011002_2$$

Shift left until carry = 1

Characteristic	Carry	Mantissa	Log_2
0100	1	10010000.10010000	

Figure 7. Straight Line Approximation of $Base_2$ Log

The resulting number when viewed as an integer characteristic and a fractional mantissa is 4.5625_{10}. The fraction 0.5625 is a straight line approximation of the logarithmic curve between the correct values for the $base_2$ logs of 2^4 and 2^5. The accuracy of this approximation is sufficient for many applications. The error can be corrected, as will be seen later in this discussion, but for now let's look at the problem of exponents or the conversion to an antilog.

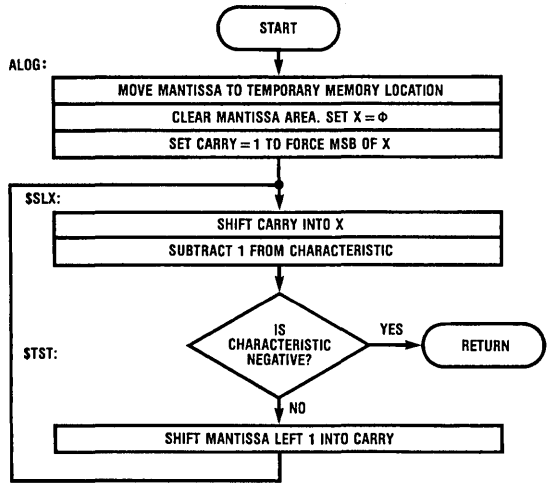
To reconstruct the original value of X, find the antilog, requires only restoration of the most significant bit and then its alignment with the power of 2 position indicated by the characteristic. In the example, approximation ($\log_2$ 25 = 0100.1001) restoration of MSB can be accomplished by shifting the mantissa (only) one position to the right. In the process a one is shifted into the MSB position.

Approximation of Log_2 X	Restoration of MSB
Char. Mantissa	Char. Mantissa
0100.10010000	0100.11001000

The value of the characteristic is 4 so the mantissa must be shifted to the right until MSB is aligned with the 2^4 position.

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

The completion of this operation restores the value of X (X = 25) and is the procedure used to find an antilog. *Figure 8* is a flow chart for finding an antilog using this procedure. Ths implementation in source code is shown in *Figure 9.*



TL/DD/6942–3

FIGURE 8. Flow Chart for Conversion to Antilog

```
COP CROSS ASSEMBLER PAGE 3
LOGS

                        . FORM          ; ------> CONVERT TO ANTILOG <------ ;
                        ; THE FOLLOWING SUBROUTINE CONVERTS THE STRAIGHT LINE
                        ; THE APPROXIMATION OF A BASE 2 LOGARITHM TO ITS CORRESPONDING
                        ; ANTILOG. UPON EXIT FROM THE ROUTINE THE CONTENTS OF CH
                        ; WILL BE EQUAL TO THE HEXADECIMAL VALUE OF 'ΦF'.
                                        . LOCAL
00D  A4   ALOG:         JSRP    SDB2    ; SET ACC TO 0.
00E  00   CLRA                          ; CLEAR MANTISSA AREA.
00F  36                 X       03      ; AND MOVE MANTISSA TO
010  34                 XIS     03      ; TEMPORARY STORAGE.
011  00                 CLRA            ; LEAVE POINTER AT LO
012  36                 X       03      ; ORDER OF MANTISSA.
013  37                 XDS     03
014  22                 SC              ; RESTORE MSB OF X.
015  D8                 JP      $SLX
01   A9   $SLM:         JSRP    SHLR    ; SHIFT REMAINDER
                                        ; LEFT INTO CARRY.
017  A3                 JSRP    SDR2    ; MOVE BACK 2 DIGITS.
018  AA   $SLX:         JSRP    SHLC    ; SHIFT X LEFT 1.
019  05                 LD              ; LOAD CHARACTERISTIC.
01A  5F   $TST:         AISC    -1      ; CHARACTERISTIC -1.
01B  48   $LST:         RET             ; IF NO CARRY — FINIS.
01C  36                 X       03      ; STORE REMAINDER AND MOVE
                                        ; DOWN ONE REGISTER.
01D  A4                 JSRP    SDB2    ; MOVE BACK 2 DIGITS.
01E  D6                 JP      $SLM    ; DO IT AGAIN.

                        ; 4 ROUTINES ARE CALLED FROM THE SUBROUTINE PAGE BY THIS
                        ; PROGRAM: SDB2, SDR2, SHLR, SHLC.
```

TL/DD/6942–6

FIGURE 9

Using the linear approximation technique just described, some error will result when converting any value of X that is not an even power of 2.

Figure 10 contains a table of correct base 2 logarithms for values of X from 1 through 32 along with the error incurred for each when using linear approximation. Notice that no error results for values of X that are even powers of 2. Also notice that the error incurred for multiples of even powers of 2 of any given value of X is always the same.

Value of X	Error
5	0.12
$2 \times 5 = 10$	0.12
$4 \times 5 = 20$	0.12
3	0.15
$2 \times 3 = 6$	0.15
$4 \times 3 = 12$	0.15
$8 \times 3 = 24$	0.15

X	Hexadecimal Log Base	Linear Approximation of Log Base 2	Error Hexadecimal	$E_M - 1 + \frac{EM - EM - 1}{2}$
1	0.00	0.00	0.00	
2	1.00	1.00	0.00	
3	1.95	1.80	0.15	
4	2.00	2.00	0.00	
5	2.52	2.40	0.12	
6	2.95	2.80	0.15	
7	2.CE	2.C0	0.0E	
8	3.00	3.00	0.00	
9	3.2B	3.20	0.0B	
10	3.52	3.40	0.12	
11	3.75	3.60	0.15	
12	3.95	3.80	0.15	
13	3.B3	3.A0	0.13	
14	3.CE	3.C0	0.0E	
15	3.E8	3.E0	0.08	
16	4.00	4.00	0.00	
17	4.16	4.10	0.06	0.03
18	4.2B	4.20	0.0B	0.09
19	4.3F	4.30	0.0F	0.0D
20	4.52	4.40	0.12	0.11
21	4.67	4.50	0.17	0.15
22	4.75	4.60	0.15	0.16
23	4.87	4.70	0.17	0.16
24	4.95	4.80	0.15	0.16
25	4.A4	4.90	0.14	0.15
26	4.B3	4.IA0	0.13	0.14
27	4.C1	4.B0	0.11	0.12
28	4.CE;	4.C0	0.0E	0.10
29	4.DB	4.D0	0.0B	0.0D
30	4.E8	4.E0	0.08	0.0A
31	4.F4	4.F0	0.04	0.06
32	5.00	5.00	0.00	0.02
33		5.1-		

FIGURE 10. Error Incurred by Linear Approximation of Base 2 Logs

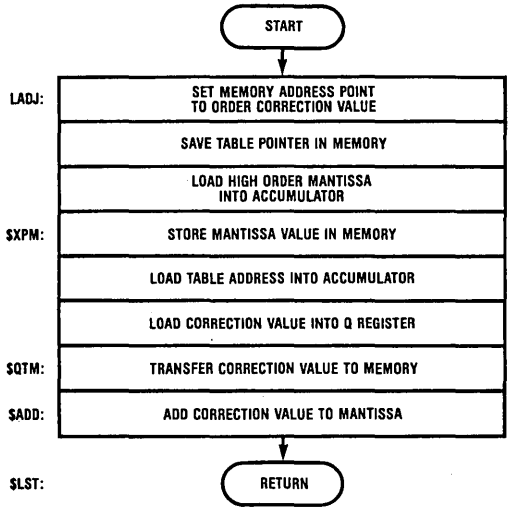
An error that repeats in this way is easily corrected using a look-up table. The greatest absolute error will occur for the least value of X not an even power of 2, X = 3, is about 8%. A 4 point correction table will eliminate this error but will move the greatest uncompensated error to X = 9 where it will be about 4%. This process continues until at 16 correction points the maximum error for the absolute value of the logarithm is less than 1 percent. This can be reduced to 0.3 percent by distributing the error. Interpolated error values are listed in *Figure 10* and are repeated in *Figure 11* as a binary table.

High Order 4 Mantissa Bits	Binary Correction Value	Hexadecimal Correction Value
0000	0000 0000	0 0
0001	0000 1001	0 9
0010	0000 1101	0 3
0011	0001 0001	1 1
0100	0001 0101	1 5
0101	0001 0110	1 6
0110	0001 0110	1 6
0111	0001 0110	1 6
1000	0001 0101	1 5
1001	0001 0100	1 4
1010	0001 0010	1 2
1011	0001 0000	1 0
1100	0000 1101	0 D
1101	0000 1010	0 A
1110	0000 0110	0 6
1111	0000 0010	0 2

FIGURE 11. Correction Table for L_2 X Linear Approximations

Notice in *Figure 10* that left justification of the mantissa causes its high order four bits to form a binary sequence that always corresponds to the proper correction value. This works to advantage when combined with the COP400 LQID instruction. LQID implements a table look-up function using the contents of a memory location as the address pointer. Thus we can perform the required table look-up without disturbing the mantissa.

Figure 12 is the flow chart for correction of a logarithm found by linear approximation. *Figure 13* is its implementation in COP400 assembly language. Notice that there are two entry points into the program. One is for correction of logs (LADJ:), the other is for correction of a value prior to its conversion to an antilog (AADJ:).



TL/DD/6942–4

FIGURE 12. Flow Chart for Correction of a Value Found by Straight Line Approximation

```
COP CROSS ASSEMBLER    PAGE:  4
LOGS

                              . FORM          ; ------> ADJUST VALUE OF LOGARITHM <------ ;
                              . LOCAL

                              ; THE FOLLOWING TABLE IS USED DURING THE CORRECTION OF VALUES
                              ; FOUND BY STRAIGHT LINE APPROXIMATION. IT IS PLACED HERE IN
                              ; ORDER TO ALIGN ITS BEGINNING ELEMENT WITH A ZERO ADDRESS AS
                              REQUIRED BY THE LQID INSTRUCTION.

 01F  44                      NOP                                  ; REGISTER WITH ZERO ADDRESS.
 020  03          TPLS:       . WORD      03,09,0D,011
 021  09
 022  0D
 023  11
 024  15          . WORD      015,016,016,016
 025  16
 026  16
 027  16
 028  15                      . WORD      015,014,012,010
 029  14
 02A  12
 02B  10
 02C  0D                      . WORD      0D,0A,06,02
 02D  0A
 02E  06
 02F  02

                              ; THE FOLLOWING SUBROUTINE ADJUSTS THE VALUE OF A BASE 2
                              ; LOGARITHM FOUND BY STRAIGHT LINE APPROXIMATION. THE
                              ; CORRECTION TERMS ARE TAKEN FROM THE TABLE ABOVE. THE
                              ; SUBROUTINE HAS 2 ENTRY POINTS:
                              ;
                              ;           LADJ: — ADJUSTS A VALUE DURING CONVERSION TO A LOG
                              ;
                              ;           AADJ: — ADJUSTS A VALUE DURING CONVERSION TO ANTILOG
                              ;
                              ; THE CARRY FLAG IS SET UPON ENTRY TO DISTINGUISH BETWEEN LOG
                              ; (C = 1) AND ANTILOG (C = 0) CONVERSIONS. DURING A LOGARITHM
                              ; CONVERSION THE VALUE FOUND IN THE ABOVE TABLE IS ADDED TO
                              ; THE MANTISSA. DURING AN ANTILOG CONVERSION THE VALUE FOUND
                              ; IN THE ABOVE TABLE IS SUBTRACTED FROM THE MANTISSA.

 030  32          AADJ:       RC                                   ; C = 0 FOR ANTILOG
 031  F3                      JP          $LD                      ; CONVERSION.
 032  22          LADJ:       SC                                   ; C =  FOR LOG2 ADJ.
 033  05          $LD         LD                                   ; MOVE ADDRESS POINTER BACK
 034  07                      XDS                                  ; ONE LOCATION.
 035  05                      LD                                   ; LOAD CONTENTS OF HI MANTISSA
 036  37                      XDS         03                       ; AND STORE IT IN THE LO ORDER
 037  06                      X                                    ; OF THE TEMP MEMORY LOCATION.
 038  00                      CLRA                                 ; SET TABLE POINTER
 039  52                      AISC        TBL                      ; (ACC) TO TABLE ADDRESS.

COP CROSS ASSEMBLER    PAGE:  5
LOGS

 03A  BF                      LQID                                 ; LOAD CORRECTION VALUE TO Q.
 03B  332C        $GTM:       COMA                                 ; TRANSFER Q REGISTER
 03D  04                      XIS                                  ; CONTENTS TO MEMORY.
 03F  07                      XDS
 03F  20                      SKC                                  ; ANTILOG?

 040  80                      JSRP        COMP                     ; YES — COMPLIMENT.
 041  98          $ADD:       JSRP        ADRO                     ; ADD CORRECTION VALUE
                                                                   ; TO MANTISSA.
 042  35                      LD          03                       ; SET POINTER TO
 043  48          $LST:       RET         ; CHARACTERISTIC AND
                                                                   ; RETURN.

                              ; 2 ROUTINES ARE CALLED FROM THE SUBROUTINE PAGE BY THIS
                              ; PROGRAM: COMP, ADRO

      0020                    V1 = TPLS&OFF
      0002                    TBL = V1/16
```

TL/DD/6942–7

FIGURE 13

Subroutines Used by the Log and Antilog Programs

```
COP CROSS ASSEMBLER     PAGE:  6
LOGS

 172                                    . FORM

 173          0080      . PAGE 02                  ; ·····→ SUBROUTINES ←····· ;
 174
 175                    ; THE FOLLOWING ROUTINES RESIDE ON THE SUBROUTINE PAGE. THEY
 176                    ; ARE CALLED BY THE LOGS PROGRAM BUT ARE GENERAL PURPOSE IN
 177                    ; NATURE AND FUNCTION AS UTILITY ROUTINES.
 178
 179
 180
 181                                    ; ·····→ COMPLEMENT 8 BITS ←····· ;
 182
 183                                    . LOCAL
 184
 185                    ; THIS ROUTINE FORMS IN MEMORY THE 2'S COMPLEMENT OF THE TWO
 186                    ; ADJACENT DIGITS IDENTIFIED BY THE ADDRESS POINTER. THE
 187                    ; CONTENTS OF THE ADDRESS POINTER ARE NOT ALTERED.
 188
 189                    ; THERE ARE TWO ENTRY POINTS:
 190                    ;
 191                    ; COP: COMPLEMENT 8 BITS.
 192                    ;
 193                    ; CMPE: EXTEND THE COMPLEMENT TO AN ADDITIONAL 8 BITS
 194                    ;
 195
 196  080     22       COMP:          SC
 197  081     00       CMPE:          CLRA                          ; SET MINUEND = 0
 198  082     06                      X                             ; AND STORE IN MEMORY.
 199  083     10                      CASC
 200  084     44                      NOP                           ;
 201  085     04                      XIS                           ;
 202  086     00                      CLRA                          ; SET MINUEND = 0
 203  087     06                      X                             ; AND STORE IN MEMORY.
 204  083     10                      CASC                          ;
 205  089     44                      NOP
 206  08A     04                      XIS                           ;
 207  08B     44                      NOP                           ; AVOID SKIP IF DIGIT 15.
 208  08C     A4                      JP             SDB2           ; RETURN THRU SDB2
 209                                                                ; TO RESTORE POINTER.
 210
 211
 212
 213                    ; ·····→ ADD 8 BITS IN ADJACENT REGISTERS ←····· ;
 214
 215                                    . LOCAL
 216
 217
 218
 219                    ; THIS ROUTINE ADDS TWO BINARY DIGITS (8 BITS) FROM ANY REGISTER
 220                    ; TO THE CORRESPONDING TWO BINARY DIGITS IN EITHER REGISTER
 221                    ; IMMEDIATELY ADJACENT. THERE ARE THREE ENTRY POINTS:
 222                    ;
 223                    ;              LADR: — RESET CARRY AND ADD 2 DIGIT PAIRS
```

TL/DD/6942-8

```
COP CROSS ASSEMBLER     PAGE:  7
LOGS

                                ;               LADD: — ADD 2 DIGIT PAIRS WITH UNMODIFIED CARRY
                                ;               ADD1: — ADD 2 SINGLE DIGITS WITH UNMODIFIED CARRY
                                ;
                                ;
                                ;
                                ;
0008D   32      LADR:   RC                              ; RESET CARRY PRIOR TO ADD.
0008E   15      LADD:   :D      01                      ; LD ADDEND AND MOVE TO ADJ REG
0008F   30              ASC                             ; ADD AUGEND.
00090   44              NOP                             ; AVOID CARRY!
00091   14              XIS     01                      ; STORE SUM AND MOVE TO ADDEND
00092   15      ADD1:   LD      01                      ; REPEAT PROCESS
00093   30              ASC                             ; FOR
00094   44              NOP                             ; HIGH ORDER
00095   14              XIS     01                      ; DIGIT.
00096   44              NOP                             ; AVOID SKIP IF DIGIT 15.
00097   48      $LST:   RET                             ; FINISHED — RETURN!!!!
                                ;
                                ;
                                ;
                                ;
                                ;               ;------> ADD 8 BITS IN OPPOSITE REGISTERS <------ ;
                                ;
                                ;               . LOCAL
                                ;
                                ;
                                ;
                                ; THIS ROUTINE ADDS TWO BINARY DIGITS (8BITS) FROM ANY REGISTER
                                ; TO THE CORRESPONDING TWO BINARY DIGITS IN EITHER REGISTER
                                ; DIRECTLY OPPOSITE. THERE ARE THREE ENTRY POINTS:
                                ;
                                ;               ADR0: — RESET CARRY AND ADD 2 DIGIT PAIRS
                                ;               ADD0: — ADD 2 DIGIT PAIRS WITH UNMODIFIED CARRY
                                ;               AD01: — ADD 2 SINGLE DIGITS WITH UNMODIFIED CARRY
                                ;
                                ;
                                ;
                                ;
00098   32      ADR0:   RC                              ; RESET CARRY PRIOR TO ADD.
00099   35      ADD0:   LD      03                      ; LD ADDEND AND MOVE TO OPP REG
0009A   30              ASC                             ; ADD AUGEND.
0009B   44              NOP                             ; AVOID CARRY!
0009C   34              XIS     03                      ; STORE SUM AND MOVE TO ADDEND.
0009D   15      AD01:   LD      01                      ; REPEAT PROCESS
0009E   30              ASC                             ; FOR
0009F   44              NOP                             ; HIGH ORDER
000A0   34              XIS     03                      ; DIGIT.
000A1   44              NOP                             ; AVOID SKIP IF DIGIT 15.
000A2   48      $LST:   RET                             ; FINISHED — RETURN!!!!
                                ;
                                ;
                                ;
                                ;               ;------> SET DIGIT ADDRESS BACK TWO <------ ;
                                ;
```

TL/DD/6942-9

```
COP CROSS ASSEMBLER    PAGE:  8
LOGS

                                        . LOCAL

                        ; THIS ROUTINE SUBTRACTS 2 FROM THE CONTENTS OF THE
                        ; DIGIT POINTER (B REGISTER). THE CONTENTS OF THE
                        ; ACCUMULATOR ARE LOST IN THE PROCESS. THE USE OF
                        ; SDB2 ALLOWS ADDRESSING WITHIN THE LOGS SUB
                        ; ROUTINE TO BE RELATIVE TO THE CONTENTS OF THE
                        ; ADDRESS POINTER (B REGISTER) UPON ENTRY.
                        ; SDB2 IS COMMONLY USED IN BYTE OPERATIONS TO RESTORE THE
                        ; DIGIT POINTER TO THE LOW ORDER POSITION.
                        ; THERE ARE TWO ENTRY POINTS:
                        ;
                        ; SDR2:      SET DIGIT ADDRESS BACK 2 AND MOVE TO OPPOSITE REGISTER.
                        ;
                        ; SDB2: SET DIGIT ADDRESS BACK 2 RETAINING PRESENT REGISTER.


 0A3  35    SDR2:       LD      03          ; MOVE TO OPPOSITE REGISTER.
 0A4  4E    SDB2:       CBA                 ; PLACE DIGIT COUNT IN ACC.
 0A5  5E                AISC    -2          ; SUBTRACT 2.
 0A6  44                NOP                 ; SHOULD ALWAYS SKIP.
 0A7  50                CAB                 ; PUT DIGIT COUNT BACK.
 0A8  48                RET                 ; FINISHED — RETURN!!


                        ; ·····→ SHIFT LEFT ←····· ;

                                        . LOCAL

                        ; THIS ROUTINE SHIFTS LEFT THE CONTENTS OF TWO MEMORY
                        ; LOCATIONS ONE BIT. THERE ARE THREE ENTRY POINTS:

                        ;                   SHLR: RESETS THE CARRY BEFORE SHIFTING
                        ;                         IN ORDER TO FILL THE LOW ORDER
                        ;                         BIT POSITION WITH A 0.
                        ;
                        ;                   SHLC: SHIFTS THE STATE OF THE CARRY INTO
                        ;                         THE LOW ORDER BIT POSITION.
                        ;
                        ;                   SHL1: SHIFTS LEFT THE CONTENTS OF ONLY
                        ;                         ONE MEMORY LOCATION. THE STATE
                        ;                         OF THE CARRY IS SHIFTED INTO THE
                        ;                         LOW ORDER POSITION OF MEMORY.


 0A9  32    SHLR:       RC                  ; CLEAR CARRY PRIOR TO SHIFT.
 0AA  05    SHLC:       LD                  ; LOAD FIRST MEM DIGIT.
 0AB  30                ASC                 ; DOUBLE IT.
 0AC  44                NOP                 ; AVOID SKIP.
 0AD  04                XIS                 ; STORE SHIFTED DIGIT.
 0AE  05    SHL1:       LD                  ; LOAD NEXT MEM DIGIT.
 0AF  30                ASC                 ; DOUBLE IT TOO.

COP CROSS ASSEMBLER    PAGE:  9
LOGS

 0B0  44                NOP                 ; AVOID SKIP, IF ANY
 0B1  04                XIS                 ; STORE SHIFTED DIGIT.
 0B2  48    $LST:       RET                 ; FINISHED — RETURN!


                                        . END
```

TL/DD/6942–10

L-Bus Considerations

National Semiconductor Corp.
COP Brief 4

L-BUS CONSIDERATIONS

Users of the COP400 family of microcontrollers should be aware that certain outputs exhibit peculiarities that preclude their use as clocks for edge sensitive devices such as flip-flops, counters, shift registers, etc. All family members excluding the COP410L and COP411L may generate false states on L_0–L_7 during the execution of the CAMQ instruction. *Figure 1* contains a short program to illustrate this.

```
START:
        CLRA            ;ENABLE THE Q
        LEI   4         ;REGISTER TO L LINES
        LBI   TEST
        STII  3
        AISC  12
LOOP:
        LBI   TEST      ;LOAD Q WITH X'C3
        CAMQ
        JP    LOOP
```

FIGURE 1. Glitch Test Program

In this program the internal Q register is enabled onto the L lines and a steady bit pattern of logic highs is outpout on L_0, L_1, L_6, L_7, and logic lows on L_2–L_5 via the two-byte CAMQ instruction. Timing constraints on the device are such that the Q register may be temporarily loaded with the second byte of the CAMQ opcode (X'3C) prior to receiving the valid data pattern. If this occurs, the opcode will ripple onto the L lines and cause negative-going glitches on L_0, L_1, L_6, L_7, and positive glitches on L_2–L_5. Glitch durations are under 2 microseconds, although the exact value may vary due to data patterns, processing parameters, and L line loading. These false states are peculiar only to the CAMQ instruction and the L lines. The user should expeience no difficulty interfacing with other COP420 outputs such as G_0–G_3 and D_0–D_3 to edge sensitive components.

Software and Opcode Differences in the COP444L Instruction Set

National Semiconductor Corp.
COP Brief 5

The COP444L is essentially a COP420L with double RAM and ROM. Because of this increased memory space certain instructions have expanded capability in the COP444L. Note that there are no new instructions in the COP444L and that all instructions perform the same operations in the COP444L as they did in the COP420L. The expanded capability is merely to allow appropriate handling of the increased memory space. The affected instructions are:

JMP	a	(a = address)
JSR	a	(a = address)
LDD	r,d	(r,d = RAM address Br,Bd)
XAD	r,d	(r,d = RAM address Br,Bd)
LBI	r,d	(r,d = RAM address Br,Bd; only two byte form of the instruction affected)
XABR		

The JMP and JSR instructions are modified in that the address a may be anywhere within the 2048 words of ROM space. The opcodes are as follows:

JMP | 0110 | 0 | $a_{10:9:8}$ |
| $a_{7:0}$ |

JSR | 0110 | 1 | $a_{10:9:8}$ |
| $a_{7:0}$ |

The LDD, XAD, and two byte LBI are modified so that they may address the entire RAM space. The opcodes are as follows:

LDD | 0110 | 0011 |
| 0 | r | d |

XAD | 0010 | 0011 |
| 1 | r | d |

LBI | 0011 | 0011 |
| 1 | r | d |

The XABR instruction change is transparent to the user. The opcode is not changed nor is the function of the instruction. The change is that values of 0 through 7 in A will address registers in the COP444L—i.e. the lower three bits of A become the Br value following the instruction. In the COP420L, the lower two bits of A became the Br value following an XABR instruction.

Note that those instructions which have an exclusive-or argument (LD, X, XIS, XDS) are not affected. The argument is still two bits of the opcode. This means that the exclusive-or aspect of these instructions works within blocks of four registers. It is not possible to toggle Br from a value between 0 and 3 to a value between 4 and 7 by means of these instructions.

There are no other software or opcode differences between the COP444L and the COP420L. Examination of the above changes indicates that the existing opcodes for those instructions have merely been extended. There is no fundamental change.

RAM Keep-Alive

National Semiconductor Corp.
COP Brief 6

A COPS™ application is a small scale computer system and the design of a power shut-down is not trivial. During the time that power is available, but out of the designed operating range, the system must be prevented from doing anything to harm protected data. This will typically involve some type of external protection of timing circuit.

There is an option on the COP420, 420L, and 410L parts called "RAM Keep-Alive" that provides a separate power supply to the RAM area of the chip via the CKO pin. The application of power to the RAM while the remainder of the chip has been powered down via V_{CC} will keep the RAM "alive".

However, the integrity of data in the RAM is not only a function of power but is also influenced by transient conditions as power is removed and reapplied. During power-on, the Power On Reset (POR) circuit will keep transients from causing changes in the RAM states. The condition of power loss will have some probability of data change if external control is not used.

At some point below the minimum operating voltage certain gates will no longer respond properly while others may still be functional until a much lower voltage. During this transition time any false signal could cause a false write to one or more cells. Another effect could be to turn on multiple address select lines causing data destruction.

Testing the rate of data change is very difficult because it must be done on a statistical basis with many turn/on-turn/off cycles. Two factors have a major bearing on the numbers derived by testing. One is to call any change in a related data block a failure, even though more than one bit in that block may have changed (this latter case may well be due to the "address select mode"). The second factor is that without massive instrumentation it is impossible to examine the data after each power cycle. Indeed, to do so might have caused errors!

By running the power cycle for a period of time and then looking for changes, one could overlook multiple changes thus reducing the error rate. This has been minimized by more frequent checking which indicates that the errors are spread out randomly over time.

With a power supply that drops from 4.5 to 2V in approximately 100 ms, the drop-out rate is 1 in 5k to 6k power cycles. Reducing the voltage fall time will cause an improvement in the number of cycles per drop-out. This will reach a limit condition of a very high number (1 per 1 million?) when the power falls within one instruction cycle (4–10 μs for the 420, 15–40 μs for the "L" parts). Attaining very rapid fall time may cause problems due to the lack of decoupling/bypass capacitance. By inserting an electronic switch between the regulator and V_{CC} of the COP chip one might be able to meet this type of fall time. By implication some type of sensing is required to cause the switching.

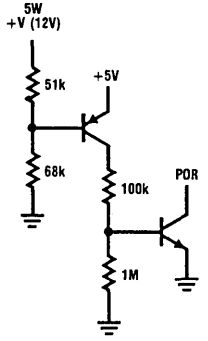
The desirable approach is to force the COP reset input to zero before the voltage falls below 4.5V. This provides a drop out rate of approximately 1 in 50k for the "L" parts and 1 in 100k for the 420. By also stopping the clock of the "L" parts they can achieve a drop-out rate similar to the 420. While not perfect, the number of cycles between data error should be considered with respect to the needs of the application.

The external circuitry to control the chip during the power transition has several implementations each one being a function of the application. The simplest hardware is found in a battery powered (automotive) application. The circuit must sense that the switched 12V is falling (e.g., at some value much below 12V and still greater than 5V). This can be done by using the unswitched 12V as a reference for a divider to a nominal voltage of 8V. As the switched 12V drops below the reference a detector will turn on a clamp transistor to a series switch, the POR, and/or the clock circuit *(Figure 1)*. It should be noted that this draws current during the absence of the switched 12V circuit.

In non-automotive usage a similar circuit can be used where there is a stable reference voltage available to use with the comparator/clamp. Thus a 3.6V rechargable Ni-Cad battery could be used as the reference voltage and V_{RAM} if the appropriate divider is used to level shift to this operating range.

In AC line-powered applications, a similar method could be used with the raw DC being sensed for drop. Another method would be to sense that the line had missed 2–3 cycles either by means of a charge pump or peak detection technique. This will provide the signal to turn on the clamp. One must make this faster than the time to discharge the output capacitance of the power supply, thus assuring that the clamp has performed its function before the supply falls below spec value.

In conclusion, to protect the data stored in RAM during power-off cycle, the POR should go low before the V_{CC} power drops below spec and come up after V_{CC} is within spec. The first item must be handled with an external circuit like *Figure 1* and the latter by an RC per the data sheet.



TL/DD/6946–1

FIGURE 1

National Semiconductor Corporation

COP420-QRX/N Preprogrammed Single-Chip Microcontroller for Musical Organ

Features and Functions

Play Mode: Twenty-five musical keys and 25 LEDs are provided to denote F to F with half notes in between. All the keys and LEDs are directly detected and driven by the microprocessor. Depression of the key will give the corresponding musical note and light up the corresponding LED.

Clear: Memory is provided to store a played tune. Depression of the CLEAR key erases the memory and the microprocessor is ready to store new musical notes. A maximum of 28 notes can be stored where each note can be of one to eight musical beats. (Two bytes of memory are required to store one musical note. Any note longer than eight muscial beats will require additional memory space for storage.)

Playback: Depression of this button will playback the tune stored in the memory since last "clear."

Preprogrammed Tunes: There are ten preprogrammed tunes (each has an average of 55 notes) masked in the chip. Any tune can be recalled by depressing the "Tune Button" followed by the corresponding "Sharp Key."

Learn Mode: This mode is for the player to learn the ten preprogrammed tunes. By pressing the "Learn Button" followed by the corresponding "Sharp Key," the LEDs will be lighted up one by one to indicate the notes of the selected tune. The LED will remain "on" until the player presses the correct musical key; the LED for the next note will then be lighted up.

Pause: In addition to the 25 musical keys, there is a special pause key. The depression of this key generates a blank note to the memory.

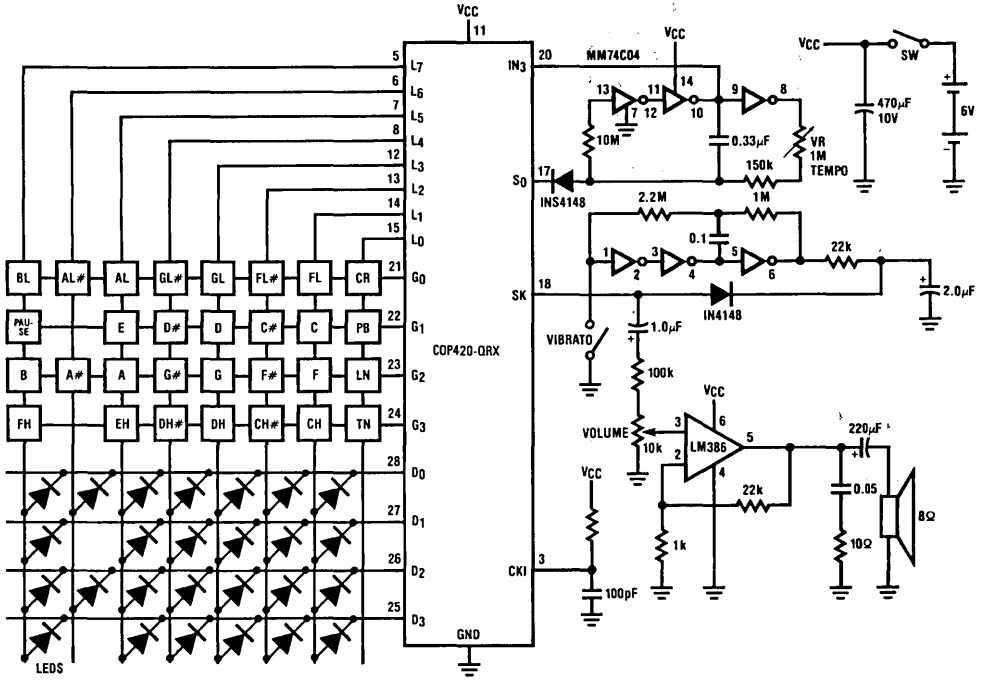
Note: In the Learn Mode when playing "Oh Susanna," the pause key must be used.

Tempo: This is a control input to the musical beat time oscillator for varying the speed of the musical tunes.

Vibrato: This is a switch control to vary the frequency vibration of the note.

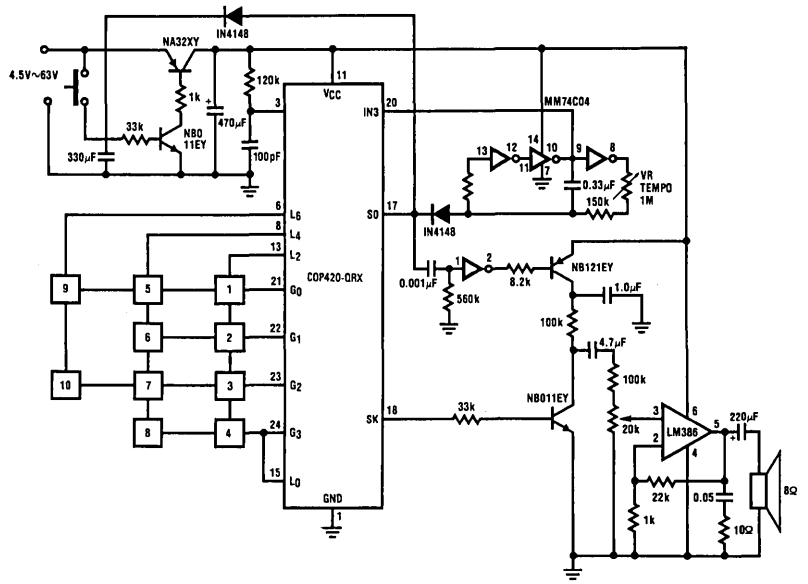
Tune Listing: The following is a listing of the ten preprogrammed tunes: 1) Jingle Bells, 2) Twinkle, Twinkle Little Star, 3) Happy Birthday, 4) Yankee Doodle, 5) Silent Night, 6) This Old Man, 7) London Bridge Is Falling Down, 8) Auld Lang Syne, 9) Oh Susanna, 10) Clementine.

Circuit Diagram of COP420 Musical Organ



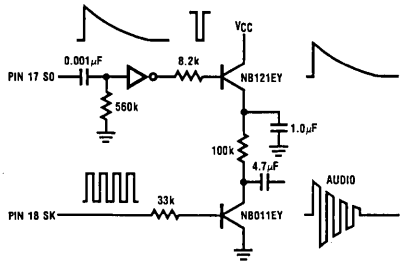
TL/DD/6923-1

Music Box Application with Direct Key Access



TL/DD/6923-2

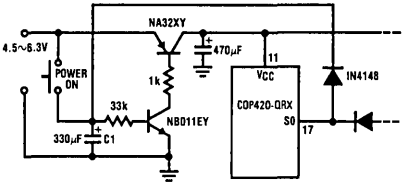
Bell Sound Circuit



TL/DD/6923-3

This additional circuit provides tinkling effect for the musical note.

Auto Power Shut-Off Circuit



TL/DD/6923-4

This circuit automatically turns off the musical organ if none of the keys are pressed within approximately 30 seconds.

Analog to Digital Conversion Techniques With COPS™ Family Microcontrollers

National Semiconductor Corp.
COP Note 1
Leonard A. Distaso

TABLE OF CONTENTS

1.0 Introduction

A variety of techniques for performing analog to digital conversion are presented. The COP420 microcontroller is used as the control element in all cases. However, any of the COPS family of microcontrollers could be used with only minor changes in some component values to allow for different instruction cycle times.

Indirect analog to digital converters are composed of three basic building blocks:

- D/A Converter
- Comparator
- Control logic

In a software driven system the D/A converter and comparator are present but the control logic is replaced by instruction sequences. There are a variety of software/hardware techniques for implementing A/D converters. They differ primarily in their approach to the included D/A. There are two primary approaches to the digital to analog conversion which can in turn be divided into a number of sub-categories:

- D/A as a function of weight closures
  - — R/2R ladder
  - — Binary weighted ladder
- D/A as function of time
  - — RC exponential charge
  - — Linear charge/discharge (dual slope)
  - — Pulse width modulation

These techniques should be generally familiar to persons skilled in the electronic art. The objective here is to illustrate the application of these established methods to a low cost system with a COPS microcontroller as the intelligent control element. Circuit configurations are provided as well as the appropriate flow charts and code to implement the function.

Some mathematical and theoretical analysis is presented as an aid to understanding the various techniques and their limits. However, it is not the purpose here to provide a definitive theoretical analysis of the analog to digital conversion process or of the various techniques described.

2.0 Simple Capacitor Charge Time Measurement

2.1 BASIC APPROACH

General

Perhaps the simplest means to perform an analog to digital conversion is to charge a capacitor until the capacitor voltage is equal to the unknown voltage. The capacitor voltage and the unknown are compared by means of a standard analog comparator. The unknown is determined simply by counting, in the microcontroller, the amount of time it takes for the charge on the capacitor to reach a value equal to the unknown voltage. The capacitor voltage is given by the standard capacitor charge equation:

$$V_C = V0 + [V1 - V0][1 - e^{**}(-t/RC)]$$

where: V_C = capacitor voltage
V0 = "dischage voltage" — low level voltage
V1 = high level voltage

The most obvious problem with this method, from the standpoint of software implementation, is the nonlinearity of the

relationship. This can be circumvented in several ways. First of all, a routine to calculate the exponential can be implemented. This, however, usually requires too much code if the exponential routine is not otherwise required in the program. Alternatively, the range of input voltages can be restricted so that only a portion of the capacitor charge curve — which can be approximated with a linear relationship or with some minor straight time curve fitting — is used. Finally, a look up table can be used which will effectively convert the measured time to the appropriate voltage. The look up table has the advantage that all the math can be built into the table, thereby simplifying matters significantly. If arithmetic routines are going to be used, it is clear that the relationship is simplified if V0 is 0V because it then drops out the equation.

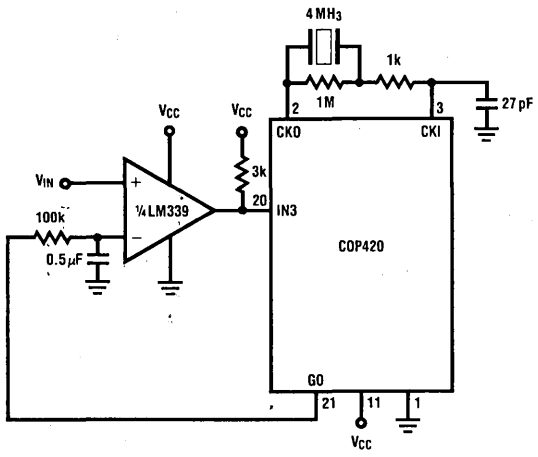
BASIC CIRCUIT IMPLEMENTATION

The circuit in *Figure 1* is the basic implementation of the capacitor charge method of A/D conversion. The selection of input and output used is arbitrary and is dictated by general system considerations. V0 is the "0" level of the G output and V1 is the "1" level of the output. The technique is basically to discharge the capacitor to V0 (which is ideally ground) and then to apply V1 and increment an internal counter until the comparator changes state. The flow chart and code for this implementation are shown in *Figure 2*.

ACCURACY CONSIDERATIONS

The levels reached by the microcontroller output constitute one of the more significant problems with this basic implementation. The levels of V1 and V0 are not V_{CC} and ground as would be desired. The level is defined by the load on the output, the value of V_{CC}, and the device itself. Furthermore, these levels are likely to change from device to device and over temperature. To be sure, the output values will be at least those given in the data sheet, but it must be remembered that those values are minimum high voltages and maximum low voltages. Typically, the high value will be greater than the spec minimum and the low value will be lower than the spec maximum. In fact, with a light load the values will be close to V_{CC} and ground. Therefore, in order to obtain any accurate result for a voltage measurement the exact values of V1 and V0 need to be measured and somehow stored in the microcontroller. Typical values of these voltages can be measured experimentally and an average could be used for final implementation.

The other problem associated with the levels is that the capacitive load on the output line is substantial and far in excess of the values used when specifying the characteristics of the various COP420 outputs. The significant effect of this is that it will take longer than "normal" for the output to reach its maximum value. In addition, it is likely that there will be dips in the output as it rises to its maximum value since the capacitor will start to draw charging current from the output. All of this will be fast relative to the other system times. Still it will affect the result since the level to which the capacitor is attempting to charge is not being applied uniformly and "instantaneously". It can be viewed as though the voltage V1 is bouncing before it stabilizes.



TL/DD/6935–01

Crystal oscillator values chosen to give 4 µs cycle time with divide by 16 option selected on COP 420 CKO/CKI Pins

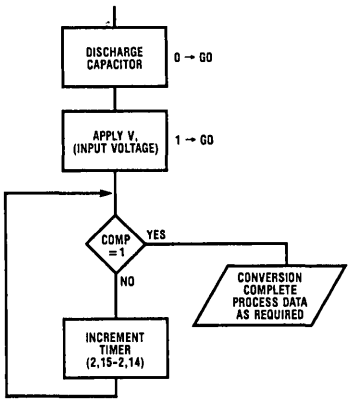
$V_{CC} = +5V$

FIGURE 1. Basic Capacitor Charge Technique

```
        OGI     0       ;TURN OFF G TO DISCHARGE CAPACITOR
        ; INSERT SOME DELAY TO MAKE SURE CAPACITOR DISCHARGED
        ;USING 12 BIT COUNTER, BUT ONLY UPPER 8 USED IN TABLE
        ;LOOK UP DUE TO ACCURACY OF RC CHARGE METHOD.  THE OTHER
        ;BITS COULD BE USED BUT THE COMPLICATIONS ARE NOT WORTH
        ;THE EFFORT FOR THIS PARTICULAR TECHNIQUE.  ALSO, HERE THE
        ;INPUT RANGE IS RESTRICTED SO THAT THE TOP 3 BITS ARE ZERO

RCAD:   OGI     1       ;TURN ON THE G LINE
INCR:   LBI     2,13    ;BINARY INCREMENT OF 12 BIT COUNTER
BINPLS: SC              ;LOWER FOUR BITS WILL BE DISCARDED
BINPL1: CLRA            ;ONLY TOP BITS USED IN TABLE LOOK UP
        ASC             ;SPEED WOULD BE IMPROVED IF THE ADD WERE
        NOP             ;STRAIGHT LINE CODED-BUT COSTS MORE CODE
        XIS
        JP      BINPL1
        ININ            ;READ IN3 TO SEE IF COMPARATOR CHANGED
        AISC    8
        JP      END
        CLRA
        JP      INCR
END:    OGI     0       ;TURN OFF THE G LINE AND DISCHARGE C
        ;DO ARITHMETIC HERE OR LOOK UP TABLE OR WHATEVER IS
        ;REQUIRED--SAMPLE LOOK UP TABLE CONTROL INDICATED BELOW
        ;SAMPLE TABLE WRITTEN CORRECTING FOR THE EXPONENTIAL
        ;RELATIONSHIP.  THE TABLE ALSO INCORPORATES A CONVERSION
        ;TO BCD.  THE VALUE IN THE TABLE IS THE RATIO OF
        ;THE CAPACITOR VOLTAGE V TO THE MAXIMUM VOLTAGE VMAX.
        ;THE NUMBER IS A TWO DIGIT BCD FRACTION.  WE ARE USING
        ;A 5 BIT COUNT IN THIS EXAMPLE.  ADDRESSING ARBITRARILY
        ;SET UP ASSUMING THAT CONTROL CODE IS IN PAGE 0 (OTHER
        ;THAN AT ADDRESS 0) AND THAT THE TABLE THEREFORE IS IN
        ;PAGE 1 (STARTING AT HEX ADDRESS 040).
        ;
        LBI     2,15    ;POINT TO TOP 4 BITS
        XDS             ;TOP 4 IN A,POINTING TO LOWER 4 IN 2,14
        AISC    4       ;THIS MERELY ADJUSTING FOR ADDRESS--NO
                        ;OTHER FUNCTION
        LQID            ;DO THE LOOK UP
        CQMA            ;FETCH THE ADJUSTED VALUE FROM Q
         ; THE ADJUSTED VALUE IS NOW IN A AND M.  FROM THIS POINT MAY
         ;USE THE VALUE IN OTHER CALCULATIONS OR OUTPUT THE INFORMATION,
         ;OR WHATEVER MAY BE REQUIRED BY THE APPLICATION.
        LBI     2,13    ;CLEAR THE COUNTER
        STII    0
        STII    0
        STII    0
        JP      RCAD:   ;JUMP BACK AND REPEAT

        .=X'040         ;SET UP TABLE ADDRESS
        .WORD 000,003,006,008  ;SET UP THE TABLE VALUES
        .WORD 011,014,016,019  ;HERE, COMPENSATED FOR EXPONENTIAL
        .WORD 021,023,026,028  ;AND CONVERTED TO BCD FRACTION
        .WORD 030,032,034,036  ;TABLE VALUE IS RATIO V/VMAX
        .WORD 038,039,041,043
        .WORD 045,046,048,049
        .WORD 051,052,053,055
        .WORD 056,057,059,060
```

TL/DD/6935-55

FIGURE 2A. Typical RC Charge A/D Code



TL/DD/6935-2

FIGURE 2B. Charge Flow Chart

A more general problem is that of the tolerance of RC time constant. The value of the voltage with respect to time is obviously related to the RC value. Therefore, a change in that value will result in a change in the voltage for a given time period t. The graph in *Figure 3* illustrates the effect of a ±10% variation in the RC value upon the voltage measured for a given time t. If one cares to work out the math, it comes out that the error is an exponential relationship in much the same manner as the capacitor voltage itself. The maximum error induced for ±10% RC variation is ±3.9%.

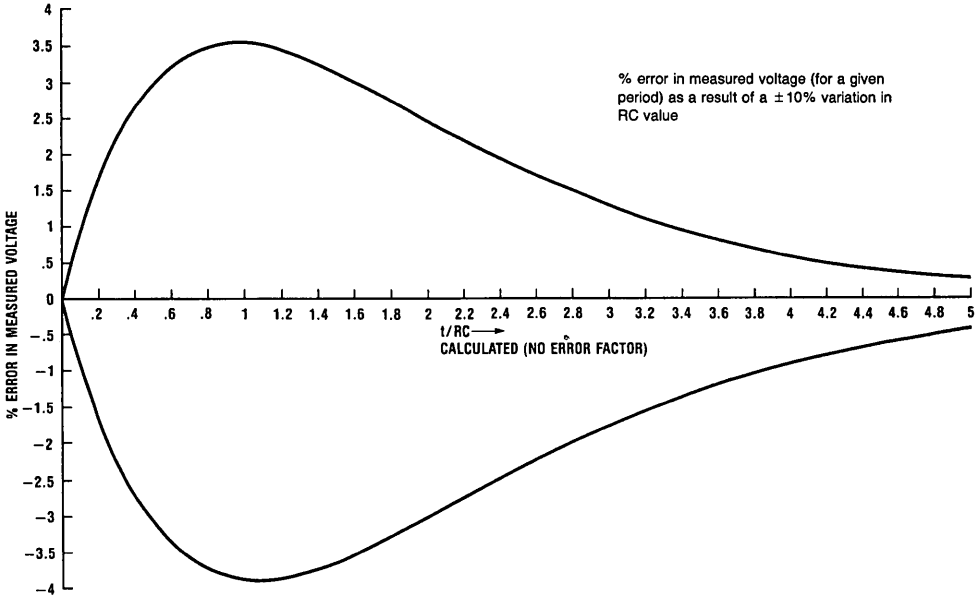
Remember also that we are measuring time. Therefore variation in the RC value will have a direct, linear effect on the time required to measure a given voltage. It is also necessary that the time base for the COP420 be accurate. A variation in the accuracy in the operating frequency of the COP420 will have a direct impact on the accuracy of the result.

Given the errors mentioned so far and assuming that no changes are made in the hardware, the accuracy of the technique then is determined by the resolution of the time measurement. This is improved in two ways: increase the RC time constant so that there is a smaller change in capacitor voltage for a given time period or try to minimize the loop time required to increment the counter. Lengthening the RC time constant is easier but the cost is increased conversion time. The minimum time to increment a 5 to 8 bit binary counter and test an input is 13 cycle times. For a 9 to 12 bit binary counter this minimum time is 17 cycle times. Note also that the minimum time to perform the function does not necessarily correspond to the minimum number of code words required to implement the function. At a cycle time of 4 μs, the 13 cycle times correspond to 52 μs.

2.2 ACCURACY IMPROVEMENTS

Several options are available if it is desired to improve the accuracy of this method. Three such improvements are shown in *Figure 4*. *Figure 4A* is the smallest change. Here a pullup resistor has been added to the G output line and the G line is run open drain internally, i.e., the internal pullup is removed. This improves the "bounce" problem mentioned earlier. The G line will go to the high state and remain there with this setup. However, the addition of the resistor does little more than eliminate the bounce. The degree of improvement is not great, but it is an easy way to eliminate a minor source of error.

Figure 4B is the next step. A 74C04 is used as a buffer. The 74C04 was chosen because of its symmetric output characteristics. Any CMOS gate with such characteristics could be used. The software can easily be adjusted to provide the proper polarity. The COP420 output drives a CMOS gate which in turn drives the RC network. This change does make significant improvements in accuracy. With a light



TL/DD/6935-3

FIGURE 3

load the CMOS gate will typically swing from ground to V_{CC} and its output level is not as likely to be affected by the capacitor discharge.

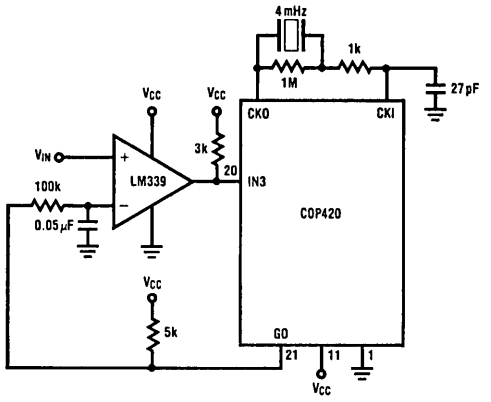
Figure 4C is the best approach, but it involves the greatest component cost. Here two G outputs are controlling analog switches. Ground is connected to the RC network to discharge the capacitor, and a positive reference is used to charge the capacitor. This reference can be any suitable voltage source: zener diodes, V_{CC}, etc. The controlling voltage tolerance is now clearly the tolerance of the reference. Precise voltage references are readily obtainable. *Figure 4C* also shows an analog switch connected directly across the capacitor to speed up the capacitor discharge time. When using this version of the basic scheme, remember to include the 'on' resistance of the analog switch connected to V_{REF} in the RC calculation. Failure to do so will introduce error into the result.

Note that the LM339 is a quad comparator. If these comparators are not otherwise needed in the system, they can be used in much the same manner as the CMOS gate mentioned above. They can be used to buffer the output of the COPS device and to reset the capacitor, or whatever other function is required. This has the advantage of fully utilizing the components in the system and eliminates the need to add another package to the system.

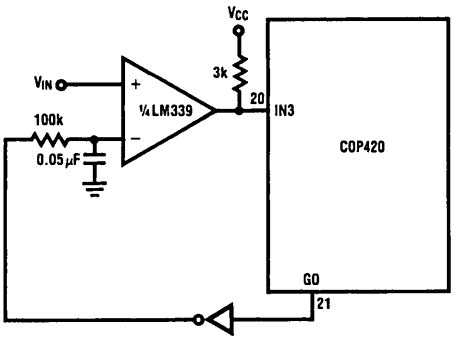
2.3 CONCLUSIONS

This approach is an inexpensive way to perform an A/D conversion. However, it is not that accurate. With a 10% V_{CC} supply and a 10% tolerance in the RC value and 10% variation in the oscillator frequency the best that can be hoped for is about 25% accuracy. If a 1% reference voltage is used, this accuracy becomes about 15%.

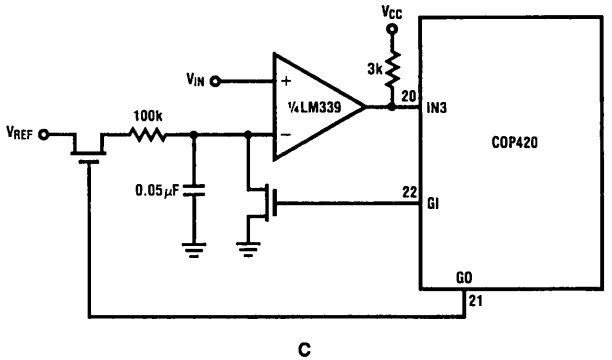
Under laboratory conditions—holding all variables constant and using precise measured values in the calculations—the configuration of *Figure 2* yielded 5 bit accuracy over an input range of 0 to 3.5V. Over the same range and under the same conditions, the circuit of *Figure 4B* yield 7 to 8 bit accuracy. It must be emphasized that these accuracies were obtained under controlled conditions. All variables were held constant and actual measured values were used in all calculations. It is unlikely that the general situation will yield these accuracies unless adjustments are provided and a calibration procedure is used. This could defeat the low cost objective.



A



B



C

FIGURE 4

3.0 Pulse Width Modulation (Duty Cycle) Technique

3.1 MATHEMATICAL ANALYSIS

The pulse width modulation, or duty cycle, conversion technique is based on the fact that if a repetitive pulse waveform is applied to an RC network, the capacitor will charge to the average voltage of the waveform provided that the RC time constant is sufficiently large relative to the pulse period. See *Figure 5*.

In this technique, the capacitor voltage V_C is compared to the voltage to be measured by means of an analog comparator. The duty cycle is then adjusted to cause V_C to approach the input voltage. The COPS device reads the comparator output and then drives one of its outputs high or low depending on the result, i.e., if V_C is lower than the input voltage, a positive voltage (V1) is applied to charge the capacitor; if V_C is higher than the input voltage, a lower voltage (V0) is applied to discharge the capacitor. Thus the capacitor voltage will seek a point where it varies above and below the input voltage by a small amount. *Figure 6* illustrates the capacitor voltage and the comparator output.

Some mathematical analysis here will be useful to help clarify the technique and to point out its restrictions. Referring to *Figure 6*, we have the following:

$V_A = V0 + [V_B - V0][e^{**}(-t1/RC)]$

$V_B = V_A + [V1 - V_A][1 - e^{**}(-t2/RC)]$

$= V1 + [V_A - V1][e^{**}(-t2/RC)]$

solving for t1 and t2 we have:

$t1 = -RC\ \ln[(V_A - V0)/(V_B - V0)]$

$t2 = -RC\ \ln[(V_B - V1)/(V_A - V1)]$

let:

$V_A = V_{IN} - d1$

$V_B = V_{IN} - d2$

substituting the above, the equations for t1 and t2 become:

$t1 = -RC\ \ln\{[1 - (d1/(V_{IN} - V0))]/[1 + d2/(V_{IN} - V0))]\}$

$t2 = -RC\ \ln\{[1 - (d2/(V_{IN} - V1))]/[1 \cdot -d1/(V_{IN} - V1))]\}$

the equations reduce by means of the following assumptions:

1. $d1 = d2 = d$
2. $|V_{IN} - V0| \gg d$

   $|V_{IN} - V1| \gg d$

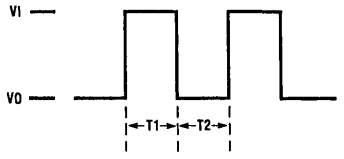
applying these assumptions, we get the following:

$t1 = -RC\ \ln[(1 + x)/(1 - x)]$ where $x = -d/(V_{IN} - V0)$

$t2 = -RC\ \ln[(1 + x)/(1 - y)]$ where $y = d/(V_{IN} - V1)$

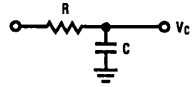
because of the assumptions above, the x and y terms in the preceding equations are less than 1, therefore the following expansion can be used:

$\ln[(1 + z)/(1 - z)] = 2[z + (z^{**}3)/3 + (z^{**}5)/5 + \ldots]$



TL/DD/6935-7

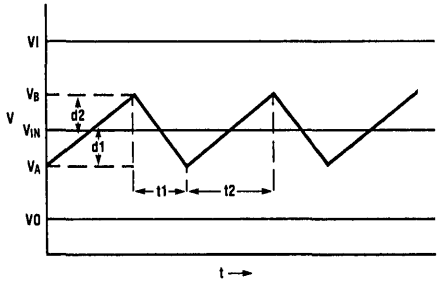
$$V_C = \frac{(V1 - V0) \times T1}{T1 + T2}$$



TL/DD/6935-8

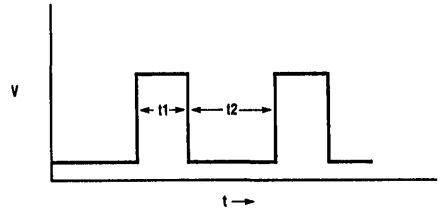
FIGURE 5

Capacitor Voltage



TL/DD/6935-9

Comparator Output



TL/DD/6935-10

FIGURE 6

substituting we have:

$t1 = -2RC[x + (x^{**}3)/3 + \ldots]$

$t2 = -2RC[y + (y^{**}3)/3 + \ldots]$

under assumption 2 above, the linear term completely swamps the exponential terms yielding the following result (after substituting back into the equation):

$t1 = 2dRC/V_{IN} - V0) \qquad t2 = -2dRC/(V_{IN} - V1)$

therefore:

$$t1/(t1 + t2) = (V1 - V_{IN})/(V1 - V0)$$

$$t2/(t1 + t2) = (V_{IN} - V0)/(V1 - V0)$$

solving for V_{IN}:

$V_{IN} = [t2/(t1 + t2)][V1 - V0] + V0$

or $V_{IN} = V1 - [t1/(t1 + t2)][V1 - V0]$

It follows from the above results that by measuring the times t1 and t2, the input voltage can be accurately determined. As will be seen the restrictions based upon the assumptions above do not cause any serious difficulty.

General Accuracy Considerations

In the preceding calculations it was assumed that the differential output above and below the input voltage was the same. If the comparator output is checked at absolutely regular intervals, and if the intervals are kept as small as possible this assumption can be fairly easily guaranteed—at least to within the comparator offset which is only a few millivolts. As we shall see, this aspect of the technique presents few, if any, difficulties. In addition, there is an RC network at the input of the comparator. The time constant of this network must be long relative to the time between checks of the comparator output. This will insure that the capacitor voltage does not change very much between checks and thereby help to insure that the differences above and below the input voltage are the same.

The next major approximation has to do with the difference between the input voltage and either V1 or V0. We have relied on this difference being much greater than the amount the capacitor voltage changes above and below the input voltage. This approximation allows the nonlinear terms in the logarithmic expansion to be discarded. In practicality, the approximation means that the input voltage must not be "close" to either V1 or V0. Therefore, it becomes necessary to determine how closely the input voltage can approach V1 or V0. It is obvious that the smaller the difference d can be made, the closer the input voltage can approach either reference. The following calculations illustrate the method for determining that difference d. Note, using either V1 or V0 produces the same result. Thus V = V1 = V0.

For at least 1% accuracy

$$x + (x^{**}3)/3 < 1.01x$$

$$\text{therefore } x < 0.173$$

since $x = d/|(V_{IN} - V)|$ we have $d < 0.173|(V_{IN} - V)|$.

Using the same analysis for 0.1% accuracy in the approximation we get $d < 0.0548|(V_{IN} - V)|$. By applying this relationship, the RC time constant can be adjusted so that, within the time interval, the capacitor voltage does not change by more than d V. The user may then select, within reason, how close to the references he can allow the input voltage to go.

The next consideration is really just one of simplification. It is clear that if V0 is zero, it drops out of the first equation and the relationship is simplified. Therefore, it is desirable to use zero volts as the V0 value. The equation then becomes:

$$V_{IN} = V1t2/(t1 + t2).$$

It is obvious by now that the heart of the technique lies in accurately measuring the times t1 and t2. Clearly this requires that the time base of the COP420 be accurate. Short term variations in the COP420 time base will clearly impact the accuracy of the result. In addition to that there is a serious problem in being able to check the comparator output often enough to get any accuracy and resolution out of simply measuring the times t1 and t2. This problem is circumvented by measuring many periods of the waveform. Doing this gives a large average, which improves the accuracy and tends to eliminate any spurious changes. Of course, the trade off is increased time to do the conversion. However if the time is available, the technique becomes restricted only by the accuracy of the external components. Those of the comparator and the reference voltage are most critical.

It is clear from the equation above that the accuracy of the result is directly dependent upon the accuracy of the reference voltage V1. In other words, it is not possible to be more accurate than the reference voltage. If, however, all that is required is a ratio between the input voltage and the reference voltage, the accuracy of the reference will not be a controlling factor provided that the input voltage tracks the reference. This requires that the input voltage be generated from the reference voltage in some form, e.g., a voltage divider with V_{IN} coming off a variable resistance.

Finally, we have noted that the difference d must be small. If the capacitor had to charge or discharge a long way toward V_{IN}, the nonlinearity of the capacitor charge curve would be significant. This therefore requires that the conversion begin with the capacitor voltage close to the input voltage.

Note that the RC value is not part of the equation. Therefore the accuracy of the time constant has no effect on the result as long as the time constant is long relative to the time between checks of the comparator output.

The final point is that the reference voltages, whatever they may be, must be hard sources. Should these voltages vary or drift at all, they will directly affect the result. In those configurations where the references are being switched in and out, the voltage should not change when it is switched into the circuit.

3.2 BASIC IMPLEMENTATION

General

The objective, then, is to measure the times t1 and t2. This is accomplished in the software by means of two counters. One of the two counters counts the t2 time; the other counter counts the total time t1 + t2.

It is necessary to check the comparator output at regular intervals. Thus the software must insure that path lengths

through the test and increment loops are equal in time. Further it is desirable to keep the time required to increment the counters as short as possible. A trade off usually comes into play here. The shortest loop in terms of code required to implement the function is rarely the shortest loop in terms of time required to execute the function. The user has to decide which implementation is best for him. The choice will frequently be governed by factors other than the A/D conversion limits.

It must be remembered that we are now dealing with analog signals. If significant accuracy is required, we are handling very small analog signals. This requires the user to take precautions that are normally required when working with linear circuits, e.g., power supply decoupling and bypassing, lead length restrictions, crosstalk, op amp and comparator stabilization and compensation, desired and undesired feedback, etc. As greater accuracy is sought these factors are more and more significant. It is suggested that the reader refer to the National Semiconductor Linear Applications Handbook and to the data sheets for the various components involved to see what specific precautions should be taken both in general and for a specific device.

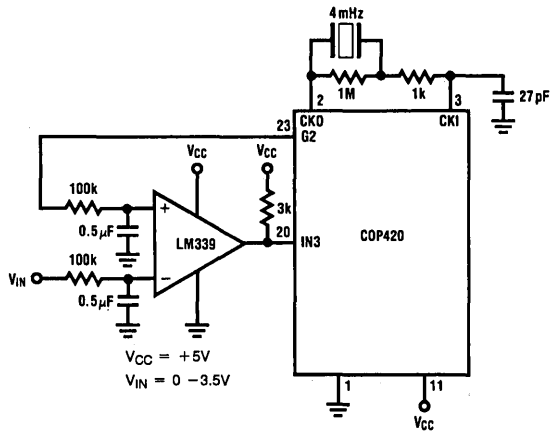
The Base Circuit

Figure 7 shows the diagram for the basic circuit required to implement the duty cycle conversion scheme. The flow chart and code required to implement the function are shown in *Figure 8*. Note that the flow chart and code do not change—except for possible polarity change on output to allow for an inverting buffer—for any of the improvements in accuracy discussed later. The only exception to this is the technique illustrated in *Figure 10* and the variations there are minor.

The code and flow chart in *Figure 8* implement the technique as described above. The large averaging technique is used as it would be too difficult to measure the times t1 and t2 in a single period. The total time, t1 + t2, is the viewing window under complete control of the software. This window is a time equal to the total number of counts, determined by desired accuracy, multiplied by the loop time for a single count. A second counter is counting the t2 time. Special care is taken to insure that all paths through the code take the same length of time since the integrity of the time count is the essence of the technique. The full conversion scheme would use the subroutine in *Figure 8*. Normally the subroutine would be called first just to get the capacitor charged close to the input voltage. The result obtained here would be discarded. Then the routine would be called a second time and the result used as required.

In the configuration in *Figure 7*, there is an RC network in both input legs of the comparator. This is to balance the inputs of the device. For this reason, R1 = R2. C1 is the capacitor whose voltage is being varied by the pulse waveform. C2 is in the circuit only for stabilization and symmetry and is not significant in the result. The comparator tends to oscillate when the + and − inputs are nearly equal without capacitor C2 in the circuit.

As would be expected, the basic circuit has some difficulties. By far the most serious of these difficulties is the output level of the G line. To be sure of the high and low level of this output the levels should be measured. The "1" level will be between the spec minimum of 2.4V and V_{CC} (here assumed to be 5V). The "0" level will be between the 0.4V spec maximum and ground. With light loads, these levels are likely to vary from device to device. Furthermore, we have the same "1" level problem that was mentioned in the simplest technique: the capacitive load is large and the capacitor is charging while the output is trying to go to the high level.



TL/DD/6935-11

FIGURE 7. Basic Duty Cycle A/D

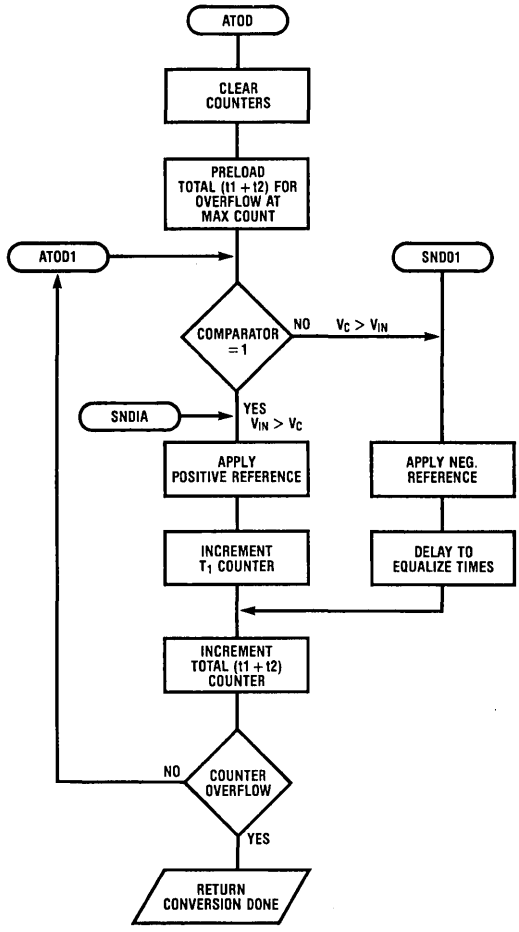
There is also a problem with the low level. When the output goes low, the capacitor begins to discharge through the output device of the COP420. This discharge current has the effect of raising the "0" level and thereby introducing error. Note that we are not talking about large changes in the voltages, especially the low level. Typically, the change will only be a few millivolts but that can translate into a loss of accuracy of several bits.

Under laboratory conditions—holding all variables constant and using precise measured values in the calculations—the circuit of *Figure 7* yielded 5 bit ± 1 bit accuracy over the range of V0 (here measured to be 0.028V) to 3.5V (the maximum specified input voltage for the comparator with V_S = 5V). Increasing the number of total counts had very little effect on the result. In the general case, the basic scheme should not be relied upon for more than 4 bits of accuracy, especially if one assumes that V1 = V_{CC} and V0 = 0. As shall be seen, it is not difficult to improve this accuracy considerably.

```
;ATOD IS THE FULL CONVERSION SCHEME WRITTEN AS A SUBROUTINE
ATOD:   LBI     1,10    ;MAKE SURE COUNTERS CLEARED
        JSRP    CLEAR
        LBI     2,10
        JSRP    CLEAR
        LBI     1,13    ;PRELOAD FOR TOTAL COUNT = 2048
        STII    0
        STII    0
        STII    8
ATOD1:  ININ            ;READ COMPARATOR--INPUT TO 420 = IN3
        AISC    8
        JP      SNDO1
SNDIA:  LBI     3,0     ;USING OMG BELOW TO SAVE STATE OF OTHER G
                ;VALUES IF IT WAS NECESSARY TO DO SO,ELSE USE OGI
        SMB     2       ;VIN > Vc,DRIVE Vc HIGHER
        OMG             ;THIS CODE STRAIGHT LINED FOR SPEED
        SC              ;APPLY POSITIVE REFERENCE
        CLRA            ;INCREMENT THE SUB COUNTER
        LBI     2,13
        ASC
        NOP
        XIS
        CLRA
        ASC
        NOP             ;BINARY INCREMENT
        XIS             ;WOULD ELIMINATE THESE 4 WORDS IF 8 BIT
        CLRA            ;COUNTER OR LESS-HERE SET UP FOR UP TO 12 BIT
        ASC             ;COUNTER
        NOP
        X
        JP      TOTAL
SNDO1:  LBI     3,0
        RMB     2
        OMG
        CLRA
        AISC    10      ;THIS PART OF THE CODE MERELY INSURES THAT
        NOP             ;ALL PATHS THROUGH THE ROUTINE ARE EQUAL IN TI
DLY:    AISC    1
        JP      DLY
TOTAL:  CLRA
        LBI     1,13
        SC
        ASC             ;INCREMENT THE TOTAL LOOP COUNTER
        NOP             ;WHEN OVERFLOW,DONE SO EXIT
        XIS
        CLRA
        ASC
        NOP
        XIS
        CLRA
        ASC
        JP      ATOD2
        RET
ATOD2:  X
        JP      ATOD1
        .PAGE   2
CLEAR:  CLRA
        XIS
        JP      CLEAR
        RET
```

TL/DD/6935-45

FIGURE 8A. Duty Cycle A/D Code



TL/DD/6935–12

FIGURE 8B. Duty Cycle A/D Flow Chart

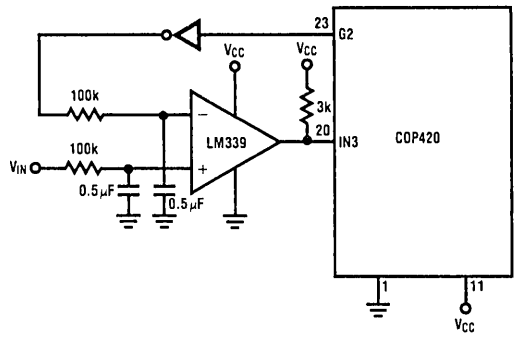
3.3 ACCURACY IMPROVEMENTS

General Improvements

Figure 9 illustrates circuit changes that will make significant improvements in the accuracy of the technique. In *Figure 9A* a CMOS buffer is used to drive the RC network. The output of the COP420 drives the CMOS gate, which here is a 74C04 because of its output characteristics. The main thing that this technique does is to reduce the difficulties with the output levels. Typically, V0 is 0V and V1 is V_{CC}. We also have a "harder" source for the voltages — the levels don't change while the capacitor is charging or discharging. Now, even more clearly than before, the accuracy of V_{CC} is the controlling voltage tolerance. The accuracy of the result will be no better than the accuracy of V_{CC} (for a system requiring absolute accuracy).

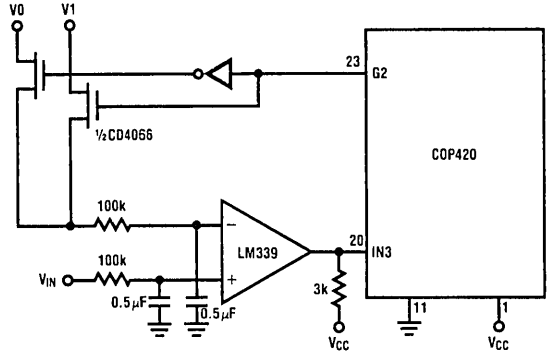
Under laboratory conditions, the circuit of *Figure 9A* yielded the accuracies as indicated below for various total counts. The accuracy increased with the total count until the count exceeded 2048. There was no significant increase in accuracy with this circuit for counts in excess of 2048. (Remember that these results were obtained under controlled conditions). We may then view the results obtained with 2048 counts as the upper limit of accuracy with the circuits of *Figure 9A*. The results were as follows:

Total Count	Resultant Accuracy
512	8 ± 1/2 bits
1024	9 ± 1bits
2048	9 ± 1/2 bits
4096	9 ± 1/2 bits



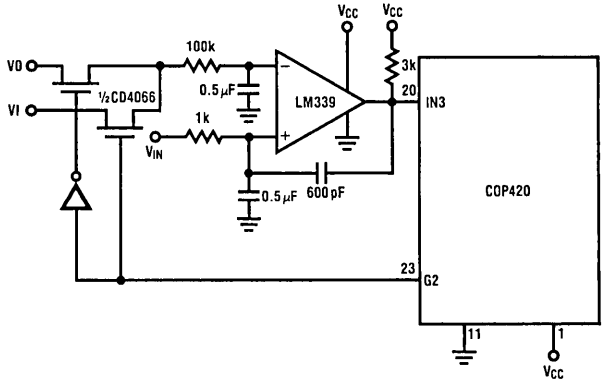
TL/DD/6935–13

A



TL/DD/6935–14

B



TL/DD/6935–15

C

FIGURE 9. Improvements to Duty Cycle A/D

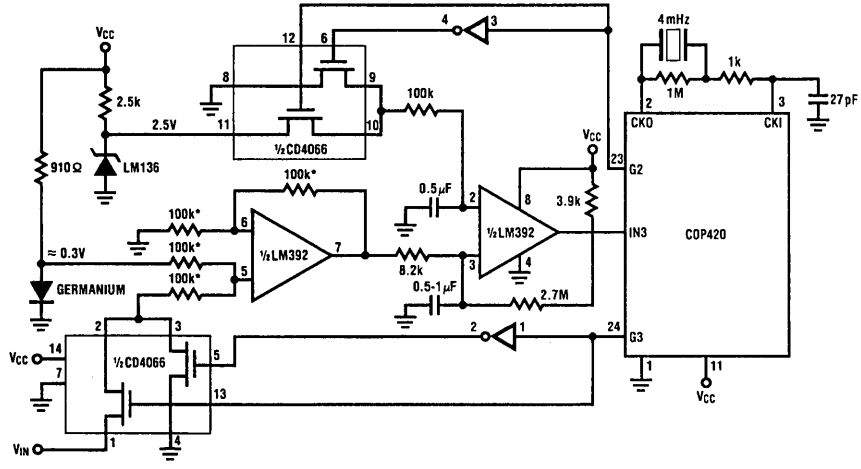
The circuit of *Figure 9B* makes a significant change to improve accuracy. Now the COP420 is controlling analog switches and switching in positive and negative references. Therefore the accuracy of the reference voltages is the controlling factor. Generally this will improve the accuracy over that obtained with *Figure 9A*. With the circuit of *Figure 9B*, with V0 = 1V (negative reference), and V1 = 3V (positive reference), 9 bit accuracy was achieved with a total count of 1024. V0 and V1 were arbitrarily chosen to place the input voltage approximately in the center of the allowable comparator input range with V_S = 5V. Remember, the accuracy of the references is controlling. The result can be no more accurate than the references. Furthermore, these references must be hard sources; i.e., they must not change when they are switched into the circuit as that contributes error into the result.

In *Figure 9C*, capacitive feedback was added to the comparator circuit and the series resistance to V_{IN} was decreased. The feedback added hysteresis and forced the comparator to slew at its maximum rate (significant errors are introduced if the comparator does not change state in a time shorter than the cycle time of the controller). Both of these changes resulted in increased accuracy of the result. With V0 = 0, V1 = 5V (V_{CC}) and V_{CC} held steady at 5.000V, an accuracy of 10 bits ± 1 bit was achieved over the input range of 0 to 3.5V.

It is obviously possible to use any combination of the configurations in *Figure 9* for a given application. What is used will depend on the user and his specific requirements.

Figure 10 illustrates a further refinement of the basic approach. This configuration can be used if greater accuracies are needed. The major change is the addition of a summing amplifier to the circuit for the purpose of adding a fixed offset voltage to the input voltage. This has the effect of moving the input voltage away from the negative reference (which is 0V here). This offset voltage should be stable as the changes in it will directly affect the result. The offset voltage should be chosen so as to place the effective input voltage (the voltage at the comparator input) approximately in the center of the range between the two references. The precise value of the offset is not critical nor is its source. The forward voltage drop across a germanium diode is used as the offset in *Figure 10*, but this offset can be generated in any convenient manner. The forward voltage drop of the germanium diode is approximately 0.3V. Given this and the negative reference of 0V and a positive reference of 2.5V, the input voltage is restricted to a range of 0 to 2V. Therefore, the effective input voltage (at the comparator input) is approximately 0.3V to 2.3V — well within the limits of the two references. The circuit also includes provision for an autozero self calibration procedure.

Note that the resistors in the summing amplifier should be matched. The absolute accuracy of these resistors is not significant, but their accuracy relative to one another can have a significant bearing on the result. The restriction is imposed so that the output of the summing amplifier is exactly the sum of the input voltage and the offset voltage. This requires unity gain through the amplifier and that the



TL/DD/6935–16

*Resistors should be matched

V_{CC} = +5V

$0 \le V_{IN} \le 2V$

FIGURE 10. Improved Duty Cycle A/D with Autozero

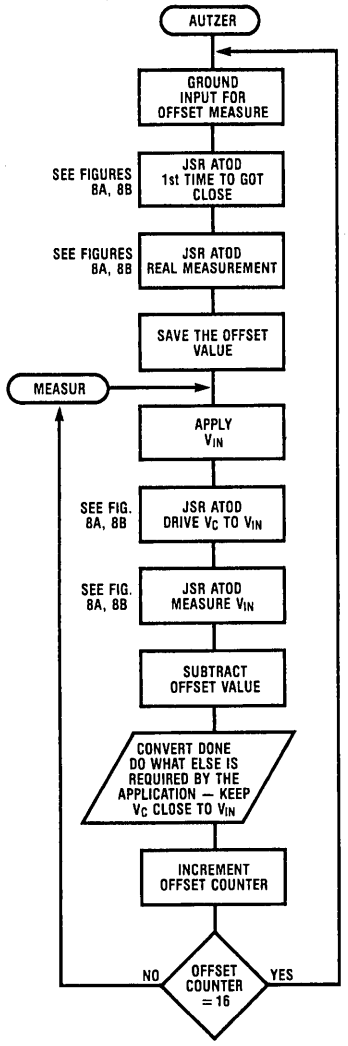
impedance in each summing leg be the same. These effects can become very serious if one is trying for significant accuracy—e.g., if 12 bit accuracy is being sought 1% matching of those resistors can introduce an error of 1% maximum. While 1% accurate is fairly good, it is significantly less than 12 bit accuracy. Related to this effect is a possible problem with the source impedance of the input voltage. If that impedance is significant in terms of its ratio to the summing resistor, errors are introduced just as if the resistors are mismatched. "Significant" is determined in terms of the desired system accuracy and the relative impedance values. The comparator section is using some feedback to provide hysteresis for stability and a low series resistance is used for the input to the comparator.

Most significantly, this configuration allows a true zeroing of the system. Through the additional analog switches shown, the COP420 can easily perform an autozero function by tying the input to ground and measuring the result. Thus the system offsets can be calculated, stored and subtracted from the result. This improves the accuracy and is also more forgiving on the choice of the comparator and op amp selected. Furthermore, the offset can be periodically recomputed by the COP420 thereby compensating for drift in system offsets. Nonetheless, the accuracy of the reference is the controlling factor. It is NOT possible to obtain an absolute (as opposed to ratiometric) accuracy of 12 bits without a reference that is accurate to 12 bits. The LM136 used in *Figure 10* is a 1% reference. Although not inherently accurate to 12 bits, the voltage of the LM136 may be trimmed to exact value by means of a variable resistor. The data sheet of the LM136 illustrates this connection. Under laboratory conditions, the circuit of *Figure 1* yielded 11 bit ±1 bit accuracy with a total count of 4096 over the input range of 0 to 2V. *Figure 11* indicates the flow chart and the code required to implement the technique of *Figure 10*.

```
        ;CODE FOR IMPROVED A TO D PULSE WIDTH METHOD
        ;SEE FIGURE 8A FOR CODE FOR ROUTINE ATOD
        ;
AUTZER: LBI     3,0     ;DO AUTO ZERO, 3,0 CONTAINS G STATUS
        RMB     3       ;SET UP TO GRND INPUT & MEASURE OFFSET
        JSR     ATOD    ;FIRST TIME IS TO GET CLOSE
        JSR     ATOD    ;MEASURE THE OFFSET
        LBI     2,13    ;NOW SAVE THE OFFSET VOLTAGE
XFER:   LD      1               ;SAVE THE OFFSET VALUE IN M3
        XIS     1
        JP      XFER
        LBI     0,0
        JP      INPUT
MEASUR:             ;NOW DO REAL MEASUR(1ST TIME IS OFFSET)
        JSR     ATOD    ;FIRST TIME TO GET CLOSE
        JSR     ATOD    ;NOW REAL MEASUREMENT
        JSRP    BINSUB  ;SUBTRACT THE OFFSET
        ;HAVE THE VALUE AT THIS POINT(IN BINARY)-NOW DO WHAT
        ;THE APPLICATION REQUIRES.  VALUE MUST BE MULTIPLIED
        ;BY (VREF+/TOTAL COUNT) TO GET FINAL VALUE IF SUCH IS
        ;DESIRED
        LBI     1,0     ;INCREMENT COUNTER FOR NEW OFFSET MEASURE
        LD
        AISC    1
        JP      SAVE
        X               ;IS 16TH TIME, MEASURE OFFSET AGAIN
        JP      AUTZER
SAVE:   X
        LBI     3,0
        SMB     3       ;SET BIT SO CAN MEASURE VIN
        JP      MEASUR
        .PAGE   2
BINSUB: LBI     3,13
        SC
BNSUB2: LD      1
        CASC
        NOP
        XIS     1
        JP      BNSUB2
        RET
```

TL/DD/6935–46

FIGURE 11A. Duty Cycle A to D, Improved Method



TL/DD/6935–17

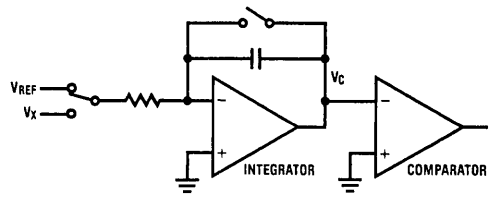
FIGURE 11B. Flow Chart for Improved Duty Cycle A/D

4.0 Dual Slope Integration Techniques

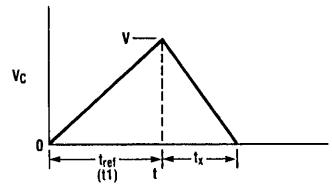
4.1 Mathematical Background

(Some of this background information is taken from National Semiconductor Linear Applications Note AN-155. The reader is referred to that document for other related general information.)

The basic approach of dual slope integration conversion techniques is to integrate a voltage across a capacitor for a fixed time, and then to integrate in the other direction with a known voltage until the starting point is reached. The ratio of the two times then represents the unknown voltage. Some of the math below in conjunction with *Figure 12* will illustrate the approach.



TL/DD/6935–18



TL/DD/6935–19

FIGURE 12. Dual Slope Integration—Basic Concept

$$I_X = C\frac{dV}{dt} = V_X/R$$

$$V_X = RC\frac{dv}{dt}$$

$$\int_{\circ}^{T1} V_X dt = \int_{\circ}^{V} RCdV$$

$$V_X\,T1 = RCV$$

$$V = V_X\,T1/RC = I_X T1/C$$

Similarly:

$$I_{REF} = C\frac{dV}{dt} = V_{REF}/R$$

$$V_{REF} = RC\frac{dV}{dt}$$

$$\int_{T1}^{T1\,+\,T_X} V_{REF} dt = \int_{V}^{\circ} RCdV$$

$$V_{REF}T_X = -RCV$$

$$V = -V_{REF}T_X/RC$$

$$-V_{REF}T_X/RC = V_X T1/RC$$

$$V_X = -V_{REF}T_X/T1$$

Two important facts arise from the preceding mathematics. First of all, there is a linear relationship involved in determining the unknown voltage. Secondly, the negative sign in the final equation indicates that the reference and the unknown, relative to some point (which may be 0V or some bias voltage), have opposite polarity. Thus, if it is desired to measure 0 to +5V, the reference voltage must be −5V. If the input is restricted to 2.5 to 5V, the reference can be 0V as the integrator and comparator are biased at +2.5V (then the 0V is in fact −2.5V relative to the biasing voltage, and the input range is 0 to 2.5V relative to the same bias voltage).

There are some difficulties with dual polarity conversion using the dual slope method. It is clear from the math above that if the input voltage will be dual polarity, it is necessary to have two references—one of each polarity. The midrange biasing arrangement briefly described above eliminates

the need for two different polarities but does not help very much since two references are still required—one at the positive value and one at the bias value. Ground is the other reference. Further, the need to select one of two references further complicates the circuitry involved to implement the approach. Also, the dual requirement brings up a difficulty with the bias currents of the integrator and comparator. They could add to the slope in one polarity and subtract in the other.

The only real operational difficulty in dual slope systems is establishing the initial conditions on the integrating capacitor. If this capacitor is not at the proper initial conditions, accuracy will be severely impaired. *Figure 12* indicates a switch across the capacitor as a means of initializing it. In a software driven system, the initialization can be accomplished by doing two successive conversions. The result of the first conversion is discarded. It is performed only to initialize the capacitor. The second conversion produces the valid result. One need only insure that there is not significant time lapse between the two conversions. They should take place immediately after one another.

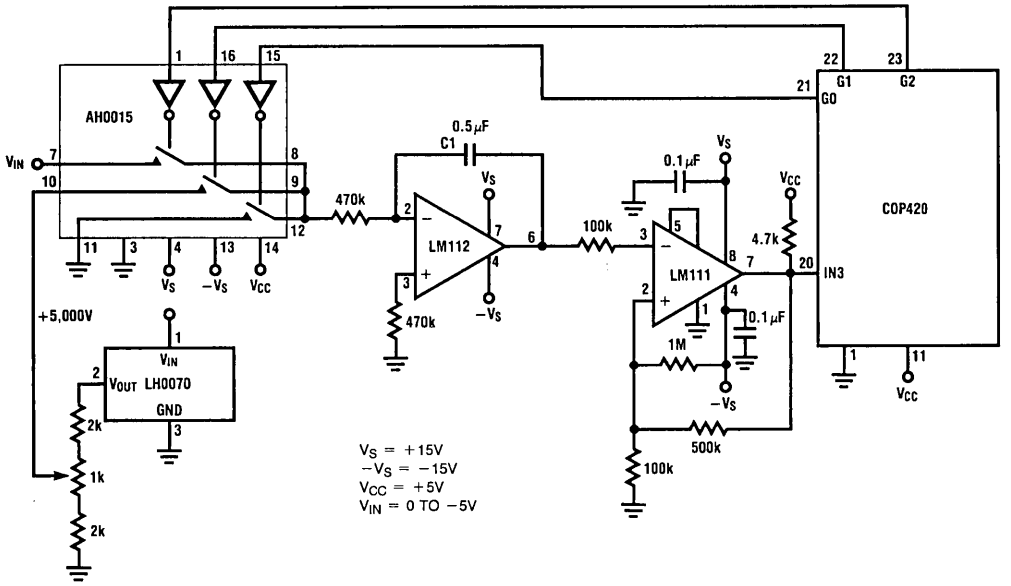
This approach obviously lengthens conversion time but it eliminates many problems. The alternative to this approach of two successive conversions is to take a great deal of care in insuring the initial state of the integrating capacitor and in selecting op amps and comparators with low offsets.

4.2 THE BASIC DUAL SLOPE TECHNIQUE

Figure 13 indicates an implementation of the basic dual slope technique. This is a single polarity system and thus requires only the single reference voltage. The circuit of *Figure 13* is perhaps not the cheapest way to implement such a scheme but it is representative and illustrates the factors that must be considered.

Consider first the means of initializing the integrating capacitor C1. The routine here connects the input to ground and does a conversion on zero volts as a means of initialization. Subsequently—and this is typical of the more usual technique—two conversions are performed. The first conversion is to initialize the capacitor. The second conversion yields the result. Some form of initialization or calibration procedure is required to achieve optimum accuracy from dual slope conversion schemes.

The comparator in this circuit is used in the inverting mode and has positive feedback as recommended in the LM111 data sheet. The voltage reference is the LH0070, which is a 0.01% reference. A resistive voltage divider on the IH0070 creates the 5V value. The use of the voltage divider brings up two difficulties (which can be overcome if the LH0070 is used at its full value, thus eliminating the divider, and the result properly scaled in the microcontroller or series integrating resistor increased). First, the impedance of the reference must be small relative to the series resistance used in the integrator. If this were not the case, the slopes would



TL/DD/6935–20

FIGURE 13. Basic Dual Slope Integration A/D Scheme

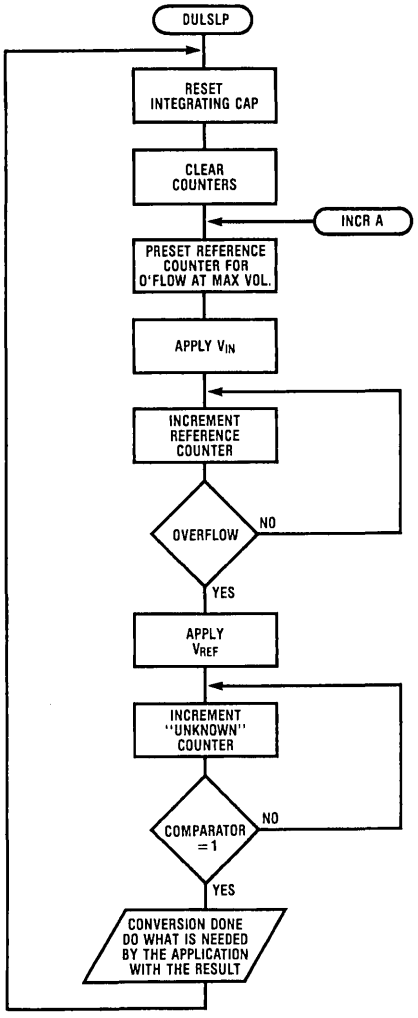
show an effect due to the difference in the R value between the applied reference voltage and the unknown input. (By the same token, the output impedance of the source supplying the unknown must also be small relative to that series integrating resistor). Secondly, the bias currents of the integrator may be such as to affect the reference voltage when it is coming from a simple resistor divider. Both problems are reduced if small resistor values are used in the divider. Note also that current mode switching would reduce the problem as well. It should be pointed out that the errors introduced by these problems are not gross deviations from the expected value. They are small errors that will not make much difference in the majority of applications. They are, however the kind of errors that can make the difference between a system accurate to 10 bits and one accurate to 12 bits (assuming all other factors are the same).

Figure 14 shows the flow chart and code required to implement the basic dual slope technique as shown in *Figure 13*. Under laboratory conditions an accuracy of 12 bits ±1 bit was achieved. The method is slow, with the maximum conversion time equal to 2 x T_{REF}. Notice that the accuracy of V_{CC} and that of the integrating resistor and capacitor are not involved in the accuracy of the result. The accuracy of V_{REF} is, of course, controlling if absolute accuracy—rather than ratiometric accuracy—is desired. The absolute accuracy of the circuit can be no better than the accuracy of the reference. If ratiometric accuracy is all that is required, there is no particular problem. The accuracy is merely relative to the reference. The R and C values do not impact the accuracy because the integration in both directions is being done through the same R and C. Results would be quite different if a different value of R or C was used for one of the slopes.

```
DUALSLP: OGI     1          ;HOLD THE INPUT TO GROUND TO RESET THE
         LBI     2,11       ;INTEGRATING CAPACITOR
         JSRP    CLEAR      ;CLEAR THE COUNTER
         JSR     INCRA      ;TO GET US CLOSE, NEXT READING IS REAL
CLEAR2:  LBI     2,11       ;NOW CLEAR THE COUNTER
         JSRP    CLEAR      ;MAKE SURE COUNTER CLEARED TO ZERO
  ;J,15 = 0 AND START AT 1,13 FOR COUNT = 4096
  ;J,15 = 14 AND START AT 1,12 FOR COUNT = 8192
  ;J,15 = 12 AND START AT 1,12 FOR COUNT = 16384
  ;FOLLOW SAME PATTERN FOR OTHER COUNTS
  ;
MEASUR:  JSR     INCRA      ;RUN THRU THE INCREMENTS
         ; NOW HAVE THE BINARY VALUE, USE IT AS IS OR
         ; MULTIPLY BY (Vref/TOTAL COUNT) TO CREATE THE VOLTAGE
         ; RESULT--THEN CONTINUE WITH THE OPERATION
         LBI     2,11
         JSRP    CLEAR      ;CLEAR THE COUNTER
         JSR     INCRA      ;TO GET CAP CLOSE TO 0 AGAIN
         JP      CLEAR2
 ;FOLLOWING SUBROUTINE INCRA IS THE REAL PART OF THE ROUTINE
 ;CONCERNED WITH THE COUNTING FOR THE CONVERSION.
INCRA:   LBI     1,15       ;R1 IS CLEARED PRIOR TO START
         STII    15         ;PRESET THE COUNTER FOR 4096
         OGI     4          ;APPLY VIN
INCR:    LBI     1,12
         SC
BINAD1:  CLRA
         ASC
         NOP
         XIS
         JP      BINAD1
         NOP                ;2 NOPS TO EQUALIZE TIMES
         NOP
         SKC
         JP      INCR
         OGI     2          ;DONE, NOW APPLY VREF
INCR2:   LBI     2,12       ;COUNT UNTIL COMPARATOR CHANGES
         SC
BINAD2:  CLRA
         ASC
         NOP
         XIS
         JP      BINAD2     ;STRAIGHT LINE THE ADD FOR SPEED
         ININ               ;SAVE WORDS BY USING G
         AISC    8          ;SEE IF IN3=1
         JP      INCR2      ;IN3 IS 0, KEEP COUNTING
OUTPUT:  OGI     1          ;KEEP INPUT AT 0
         RET
```

TL/DD/6935-47

FIGURE 14A. Dual Slope A/D Code



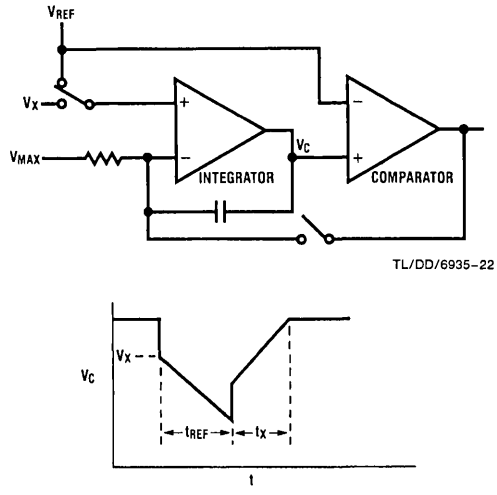
TL/DD/6935-21

FIGURE 14B. Basic Dual Slope A/D Flow Chart

4.3 MODIFIED DUAL SLOPE TECHNIQUE

General

The basic idea of the modified dual slope technique is the same as that of the basic approach. The modified approach eliminates the need for dual polarity references and is also more forgiving in the selection of the op amp and comparator required. *Figure 15* illustrates the basic idea.



TL/DD/6935-22

TL/DD/6935-23

FIGURE 15. Modified Dual Slope — Basic Concept

The math analysis is much the same:

$$I_X = C\frac{dV}{dt} = (V_X - V_{MAX})/R$$

$$V_X - V_{MAX} = RC\frac{dV}{dt}$$

$$(V_X - V_{MAX})T1 = RC$$

$$V = (V_X - V_{MAX})T1/RC$$

Similarly:

$$I_{REF} = C\frac{dV}{dt} = (V_{REF} - V_{MAX})/R$$

$$(V_{REF} - V_{MAX})T_X = -VRC$$

$$V = -(V_{REF} - V_{MAX})T_X/RC$$

$$(V_{MAX} - V_{REF})T_X = (V_X - V_{MAX})T1$$

$$V_X = V_{MAX} + (V_{MAX} - V_{REF})T_X/T1$$

The main difference between this and the basic approach is the offset voltage V_{MAX}. The main restriction is that all input voltage values (V_X) are less than V_{MAX}. It is also apparent that the total count is proportional to the difference between V_{MAX} and V_X. The only significant effect of this is, however, to slightly complicate the arithmetic required to arrive at a value for V_X.

Given that the input voltage V_X is always less than V_{MAX}, the modified dual slope technique is automatic polarity. This fact comes straight out of the equation above. Thus dual polarity references are not required. However, two precise voltages are required: V_{MAX} and V_{REF}. However, the V_{MAX} value can be used for a zero adjust as indicated in *Figure 16*. This means that the V_{MAX} value need not be so precise as it will be adjusted in a calibration procedure to produce a zero output. This adjustment amounts to a compensation for the bias currents and offsets. Thus the COP420 can use the supposed value of V_{MAX} with V_{MAX} later being "tweaked" to give the proper result at zero input. In addition, the initialization loop for the integrating capacitor includes the comparator. Thus the intial condition on the capacitor becomes

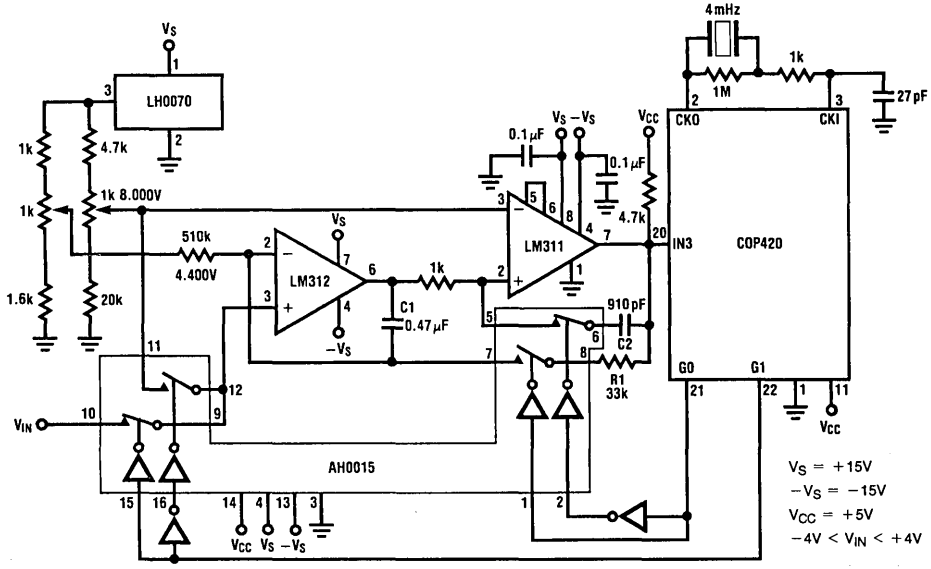
not zero but the sum of the offset voltages of the comparator and op amp. Thus the choice of these components is not critical in a modified dual slope approach.

An Example of the Modified Dual Slope Approach

Figure 16 illustrates an implementation of the modified dual slope technique. The system is calibrated by holding V_{IN} to ground and then adjusting V_{MAX} for a "0" result. Capacitor C1 is the integrating capacitor. Capacitor C2 is used only to cause a rapid transition on the comparator output. C2 is especially useful if an op amp is being used as the comparator stage. Resistor R1 is just part of the capacitor initializing loop. An LH0070 is being used to generate the reference voltage and the V_{MAX} value. The discussion previously about these being hard sources is equally relevant here. In fact, this problem was much more significant in this particular implementation and made the difference between a 10 and 12 bit system. As shown, the technique was accurate to 10 bits. Another bit was obtained when the V_{MAX} and V_{REF} values were buffered. It must be remembered that when trying to achieve accuracies of this magnitude board layout, parts placement, lead length, etc. become significant factors that must be specifically addressed by the user.

There are some other considerations in using this technique. The amount of time required to count the specified number of counts starts to become a significant factor. If it takes "too long" to do the counting, the capacitor can charge to either supply voltage depending on which direction it is integrating. This causes the wave shape shown in *Figure 15* to flatten out. This effectively limits the input range for all accuracy is lost once that waveform flattens out. In fact, this was the limiting factor on the accuracy in *Figure 16* as shown. Given the amount of time required for an increment of the counter for T_{REF} (or T_X), it was not possible to reach the 4096 counts required for 12 bit accuracy before the waveform flattened out. Decreasing the total count solves the problem at the expense of accuracy. It is therefore desirable to keep the loop time required for an increment as fast as possible. The code to implement *Figure 16* is shown in *Figure 17* and reflects that concern. The other way to solve the problem is to use a large value for R and C. This is the easiest solution and preserves accuracy. Its cost is increased conversion time.

Both the basic and modified dual slope schemes can be very accurate and are commonly used. They tend to be relatively slow. In many applications, however, speed is not a factor and these approaches can serve very well. There are various approaches to dual slope analog to digital conversion which try to improve speed and/or accuracy. These are usually multiple ramping schemes of one form or another. The heart of the approach is the basic scheme described above. It is not the purpose here to delve into all the possible ways that dual slope conversion may be accomplished. The control software is not significantly different regardless of which particular variation is used. The basic ramping control is the same as that indicated here.



TL/DD/6935–24

FIGURE 16. Modified Dual Slope Integration

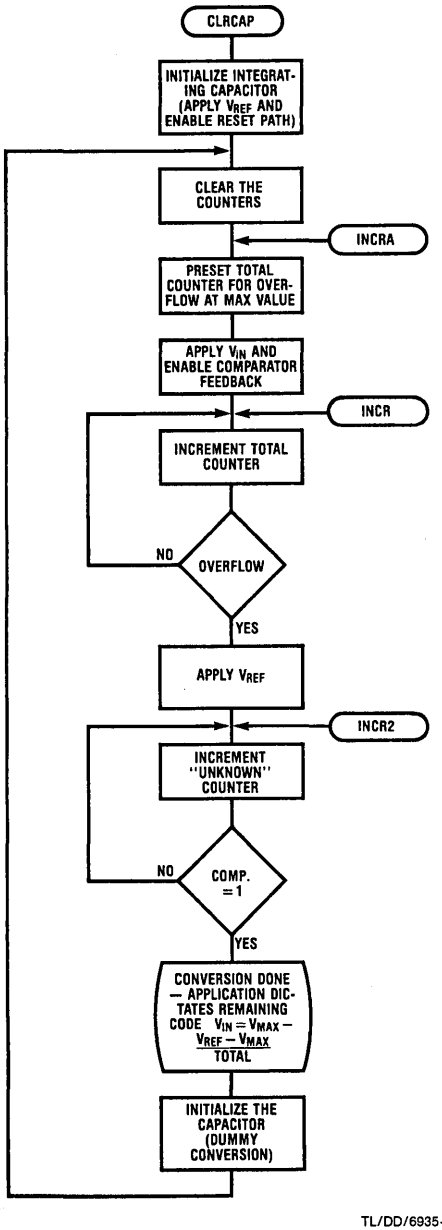
The number of components required to implement a dual slope scheme is not related to the desired accuracy. The approach is generally tolerant as to the op amps and comparators used as long as proper care is given to the initialization of the integrating capacitor.

Precise references are not required if a ratiometric system is all that is required. Cheaper switches can be safely used. The dual slope scheme controlled by a COPS microcontroller can be a very cost effective solution to an analog to digital conversion problem.

```
CLRCAP: OGI     1       ;APPLY VREF AND ENABLE RESET PATH
CLEAR2: LBI     2,11    ;NOW CLEAR THE COUNTER
        JSRP    CLEAR
  ;J,I5=15,1,14=4 AND START AT 1,12 FOR COUNT = 3072
  ;J,I5 =15 AND START AT 1,12 FOR COUNT = 4096
  ;J,I5 = 14 AND START AT 1,12 FOR COUNT = 8192
  ;J,I5 = 12 AND START AT 1,12 FOR COUNT = 16384
  ;FOLLOW SAME PATTERN FOR OTHER COUNTS
  ;
MEASUR: JSR     INCRA   ;RUN THRU THE INCREMENTS
        ;HAVE THE VALUE AT THIS POINT, DO WHAT THE APPLICATION
        ;REQUIRES--REMEMBER, TO CREATE REAL VALUE MUST MULTIPLY
        ;RESULT BY (VREF-VMAX)/TOTAL COUNT AND THEN SUBTRACT
        ;THAT RESULT FROM VMAX--DO IT IN DECIMAL OR BINARY, WHICHEVER
        ;IS BEST FOR THE APPLICATION
        LBI     1,11    ;MAKE SURE SPACE IS CLEARED
        JSRP    CLEAR
        LBI     2,11
        JSRP    CLEAR
        JSR     INCRB    ;FOR TEST-KEEP IT CLOSE
        LBI     1,11     ;MAKE SURE COUNTER IS CLEARED
        JSRP    CLEAR
        JP      CLEAR2
INCRA:  LBI     1,14
        STII    4        ;PRESET HERE FOR SMALLER COUNT
        STII    15       ;PRESET THE COUNTER FOR 4096
INCRA1: OGI     2        ;APPLY VIN AND ENABLE FEEDBACK
INCR:   LBI     1,12
        SC
BINAD1: CLRA
        ASC
        NOP
        XIS
        JP      BINAD1
        NOP              ;2 NOPS TO EQUALIZE TIMES
        NOP
        SKC
        JP      INCR
        OGI     0        ;DONE, NOW APPLY VREF
INCR2:  LBI     2,12     ;COUNT UNTIL COMPARATOR CHANGES
        SC
BINAD2: CLRA
        ASC
        NOP
        XIS
        JP      BINAD2   ;STRAIGHT LINE THE ADD FOR SPEED
        ININ             ;SAVE WORDS BY USING G
        AISC    8        ;SEE IF IN3=1
        JP      INCR2    ;IN1 IS 0, KEEP COUNTING
OUTPUT: OGI     1        ;CLEAR THE CAPACITOR, APPLY VREF
        RET
INCRB:  LBI     1,14     ;MAKE THE PASS FOR CAP INIT SHORT
        STII    7
        STII    15
        JP      INCRA1
```

TL/DD/6935-48

FIGURE 17A. Modified Dual Slope Code



TL/DD/6935-25

FIGURE 17B. Modified Dual Slope Flow Chart

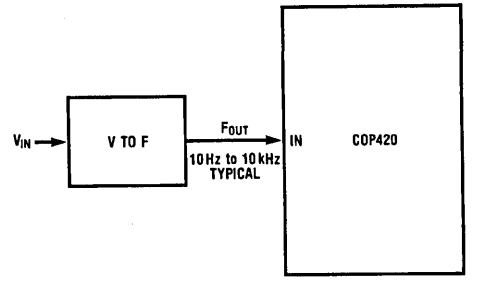
5.0 Voltage to Frequency Converters, VCO's

5.1 BASIC APPROACH

The basic idea of this scheme is simply to use the COP420 to measure the frequency output of a voltage to frequency converter or VCO. This frequency is in direct relation to the input voltage by the very nature of such devices. There are really only two limiting factors involved. First of all, the maximum frequency that can be measured is defined in the microcontroller by the amount of time required to test an input and increment a counter of the proper length. With the COP420 this upper limit is typically 10 to 15 kHz. The other limiting factor is simply the accuracy of the voltage to frequency converter or VCO. This accuracy will obviously affect the accuracy of the result.

Two basic implementations are possible and their code implementation is not significantly different. First, the number of pulses that occur within a given time period may be counted. This is straightforward and fairly simple to implement. The crucial factor is how long that given time period should be. To get the maximum accuracy from this implementation the time period should be one second. Such a time period would allow the distinction between the frequencies of 5000 Hz and 5001 Hz for example (assuming the V to F converter was that accurate or precise). Decreasing the amount of time will decrease the precision of the result. The alternate approach is to measure (by means of a counter) the amount of time between two successive pulses. This period measurement is only slightly more complicated than the pulse counting approach. The approach also makes it possible to do averaging of the measurement during conversion. This will smooth out any changes and add stability to the result. The time measurement technique is also faster than the pulse counting approach. Its accuracy is governed by how finely the time periods can be measured. The greater the count that can be achieved at the fastest input frequency — shortest period — the more accurate the result.

Figure 18 illustrates the basic concept. *Figure 19A* shows the flow charts and code implementation for both of the approaches discussed above. Note that whatever type of V to F converter is used, the code illustrated in *Figure 19A* is not significantly changed. In the code of *Figure 19A*, the interrupt is being used to test an input and thereby decreases the total time loop.



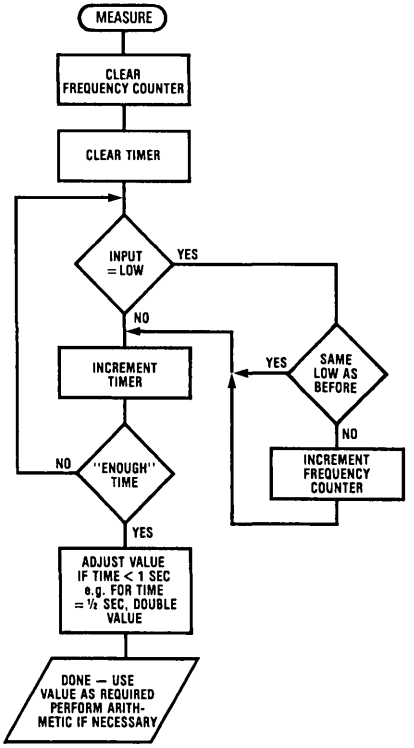
TL/DD/6935-26

FIGURE 18. V to F Converter — Basic Concept

```
MEASUR:          ;MEASURE BY COUNTING PULSES OF V TO F
;
         LEI     2        ;ENABLE INTERRUPT
         LBI     1,14     ;PRESET TIME FOR 122 COUNTS
         STII    5        ;APPROX ONE HALF SECOND
         STII    8
TIME:    SKT              ;USE INTERNAL TIMER TO FIND
         JP      TIME     ;THE 1/2 SECOND
BINPUT:  LBI     1,14     ;HAVE GOT IT, INCREMENT COUNTER
         SC
BINADD:  CLRA
         ASC
         NOP
         XIS
         JP      BINADD
         SKC              ;NOW SEE IF DONE
         JP      TIME     ;NO COUNTER OVERFLOW, CONTINUE
         LEI     0        ;DONE, DISABLE INTERRUPT
FIN:     ;AT THIS POINT HAVE THE VALUE--CONVERT IT TO DECIMAL OR
         ;SEND IT OUT OR PROCESS IT FURTHER, WHATEVER IS REQUIRED
         ;BY THE APPLICATION.  ARITHMETIC IS REQUIRED TO CREATE THE
         ;VOLTAGE VALUE, USUALLY A SIMPLE MULTIPLY
         ;MAY HAVE TO DOUBLE THE RESULT TO COMPENSATE LOOKING FOR
         ;ONLY 1/2 SECOND IN THIS CASE
         ;
         JP      MEASUR   ;DO IT OVER AGAIN
         .=X'OFF          ;SET ADDRESS TO OFF FOR INTERRUPT
INTENT:  NOP              ;ADDRESS OFF MUST BE NOP FOR INTERRUPT
INTRPT:  LBI     2,12     ;DO ADD OF THE VALUE FOR FREQ CNT
         SC
INTRI:   CLRA             ;STRAIGHT LINE THE CODE FOR SPEED
         ASC
         NOP
         XIS
         CLRA
         ASC
         NOP
         XIS
         CLRA
         ASC
         NOP
         XIS
         CLRA
         ASC
         NOP
         X
         LEI     2        ;ENABLE THE INTERRUPT AGAIN
         RET
```

TL/DD/6935-49

FIGURE 19A. V to F by Counting Pulses



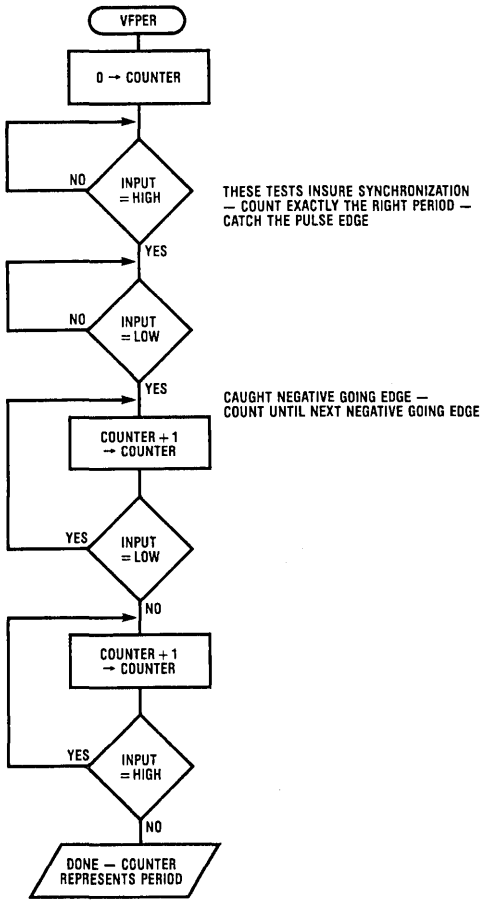
TL/DD/6935-27

FIGURE 19B. V to F by Counting Pulses

```
         ;USE INTERRUPT FOR CATCHING THE PULSE EDGE

VFPIN:   LBI     0,12     ;CLEAR COUNTER SPACE AND FLAG
         STII    0
         STII    0
         STII    0
         STII    0
         LBI     0,12
         LEI     2        ;NOW ENABLE THE INTERRUPT
WAIT:    SC               ;DUMMY WAIT LOOP, WAITING FOR SIGNAL TO
         LBI     0,12     ;INTERRUPT THE CONTROLLER
         JP      WAIT
         .=X'OFF          ;SET ADDRESS TO OFF--INTERRUPT ENTRY POINT
INTENT:  NOP              ;REQUIRED FOR INTERRUPT ENTRY
COUNT:   LBI     0,12     ;NOW CHECKING TO SEE IF SECOND INTERRUPT
         SKMBZ   0        ;I.E., ARE WE DONE?
         JP      DONE
         SMB     0        ;SET BIT FOR NEXT INTERRUPT
         LEI     2        ;ENABLE INTERRUPT AGAIN
PLUS1:   LBI     0,13     ;NOW START COUNTING
         SC
         CLRA             ;STRAIGHT LINE THE CODE FOR SPEED
         ASC
         NOP
         XIS
         CLRA
         ASC
         NOP
         XIS
         CLRA
         ASC
         NOP
         X
         JP      PLUS1
DONE:    ;FINISHED WHEN GET HERE--THE COUNT REPRESENTS THE PERIOD
         ;WITH ABOVE CODE, THE ACTUAL PERIOD IS THE COUNT MULTIPLIED
         ;BY 15(THE NUMBER OF WORDS TO INCREMENT BY 1) PLUS AN OVERHEAD
         ;OF 9 CYCLE TIMES =  24 CYCLE TIMES.  AT 4us THIS IS 96 us
         ;OR A FREQUENCY OF JUST OVER 10KHz.  MAX COUNT HERE IS 4095.
         ;THIS GIVES A MAXIMUM PERIOD = 61434 CYCLE TIMES(=245.736ms AT
         ;4us).  THIS CORRESPONDS TO A FREQUENCY OF JUST OVER 4Hz
         ;NOTE, THIS IS 12 BIT RESOLUTION
```

TL/DD/6935-50

FIGURE 19C. A to D with VF Converter/VCO by Measuring Period



TL/DD/6935-28

FIGURE 19D. V to F—Measure Period

5.2 THE LM131/LM231/LM331

The LM131 is a standard product voltage to frequency converter with a linear relationship between the input voltage and the resultant frequency. The reader should refer to the data sheet for the LM131 for further information on the device itself and precautions that should be taken when using the device. *Figure 20* is the basic circuit for using the LM131. *Figure 21* represents improvements that increase the accuracy (by increasing the linearity) of the result. Note that these circuits have been taken from the data sheet of the LM131 and the user is referred there for a further discussion of their individual characteristics. With the LM131 the frequency output is given by the relationship:

$$F_{OUT} = (V_{IN}/2.09)\,(1/R_TC_T)\,(R_S/RL)$$

It is clear from the expression above that the accuracy of the result depends upon the accuracy of the external components. The circuit may be calibrated by means of a variable resistance in the R_S term (a gain adjust) and an offset adjust. The offset adjust is optional but its inclusion in the circuit will allow maximum accuracy to be obtained. The standard calibration procedure is to trim the gain adjust (R_S) until the output frequency is correct near full scale. Then set the input to 0.01 or 0.001 of full scale and trim the offset adjust to get F_{OUT} to be correct at 0.01 or 0.001 of full scale. With that calibration, the circuit of *Figure 20* is accurate to within ±0.03% typical and ±0.14% maximum. The circuit of *Figure 21* attains the spec limit accuracy of ±0.01%.

5.3 VOLTAGE CONTROLLED OSCILLATORS (VCO's)

A VCO is simply another form of voltage to frequency converter. It is an oscillator whose oscillation frequency is dependent upon the input voltage. Numerous designs for VCO's exist and the reader should refer to the data sheets and application notes for various op-amps and VCO devices. The code in *Figure 19* is still applicable if a VCO is used. The only possible difficulty that might be encountered is if the relationship between frequency and input voltage is non-linear. This does not affect the basic code but would affect the processing to create the final result. A sample circuit, taken from the data sheet of the LM358, is shown in *Figure 22*. The accuracy of the VCO is the controlling factor.

5.4 A COMBINED APPROACH

Elements of the period measurement and pulse counting techniques can be combined to produce a system with the advantages of both schemes and with few problems. Such a system is only slightly more complicated in terms of its software implementation than the approaches mentioned above. Note that in a microcontroller driven system, no additional hardware beyond the voltage to frequency converter is required to implement this approach. Basically, the microcontroller establishes a viewing window during which time the microcontroller is both measuring time and counting pulses. The result can be very precise if two conditions are met. First, when the microcontroller determines that it needs the conversion information, the microcontroller does not begin counting time or pulses until the first pulse is received from the VFC (first pulse after the microcontroller "ready"). Note, the COPS microcontroller could provide a "start conversion" pulse to enable the VFC if such an arrangement were desirable. The time would be counted for a fixed period and the number of pulses would be counted. After the fixed period of time the controller would wait for the next pulse from the VFC and continue to count time until that pulse is received. The ratio of the total time to the number of pulse is a very precise result provided that all the system times are slow enough that the microcontroller can do its job. The speed limits mentioned previously apply here. It is clear that the total time is not fixed. It is some basic time period plus some variable time. This is a little more complicated than simply using a fixed time, but it allows greater accuracies to be achieved. Also, the approach takes approximately the same amount of time for all conversions. It is also faster than the simple pulse counting scheme.

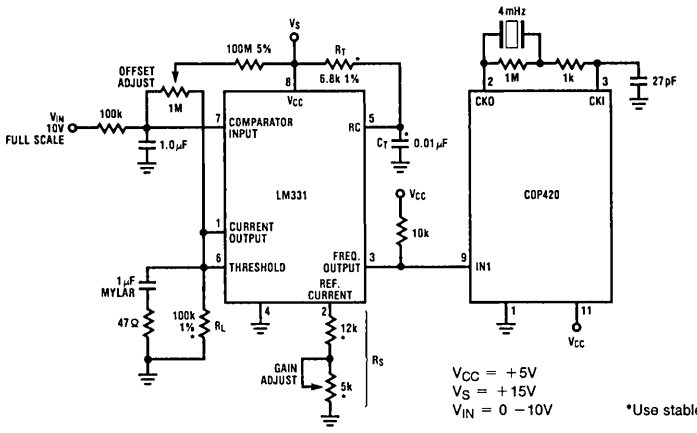


FIGURE 20. Basic LM331 Connection

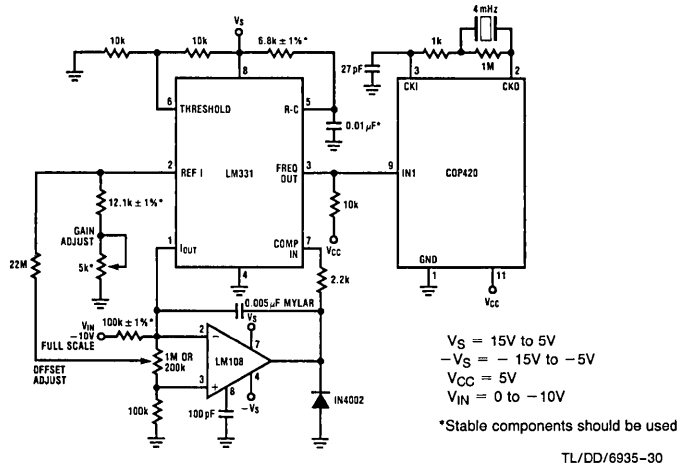


FIGURE 21. A to D with Precision Voltage to Frequency Converter

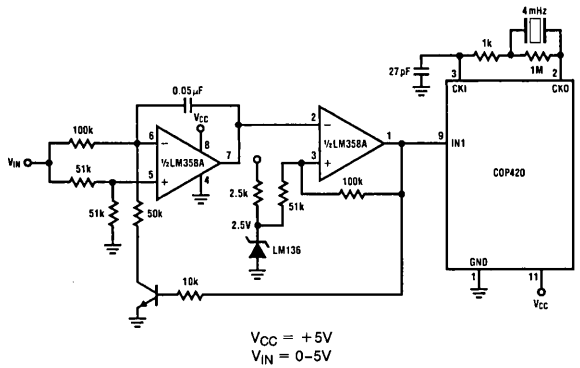


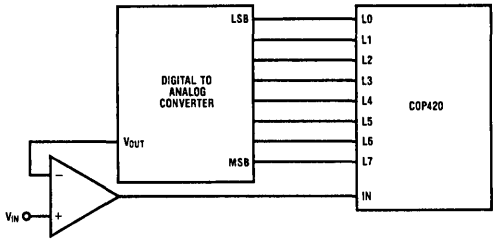
FIGURE 22. A to D with VCO

6.0 Successive Approximation

6.1 BASIC APPROACH

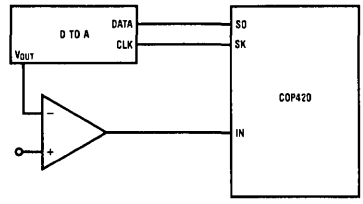
The successive approximation technique is one of the more standard approaches in analog to digital conversion. It requires a counter or register (here provided by the COP420), a digital to analog converter, and a comparator. *Figure 23A/B* illustrates the basic idea with the COP420. In the most basic scheme, the counter is reset to zero and then incremented until the voltage from the digital to analog converter is equal to the input voltage. The equality is determined by means of the comparator. *Figure 24B* illustrates the flow chart and code for this most basic approach. The preferred approach is illustrated in *Figure 25A/B*. This is the standard binary search method. The counter or register is set at the midpoint and the "delta" value set at one half the midpoint. The "delta" value is added or subtracted from the initial guess depending on the output of the comparator. The "delta" value is divided by 2 before the next increment or decrement. The method repeats until the desired resolution is achieved. While this approach is somewhat more complicated than the basic approach it has the advantage of always taking the same amount of time for the conversion regardless of the value of the input voltage. The conversion time for the basic approach increases with the input voltage. The preferred approach is almost always faster than the basic approach. The basic approach is faster only for those voltages near zero where it has only a few increments to perform.

The accuracy of the approach is governed by the accuracy of the digital to analog converter and the comparator. Thus, the result can be as accurate as one desires depending on the choice of those components. Digital to analog converters of various accuracies are readily available as standard parts. Their cost is usually in direct relation to their accuracy. The reader should refer to the National Semiconductor Data Acquisition Handbook for some possible candidates for digital to analog converters. It is not the purpose here to compare those parts. The COPS interface to these parts is generally straightforward and follows the basic schematics shown in *Figure 23*. The user should take note and make sure the input and output ports of the converter are compatible — in terms of voltages and currents — with the COPS device. This is generally not a problem as most of the parts are TTL compatible on input and output. The precautions and restrictions as to the use of any given device are governed by that device and are indicated in the respective data sheets.



TL/DD/6935-32

FIGURE 23A. Basic Parallel Implementation



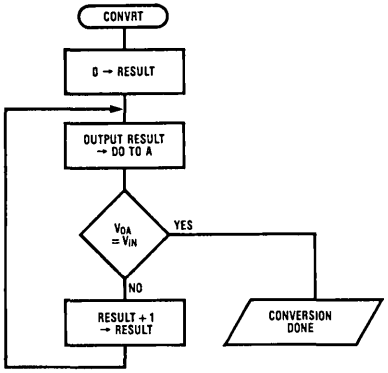
TL/DD/6935-33

FIGURE 23B. Basic Serial Implementation

```
                ;8 BIT SUCCESSIVE APPROXIMATION--BASIC SCHEME
                ;COMPARATOR INPUT TO COP = IN3
                ;OUTPUTS TO D TO A ARE L7 THRU L0 WITH L7 = MSB, L0 = LSB

CONVRT:  LBI     2,14      ;SET THE RESULT VALUE TO ZERO
         STII    0
         STII    0
         LEI     4         ;ENABLE THE L PORT AS OUTPUTS
         JP      OUTPUT
INCR:    SC                ;ROUTINE FOR INCREMENTING THE RESULT VALUE
PLUS1:   CLRA
         LBI     2,14
         ASC
         NOP
         XIS
         JP      PLUS1
OUTPUT:  LBI     2,15      ;SEND THE RESULT VALUE, STORED IN 2,15-2,14 TO
         LD                ;G AND THEREBY OUT THROUGH L
         XDS
         CAMQ
         JSR     DELAY     ;THIS IS ANY CONVENIENT ROUTINE TO MAKE SURE
                           ;THAT THE COP DOES NOT TEST THE COMPARATOR UNTIL
                           ;THE D TO A CONVERTER HAS HAD ENOUGH TIME TO DO
                           ;THE CONVERSION--THE AMOUNT OF TIME REQUIRED
                           ;IS CLEARLY DEPENDANT UPON THE D TO A CONVERTER
                           ;USED
         ININ              ;NOW READ THE COMPARATOR INPUT TO COP
         AISC    8         ;COULD SAVE A WORD IF USE G LINE AS INPUT
         JP      INCR      ;INPUT VOLTAGE STILL > CONVERTED ANALOG VOLTAGE

;CONVERSION DONE AT THIS POINT--THE COMPARATOR HAS CHANGED STATE
;HENCE, CONVERTED ANALOG VOLTAGE > INPUT VOLTAGE--SO STOP
```

TL/DD/6935-51

FIGURE 24A. Code for Basic Approach of Successive Approximation



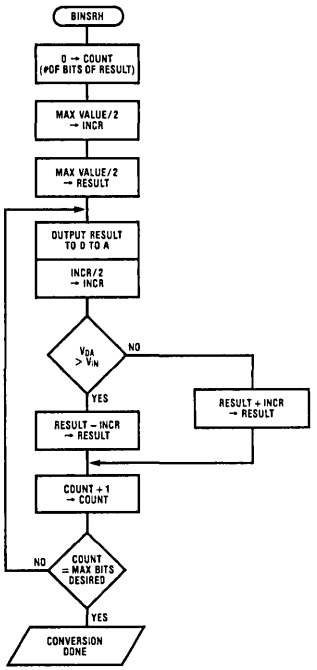
TL/DD/6935-34

FIGURE 24B. Basic Approach, Successive Approximation

```
        ;8 BIT BINARY SEARCH SUCCESSIVE APPROXIMATION
        ;INPUT TO COP IS IN3,L BUS IS OUTPUT TO D TO A,L7=MSB,L0=LSB
        ;COMPARATOR=0 WHEN D TO A VOLTAGE > VIN, OTHERWISE = 1

BINSRH: LBI     3,14    ;SET INCREMENT = MAX VALUE/2(WILL BECOME
        STII    0       ;MAX VALUE/4 BEFORE FIRST USE)
        STII    8
        LBI     2,14    ;SET INITIAL VALUE OF RESULT TO MAX VALUE/2
        STII    0
        STII    8
        LEI     4       ;ENABLE THE L BUS AS OUTPUTS
        LDI     1,15    ;NOW SET UP THE BIT COUNTER-OVERFLOW WHEN 8 BITS
        CLRA
        AISC    9       ;DO IT THIS WAY FOR COMPATIBILITY WITH INCREMENT
OUTPUT: X       3       ;SAVE THE BIT COUNTER VALUE AND POINT TO RESULT
        LD
        XDS             ;SEND THE RESULT TO G AND HENCE TO L
        CAMQ
DIVIDE: LBI     3,15    ;DIVIDE THE INCREMENT VALUE BY 2, CAN BE DONE
DIVA:   LD              ;IN SEVERAL WAYS SINCE THIS IS A VERY SPECIAL
        AISC    8       ;PURPOSE DIVIDE FUNCTION
        JP      DIV1    ;ALSO,DO THE DIVIDE HERE TO GIVE THE D TO A TIME
        STII    4       ;TO DO THE DIGITAL TO ANALOG CONVERSION
        JP      TEST
DIV1:   AISC    4
        JP      DIV2
        STII    2
        JP      TEST
DIV2:   AISC    2
        JP      DIV3
        STII    1
        JP      TEST
DIV3:   LBI     3,14
        AISC    1
        JP      DIVA
        STII    8
        STII    0
        ;DEPENDING ON THE D TO A USED, MAY NEED MORE DELAY HERE
        ;MUST BE SURE THE RESULT IS STEADY BEFORE TEST THE COMPARATOR
TEST:   LBI     3,14
        ININ
        AISC    8       ;COULD SAVE A WORD IF USED G LINE AS INPUT
        JP      INCR
DECR:   SC              ;INPUT LESS THAN D TO A CONVERTED VOLTAGE
SUB:    LD      1       ;SUBTRACT THE INCREMENT VALUE FROM RESULT
        CASC
        NOP
        XIS     1
        JP      SUB
        JP      BITPL1
INCR:   RC              ;INPUT > D TO A CONVERTED VOLTAGE
ADD:    LD      1       ;ADD THE INCREMENT VALUE TO RESULT VALUE
        ASC
        NOP
        XIS     1
        JP      ADD
BITPL1: LBI     1,15    ;NOW INCREMENT BIT COUNTER TO SEE IF DONE
        LD
        AISC    1
        JP      OUTPUT
        ;CONVERSION DONE AT THIS POINT
```

TL/DD/6935-52

FIGURE 25A. Binary Search Successive Approximation Code



TL/DD/6935-35

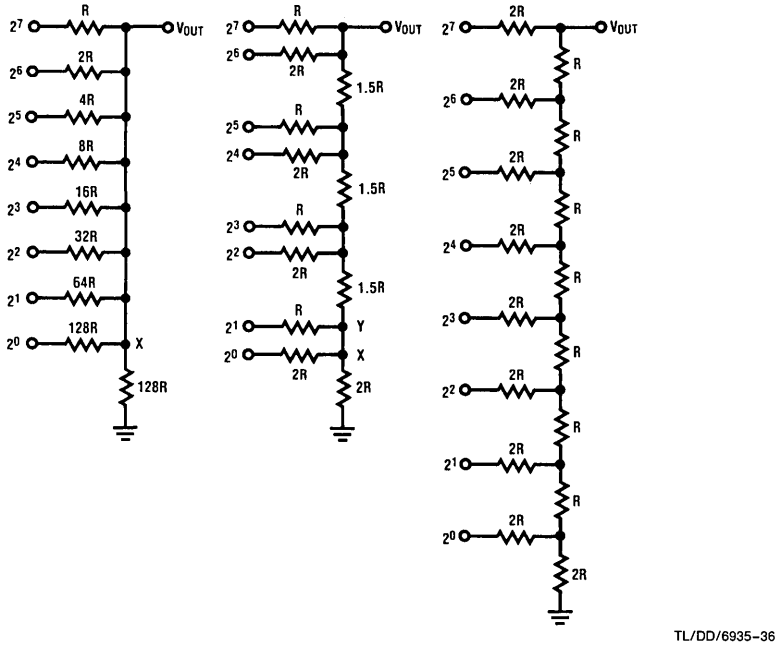
FIGURE 25B. Binary Search Successive Approximation Flow Chart

6.2 SOME COMMENTS ON RESISTOR LADDERS

If the user does not wish to use one of the standard digital to analog converters, he can always build one of his own. One of the most standard methods of doing so is to use a resistor ladder network of some form. *Figure 26* illustrates the basic forms of binary ladders for digital to analog converters. The figures also show the transition from the basic binary weighted ladder in *Figure 26A* to the standard R-2R ladder *Figure 26C*.

Consider *Figure 26A*. The choice of the terminating resistor is made by hypothesizing that the ladder were to go on ad infinitum. It can then be shown that the equivalent resistance at point X in that figure would be equal to 128R, the same value as the resistor to the least significant bit output. This fact is used to create the intermediate ladder of *Figure 26B*. This step is done because it is usually undesirable to have to find the multitude of resistor values required in the basic binary ladder. Thus, the modification in *Figure 26B* significantly reduces the number of resistor values required. As stated earlier, the resistance looking down the ladder at point X in *Figure 2* is equal to the resistor connected to the binary output at that point; here the value is 2R. Remembering the objective is to minimize the number of different values required, if we simply use the same R-2R arrangement as before with a termination of 2R we get an effective resistance at point Y of *Figure 26B* or 0.5R. This means that a serial resistance of 1.5R is required to maintain the integrity of the ladder. If we carry this on through 8 bits, the circuit of *Figure 26B* results. From this it is only a small step to create the standard R-2R network. The analysis is the same as done previously.

There is absolutely no restriction that the ladders must be binary. A ladder for any type of code can be constructed with the same techniques. Ladders comparable to *Figures 26A* and *26B* are shown in *Figure 27* for a standard 8421 BCD code. With the BCD code, the input must be considered in groups of digits with four bits creating one digit. This is the direct analog of 1 binary digit per unit. We need four inputs to create one decimal digit. Thus the resistor values in each decimal digit are 10 times the values in the previous decimal digit just as the resistor value for each successive binary digit was twice the value for the preceding binary digit. Note that this analysis can be easily extended to any code. The termination resistance is calculated in the same manner—assume the decimal digit groupings extend out to infinity. It can be shown that the resistance of the ladder at point X in *Figure 27A* is 480R. Thus *Figure 27A* represents the basic 8241 BCD ladder for three digit BCD number. This termination resistance will vary with where it is placed. Basically this resistance is equal to nine times (for a decimal ladder) the parallel resistance of the last digit implemented. (This relation can be shown mathematically if one desired, the multiplier is a function of the type of ladder used—multiplier = 1 for binary systems, 9 for decimal systems, etc.) Thus the termination resistance would be 48R if the network were terminated after the 2nd digit and 4.8R if the network were terminated after the 1st digit implemented. In



TL/DD/6935-36

FIGURE 26. Binary Ladders

Figure 27B we are attempting to use only the resistor values for one decimal digit. This means that the last terminating resistor must be a 4.8R by the analysis above. Thus at point X in *Figure 27B* we must have an equivalent of resistance of 4.8R. The equivalent resistance at point Y of *Figure 27B*, looking down from the ladder, is 0.48R. Thus the other series resistance must be 4.32 R (4.8R−0.48R). Thus the network of *Figure 27B* results.

Generally, ladders can be very effective tools when understood and used properly. They can be significantly more involved than indicated here. There are a number of texts and articles that cover the subject very nicely and the reader is referred to them if more information on ladder design, the use of ladders, and advanced techniques with ladders is desired.

One final note is of some interest. The ladders may be readily constructed for any type of code to create the analog voltage. Note that there is no restriction that the code, or the ladder network, be linear. Thus, effective use of ladder networks may significantly reduce system difficulties and complexities caused by the fact that the analog to digital conversion is being performed on a voltage source that changes nonlinearly, for example a thermistor temperature probe. By using the properly designed ladder network, the nonlinearity can effectively be eliminated from consideration in the code implementation of the analog to digital conversion.

The accuracy of ladders is a direct function of the accuracy of the resistors and the accuracy of the voltage source inputs. This is obvious since the analog voltage is in fact created by means of equivalent voltage dividers created when the various inputs are on or off. It is also essential that the ladder sources be the precise same value at all inputs to the ladder network. If this is not the case, errors will be introduced. In addition, the output impedance of the voltage source should be as small as possible. The success of the ladder scheme depends on the ratios of the resistance values. Inaccuracies are introduced if those ratios are disturbed. Some possible implementations of the successive approximation approach with a ladder network used for the digital to analog conversion are indicated in *Figure 28.*

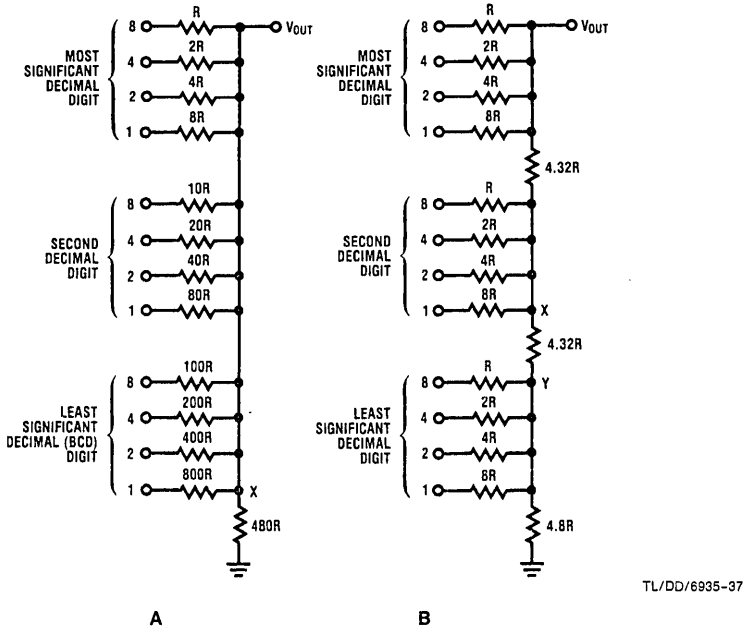
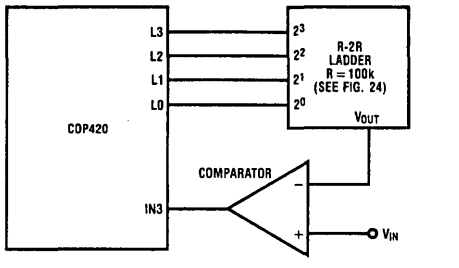


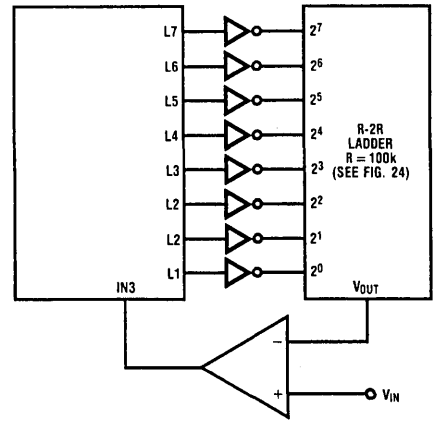
FIGURE 27. 8421 BCD Ladders

Note that these are functional diagrams. Feedback or hysteresis for comparator stabilization are not shown. The reader should be aware that his particular application may require that these factors be considered. *Figure 28A* is the simplest scheme and also the least accurate. With little or no load, the high output level of the L buffer should be very close to V_{CC} and the low level close to ground. Also the output impedance of the buffers must be considered. Therefore, rather large resistor values are used—both to keep the load very small and to dwarf the effect of the output impedance. With the configuration in *Figure 28A*, four bit accuracy is about the best that can be achieved. By being extremely careful and using measured values, an additional bit of accuracy may be obtained but care must be used. However, the schematic of *Figure 28A* is very simple. *Figure 28B* represents the next step of improvement. Here we have placed CMOS buffers in the network. This eliminates the output impedance and reduces the level problems of the circuit of *Figure 28A*. The CMOS buffer will swing rail to rail, or nearly so. The accuracy of V_{CC} and the resistor network is then



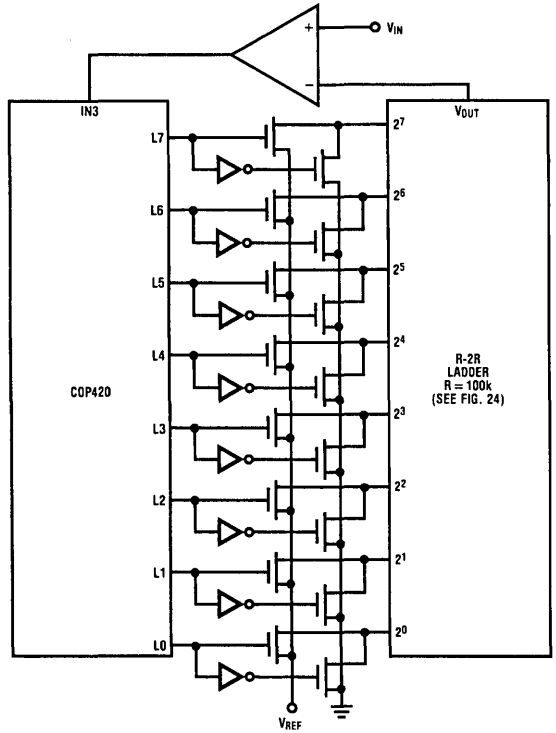
TL/DD/6935-38

A



TL/DD/6935-39

B



TL/DD/6935-40

C

FIGURE 28. Interfaces to Ladder Networks

controlling. Using 1% resistors and holding V_{CC} constant, the user should be able to achieve 7 to 8 bit accuracy without much difficulty. Remember, however, that V_{CC} is one of the controlling factors. If V_{CC} is ±5%, there is no point in using 1% resistors since the V_{CC} tolerance swamps their effect. *Figure 28C* is the final and most accurate approach. Naturally enough, it is the most expensive. However, one can get as accurate as one desires. Here, an accurate reference is required. That reference is switched into the network by means of the analog switch. Alternately, ground may be connected to the input. Now the user need only consider the accuracy of the reference and the accuracy of the resistors. However, the on impedance of the switches must be considered. It is necessary to make this on impedance as low as possible so as not to alter the effective resistor values.

7.0 "Offboard" Techniques

7.1 GENERAL COMMENTS

This section is devoted to a few illustrations of interfacing the COP420 to standard, stand alone analog to digital converters. These standard converters are used as peripherals to the COPS device. Whenever the microcontroller requires a new reading of some analog voltage, it simply initiates a read of the peripheral analog to digital converter. As a result, the accuracies and restrictions in using the converters are governed by those devices and not by the COPS device. These techniques are generally applicable to other A to D converters not mentioned here and the user should not have difficulty in applying these principles to other devices. It should be pointed out that in almost every instance, the choice of COP420 inputs and outputs is arbitrary. Obviously, when there is an 8-bit bus it is natural, and most efficient, to use the L port to interface to the bus. Generally, the G lines have been used as outputs rather than the D lines simply because the G lines are, in many instances, somewhat easier to control. The choice of input line is also free. If the interrupt is not otherwise being used, it may be possible to utilize this feature of IN1 for reading a return signal from the converter. However, this is by no means required. If there is a serial interface it is clearly more efficient to use the serial port of the COP420 as the interface. If a clock is required, SK is the natural choice.

7.2 ADC0800 INTERFACE

The ADC0800 is an 8-bit analog to digital converter with an 8-bit parallel output port with complementary outputs. The ADC0800 requires a clock and a start convert pulse. It generates an end of conversion signal. There is an output enable which turns the outputs on in order to read the 8-bit result.

The reader is referred to the data sheet for the ADC0800 for more information on the device. The circuit of *Figure 29* illustrates the basic implementation of a system with the ADC0800. The interface to the COP420 is straightforward. The appropriate timing restrictions on the control signals are easily met by the microcontroller.

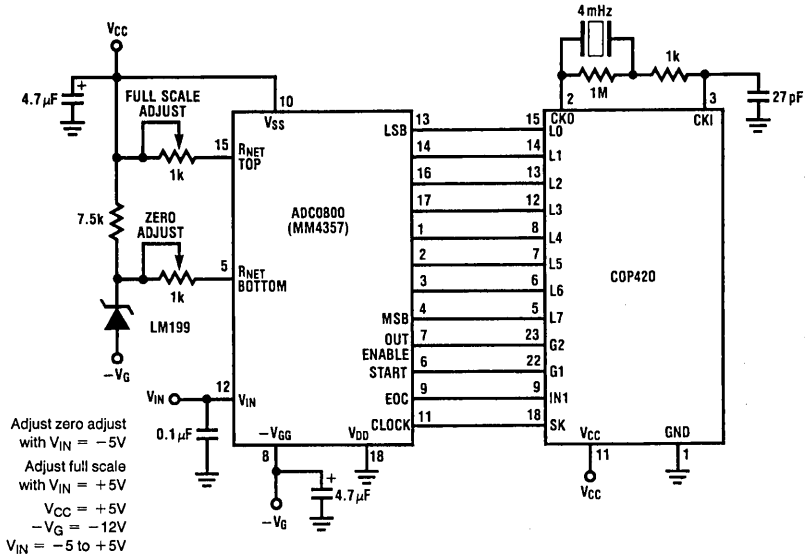


FIGURE 29. Simple A/D with ADC0800

Figure 30 is the flow chart and code required to do the interfacing. As can be seen, the overhead in the COP420 device is very small. The choice of inputs and outputs is arbitrary. The only pin that is more or less restricted is the use of SK as the clock for the converter. SK is clearly the output to use for that function as, when properly enabled, it provides pulses at the instruction cycle rate.

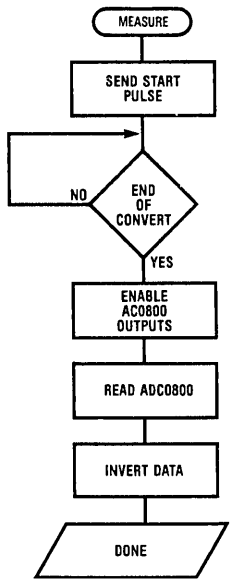
7.3 ADC0801/2/3/4 INTERFACE

The ADC0801 family of analog to digital converters is very easy to interface and is generally a very useful offboard converter. The interface is not significantly different from that of the ADC0800, but the ADC0801 famliy are a much better device. The four control signals are somewhat different, although there are still four control lines. Here we have a chip select, a read, a write, and an interrupt signal. All are negative going signals. Start conversion is the ANDing of chip select and write. Output enable is the ANDing of chip select and read. The interrupt output is an end convert signal of sorts. The device may be clocked externally or an RC may be connected to it and it will generate its own clock for the conversion. In addition the device has differential inputs

```
MEASUR: LEI     0       ;FLOAT THE L LINES
        SC
START2: CLRA            ;MAKE SURE SO STAYS ZERO
        XAS             ;MAKE SURE SK STAYS CLOCK
        OGI     2       ;SEND START PULSE
        OGI     0
        LBI     2,13
READI1: ININ
        AISC    14      ;WAIT FOR EOC SIGNAL
        JP      READI1
        OGI     4       ;HAVE EOC, ENABLE OUTPUTS
        INL             ;READ THE L LINES
        X
        COMP            ;CREATE PROPER POLARITY
        XDS
        COMP
        X
        OGI     0       ;DISABLE ADC0800 OUTPUT
        ;HAVE THE RESULT AT THIS POINT--USE IT IN WHATEVER
        ;MANNER IS REQUIRED BY THE APPLICATION
        LBI     2,10
        JSRP    CLRR
        JP      MEASUR
```

TL/DD/6935-53

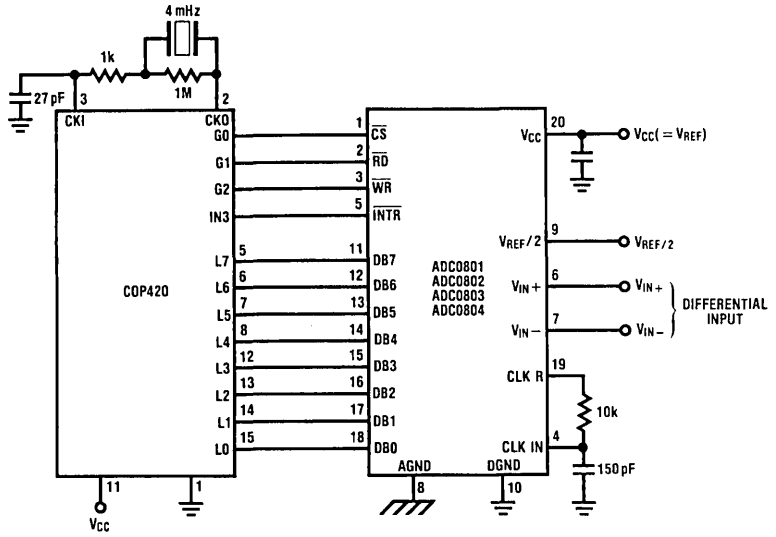
FIGURE 30A. A to D with ADC0800



TL/DD/6935-42

FIGURE 30B. ADC0800 Interface Flow

which allow the 8-bit conversion to be performed over a given window or range of input voltages. The reader should refer to the ADC0801 family data sheet for more information. *Figure 31* indicates a basic interface of the ADC0801 family to the COP420. Again, the interface is simple and straightforward. The code required to interface to the device is minimal. *Figure 32* illustrates the flow chart and code required to do the interface.



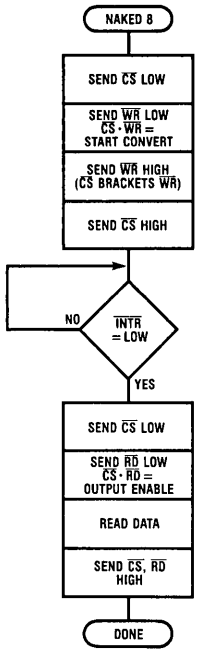
TL/DD/6935-43

FIGURE 31. COP420—ADC0801 Family Interface

```
        ; INTERFACE TO NAKED 8
        ;
NAKED:  OGI     15      ; SET ALL G LINES HIGH(USUALLY DONE AT
                        ; POWER UP
        LEI     0       ; TRI STATE THE L LINES FOR READING
LOOP:   OGI     14      ; SEND CHIP SELECT LOW(CS BRACKETS OTHER SIGNAL)
        OGI     10      ; CS LOW AND WR LOW = START CONVERSION
        OGI     14      ; RAISE WR
        OGI     15      ; RAISE CS, NAKED 8 IS NOW CONVERTING
LOOP2:  ININ            ; WAIT FOR THE INTR SIGNAL--COULD SAVE THIS TES
        AISC    8       ; IF USED IN1 AND THE INTERRUPT FEATURE OF COP4
        JP      READ    ; INTR IS LOW, DATA IS READY
        JP      LOOP2
READ:   LBI     0,0     ; SET UP RAM LOCATION FOR READ
        OGI     14      ; SEND CS
        OGI     12      ; SEND CS AND READ = OUTPUT ENABLE
        NOP             ; WAIT-NEED WAIT ONLY 125NS, BUT 1 CYCLE IS MIN
                        ; TIME WE CAN WAIT
        INL             ; READ THE L LINES
        OGI     15      ; TURN OFF THE NAKED 8--CS AND RD HIGH
        ;
        ; DONE AT THIS POINT, DO WHATEVER IS REQUIRED  WITH THE RESULT
        ;
```

TL/DD/6935-54

FIGURE 32A. COP420/ADC0801 Family Sample Interface Code



TL/DD/6935–44

FIGURE 32B. COP420/ADC0801 Family Interface Flow

8.0 Conclusion

Several analog to digital techniques using the COPS family have been presented. These are by no means the only techniques possible. The user is limited only by his imagination and whatever parts he can find. The COPS family of parts is extremely versatile and can readily be used to perform the analog to digital conversion in almost any method. Generally, those techniques where the COPS device is doing the counting or timekeeping are slow. However, those techniques are generally slow inherently. The fastest methods are those where the conversion is being done offboard and the COPS device is merely reading the result of the conversion when required. Also, an attempt has been made to illustrate the lower cost techniques of analog to digital conversion. This, by itself, restricts most of the techniques described to about 8-bits accuracy. As was mentioned several times, the greater the accuracy that is desired the more accurate the external circuits must be. Ten and twelve-bit accuracies, and more, require references that are accurate. These get very expensive very rapidly. There is nothing inherent in the COPS devices that prevents them from being used in accurate systems. The precautions are to be taken in the system regardless of the microcontroller. The only problem is that, in those accurate systems where the COPS device is doing the timekeeping and counting, this increased accuracy is paid for by increased time to perform the conversion.

Several devices have been used in conjunctions with the COPS device in the previous sections. It is again recommended that the user refer to the specific data sheets of those devices when using any of those circuits. It must again be mentioned that the standard precautions when dealing with analog signals and circuits must be taken. These are described in the National Semiconductor Linear Applications Handbook and in the data sheets for the various linear devices. These precautions are especially significant when greater accuracy is desired.

The COPS family of microcontrollers has shown itself to be very versatile and powerful when used to perform analog to digital conversions. Most techniques are code efficient and the microcontroller itself is almost never the limiting factor. It is hoped that this document will provide some guidance when it is necessary to perform analog to digital conversion in a COPS system.

9.0 References

1. "Digital Voltmeters and the MM5330", National Semiconductor Application Note AN-155.
2. Walker, Monty, "Exploit Ladder Network Design Potential". Part One of two part article on ladder networks. Magazine and date unknown.
3. Wyland, David C., "VFC's give your ADC design high resolution and wide range". *EDN*, Feb. 5, 1978.
4. Redfern, Thomas P., "Pulse Modulation A/D Converter" *Society of Automotive Engineers Congress and Exposition Technical paper #780435*, March 1978.
5. National Semiconductor Linear Applications Handbook, 1978.
6. National Semiconductor Linear Databook, 1980.
7. National Semiconductor Data Acquisition Handbook, 1978.

The COP444L Evaluation Device 444L-EVAL

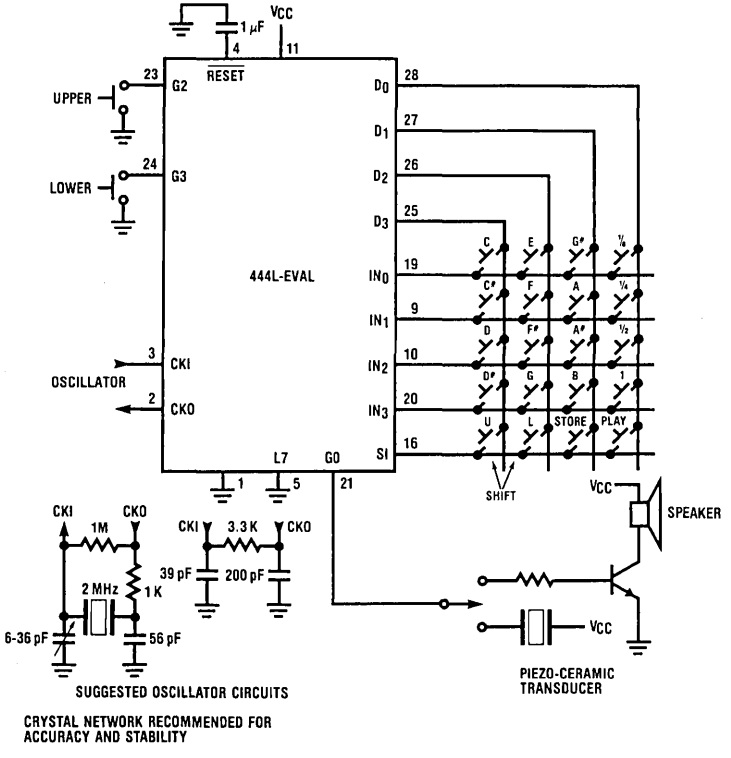
National Semiconductor Corp.
COP Note 4
Leonard A. Distaso

The 444L-EVAL is a preprogrammed COP444L intended to demonstrate operating characteristics and facilitate user familiarization and evaluation of the COP444L and the COPS™ family in general.

The 444L-EVAL has two mutually exclusive operating modes: an up/down counter/timer or a simple music synthesizer. The state of pin L7 at power up determines the operating mode.

1.0 THE 444L-EVAL AS A SIMPLE MUSIC SYNTHESIZER

Figure 1 indicates the connection of the 444L-EVAL as a simple music synthesizer. As the diagram indicates, the connections required for operation are minimal. The oscillator may be a crystal circuit using CKI and CKO; an external oscillator to CKI; or an RC network using CKI and CKO. As should be expected, the crystal circuit provides the greatest frequency stability and precision. The RC network will provide an acceptable oscillation frequency but that frequency will be neither precise nor stable over temperature and voltage. The external oscillator, of course, is as good as its source. The frequencies for the various notes and delay times are set up assuming that the oscillator frequency is 2 MHz. Three modes of operation are available in the music synthesizer mode: play a note; play one of four stored tunes; or record a tune for subsequent replay.



TL/DD/6937–1

FIGURE 1. 444L-EVAL as Simple Music Synthesizer

1.A. PLAY A NOTE

Twelve keys, representing the twelve notes in one octave, are labeled "C" through "B". Depressing a key causes a square wave of the corresponding frequency to output at GO. The user may drive a piezo-ceramic transducer directly with this signal. With the appropriate buffering, the user may use this signal to drive anything he wishes. A simple transistor driver is sufficient to drive a small speaker. The user can be as simple or as complex as he desires at this point—e.g. he can do some wave shaping, add an audio amplifier, and drive a high quality speaker.

The 444L-EVAL has a range of two and one-half octaves: the basic octave on the keyboard (which is middle C and the 11 notes above it in the chromatic scale), one full octave above the basic octave and one-half octave below the basic octave. The notes in the basic octave are played by depressing the appropriate key (one key at a time—the keyboard has no rollover provisions). A note in the upper octave is played by first depressing and releasing the U SHIFT key and then depressing the note key. Similarly, a note in the lower one-half octave is played by first depressing and releasing the L SHIFT key and then depressing the note key. Two other shift keys are present: UPPER and LOWER. All notes played while the UPPER key is held down will be in the upper octave. Similarly, note F# through B when played while the LOWER key is held down will be in the lower one-half octave. The lower octave notes C through F are not present and depressing any of these 6 keys while the LOWER key is held down or after depressing the L SHIFT key will play the note in the basic octave.

1.B. PLAY STORED TUNE

The 444L-EVAL can play four preprogrammed tunes. Depressing PLAY followed by "⅛", "¼", "½", or "1" will cause one of these tunes to be played. The tunes are:

PLAY 1 —Music Box Dancer

PLAY ½ —Santa Lucia

PLAY ¼ —Godfather Theme

PLAY ⅛ —Theme from Tchaikowsky Piano Concerto #1

1.C. RECORD A TUNE

Any combination of notes and rests up to a total of 48 may be stored in RAM for later replay. A note is stored by depressing the appropriate key(s), followed by the duration of the note (1/16 note, ⅛ note, 3/16 note, ¼ note, ⅜ note, ½ note, ¾ note, whole(1) note), followed by STORE. A rest is stored by selecting the duration and depressing STORE. The rests or durations of 1/16, 3/16, ⅜, and ¾ are obtained by first depressing L SHIFT and then ⅛, ¼, ½, or 1 respectively. When the tune is complete press PLAY followed by STORE. The tune will be played for immediate audition. Subsequent depression of PLAY and then STORE will play the last stored tune.

Only one tune may be stored, regardless of length. Attempts to store a new or second tune will erase the previously stored tune. There are no editing features in this mode. (In a "real system" of this type some form of editing would be desirable. It would not be difficult to add editing features.)

Note: The accuracy of the tones produced is a function of the oscillator accuracy and stability. The crystal oscillator, or an accurate, stable external oscillator is recommended.

2.0. THE 444L-EVAL AS AN UP/DOWN COUNTER/TIMER

By connecting pin L7 to V_{CC} and providing power and oscillator the 444L-EVAL functions as an 8 digit binary/BCD up/down counter. In addition, an approximate 1 Hz signal is produced by the device. The 444L-EVAL can drive a single digit LED display directly. With the appropriate driver (COP472, COP470, MM5450/5451) the device can drive a 4 digit LCD, VF, or LED display. Any combination of these displays can be connected at any given time.

The binary/BCD and and up/down modes are controlled by the states of input pins IN0 and IN2 as indicated below:

IN0 = 1 (Default state) —BCD counter
IN0 = 0 —Binary Counter
IN2 = 1 (Default state) —Count Up
IN2 = 0 —Count Down

The up/down control may be changed at any time. Changing the binary-BCD control during operation clears the counter before counting begins in the new mode.

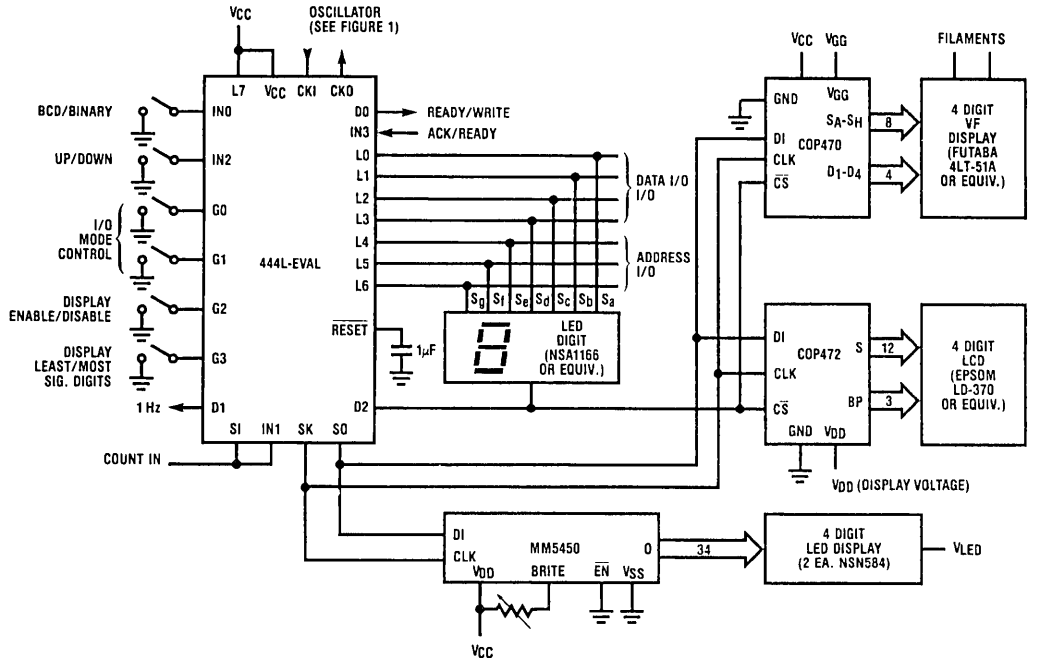
Pins G2 and G3 provide display control to the user. He can choose to view either the most significant 4 digits of the counter or the least significant 4 digits of the counter. Further, the user can disable the update of the 4 digit displays. The controls are as follows:

G2 = 1 (Default state) —Enable update of 4 digit displays
G2 = 0 —Disable update of 4 digit displays
G3 = 1 (Default state) —Display least significant 4 digits of counter
G3 = 0 —Display most significant 4 digits of counter

The single digit LED display displays the least significant digit of the counter. (Note, the direct drive capability for the single digit LED display refers to a small LED digit—NSA1541A, NSA1166k, or equivalent.)

2.A. I/O MODE

The 444L-EVAL has the capability to allow the user to read or write the 8 digit counter through the L port. In the I/O mode, the single digit LED display is disabled. The 4 digit displays are not affected. In this mode pins D0 and IN3 are used for the handshaking sequence. D0 is a Ready/Write signal from the 444L-EVAL to the outside; IN3 is a Write/Acknowledge from the outside to the 444L-EVAL. Data I/O is via L0–L3 with L0 being the least significant bit. Data is standard BCD for the BCD counter mode or standard hex for the binary counter mode. The digit address is on pins L4–L6 with L4 being the least significant bit. Digit address



TL/DD/6937–2

FIGURE 2. 444L-EVAL in Counter Mode

0 is the least significant digit of the counter; digit address 7 is the most significant digit of the counter. The I/O modes are controlled by pins G0 and G1 as follows:

G0	G1	
0	0	Output data with handshake, single digit LED off
0	1	Input data with handshake, single digit LED off
1	0	Auto output, no handshake, single digit LED on
1	1	Default condition, No I/O, single digit LED displays least significant digit of counter

2.A.1. Output Data with Handshake

With this mode selected the 444L-EVAL will output data with a handshake sequence. Note that the outputting of data is relatively slow as the device is counting and updating displays between successive digit outputs.

Before data is output, or the next digit of the counter is output, the 444L-EVAL must see IN3 (Acknowledge or ready from the external world high). The Ready/Write pin (D0) is assumed to be high at this point. With D0 high and IN3 high, the device will output the data and digit address. After the data and address are output, the D0 line—functioning as a write strobe here—goes low. The 444L-EVAL then expects the signal at IN3 to go low indicating that the external world has read the data. When the device sees IN3 go low, D0 will be brought high indicating that the sequence is ready to repeat as soon as IN3 goes high again. The counter digits are output sequentially from least significant digit (digit address 0) through most significant digit (digit address 7). The sequence will continuously repeat as long as this mode is selected.

2.A.2. Input Data with Handshake

The 444L-EVAL will take data supplied to it and load the counter. The sequence is similar to that described above for the output mode. The external device(s) supplies both the data and the digit address where that data is to be loaded.

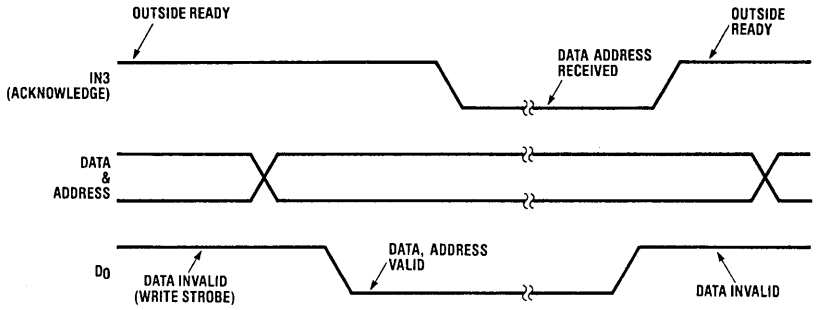
When sending data to the 444L-EVAL, the external circuitry must test that the device is ready to receive data (D0 high). Then the data and address should be presented at the L port. Then the Write signal (IN3) should be driven low. The 444L-EVAL will read the data and then drive D0 low. When D0 goes low, the external circuitry should bring IN3 high. After IN3 returns high, the 444L-EVAL will signal it is ready to receive data by sending D0 high. Note that this sequence is relatively slow. The 444L-EVAL is performing several operations between successive read operations.

2.A.3. Automatic Output Mode

In the automatic output mode, the single digit LED is on. It is not displaying the least significant digit of the counter in this mode. The display is on so that the user can connect this LED digit, select the automatic output mode, and observe the states of the L lines without having to put more sophisticated equipment or circuitry external to the 444L-EVAL. Segments a through d are pins L0 thorugh L3; segments,

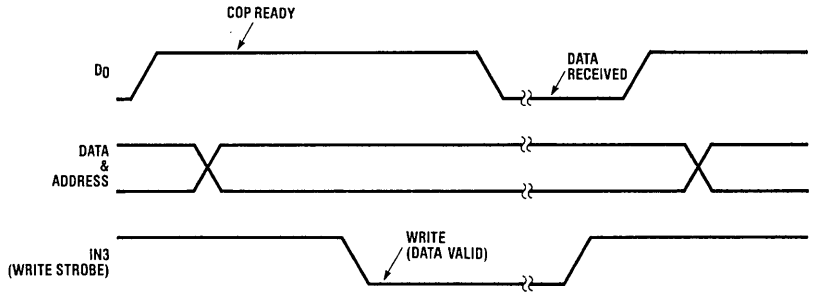
e, f, g are pins L4, L5, and L6. Thus the user can observe the digit address changing and observe the corresponding data.

In this mode, the state of pin IN3 is irrelevant. The 444L-EVAL sequentially outputs the digits of the counter. D0 goes high when the data and address is being changed. D0 goes low when the data is valid. As in the other I/O modes, the process is slow. There is about 4 to 5 milliseconds between the successive digit outputs.



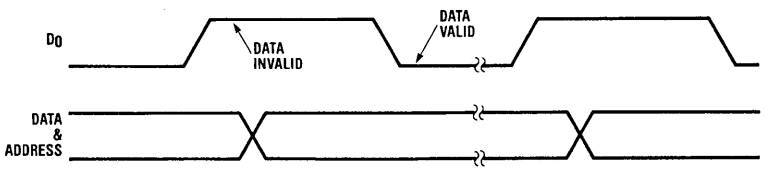
TL/DD/6937–3

FIGURE 3A. Relative Timing—Output Handshake



TL/DD/6937–4

FIGURE 3B. Relative Timing—Input Handshake



TL/DD/6937–5

FIGURE 3C. Relative Timing—Automatic Output

3.0 SELECTED OPTIONS

The 444L-EVAL has the following options selected:

GND	Option 1 = 0	
CKO	Option 2 = 0	CKO is clock generator output to crystal
CKI	Option 3 = 0	CKI oscillator input divide by 32
RESET	Option 4 = 0	Load device to V_{CC} on RESET
L7	Option 5 = 0	Standard output on L7
L6	Option 6 = 2	High current LED direct segment drive on L6
L5	Option 7 = 2	High current LED direct segment drive on L5
L4	Option 8 = 2	High current LED direct segment drive on L4
IN1	Option 9 = 0	Load device to V_{CC} on IN1
IN2	Option 10 = 0	Load device to V_{CC} on IN2
V_{CC}	Option 11 = 1	4.5V to 9.5V operation
L3	Option 12 = 2	High current LED direct segment drive on L3
L2	Option 13 = 2	High current LED direct segment drive on L2
L1	Option 14 = 2	High current LED direct segment drive on L1
L0	Option 15 = 2	High current LED direct segment drive on L0
SI	Option 16 = 0	Load device to V_{CC} on SI
SO	Option 17 = 2	Push-pull output on SO
SK	Option 18 = 2	Push-pull output on SK
IN0	Option 19 = 0	Load device to V_{CC} on IN0
IN3	Option 20 = 0	Load device to V_{CC} on IN3
G0	Option 21 = 0	Very high current standard output on G0
G1	Option 22 = 2	High current standard output on G1
G2	Option 23 = 4	Standard LSTTL output on G2
G3	Option 24 = 4	Standard LSTTL output on G3
D3	Option 25 = 0	Very high current standard output on D3
D2	Option 26 = 0	Very high current standard output on D2
D1	Option 27 = 0	Very high current standard output on D1
D0	Option 28 = 0	Very high current standard output on D0
	Option 29 = 0	Standard TTL input levels on L
	Option 30 = 0	Standard TTL input levels on IN
	Option 31 = 0	Standard TTL input levels on G
	Option 32 = 0	Standard TTL input levels on SI
	Option 33 = 1	Schmitt trigger inputs on RESET
	Option 34 = 0	CKO input levels, not used here
	Option 35 = 0	COP444L
	Option 36 = 0	Normal RESET operation

4.0 CONCLUSION

The 444L-EVAL demonstrates much of the capability of the COP444L. It does not indicate the limits of the device by any means. The I/O features were included to demonstrate that capability. The fact that they are slow is due strictly to the program. If such I/O capability were a necessary part of an application it could be accomplished much much faster than was done here. The counter modes are quite versatile and are generally self explanatory. It was fairly easy to provide a counter with the versatility of that included here. The music synthesis mode demonstrates clearly the program efficiency of the device.

The 444L-EVAL is intended for demonstration. There is no question that aspects of its operation could be improved and tailored to a specific application. It is unlikely that this particular combination of features would be found in any one application. It is also interesting to note that the program memory in the device is not full. There is still a significant amount of room left in the ROM. This should serve to make it clear that the capabilities of the device have not been stretched at all in order to include these demonstration functions.

Oscillator Characteristics of COPS™ Microcontrollers

National Semiconductor Corp.
COP Note 5

Table of Contents

1.0 INTRODUCTION

COPS microcontrollers will operate with a wide variety of oscillator circuits. This paper focuses on two of the oscillator options available on COPS microcontrollers: the internal RC oscillator, and the crystal or inverter oscillator. The typical behavior of the RC oscillator with temperature and voltage (and typical values of R and C) is documented. For the crystal or inverter option, circuit configurations (RC, RL, RLC, R, LC, L) are presented which will allow the microcontroller to operate properly without the use of ceramic resonator or crystal.

The passive components used were inexpensive, uncompensated devices: standard carbon resistors, ceramic or foil capacitors, and air core or iron core inductors. To provide reasonably clear data on the characteristics of the microcontroller itself, no attempt at compensation for the external components was made.

2.0 RC OSCILLATOR OPTION

With the RC oscillator option selected, the graphs in *Figures 1* through *6* indicate the variation of the instruction cycle time of the microcontroller with temperature and voltage. Typical R and C values, as recommended in the respective device data sheets, were used. The graphs are composite graphs reflecting the worst case variations of the devices tested. Therefore, the graphs show a percentage change of the instruction cycle time from a base or reference value. Where the results are plotted against voltage the reference is the value at V_{CC} = 5V. Where the results are plotted against temperature, the reference is the value at T = 20°C. A positive percent variation indicates a longer instruction cycle time and therefore a slower oscillator frequency. Similarly, a negative percent variation indicates a shorter instruction cycle time and therefore a faster oscillator frequency.

The measurements were taken by holding the RESET pin of the device low and measuring the period of the waveform at pin SK. In this mode the SK period is the instruction cycle time. For divide by 4 the oscillator frequency is given by the following:

$$\text{frequency} = \frac{4}{\text{SK period}}$$

Measurements were taken at temperatures between −40°C and +85°C and at V_{CC} values between 4.5V and 9.5V. However, the reader must remember that the COP400 series is specified only between 0°C and +70°C. The reader must also remember that the COP420 is specified at V_{CC} levels between 4.5V and 6.3V only. The data here is usable for the COP300 series, which is specified at the extended temperature range of −40°C to +85°C. However, the reader must keep in mind the generally more restricted V_{CC} range for some of the various COP300 series microcontrollers.

The graphs in *Figures 1* through *6* reflect the variation of the microcontroller only. The resistor and capacitor were not in the temperature chamber with the COPS device. Obviously, the results will be affected by the variation of the R and C with temperature. However, this can vary dramatically with the type of components used. The user will have to combine the data here with the characteristics of the external components used to determine what type of variation may be expected in his system.

3.0 CRYSTAL OR INVERTER OPTION

With the crystal or inverter option selected on the COPS microcontroller there is, effectively, an inverter between the CKI and CKO pins. CKI is the input to the inverter and CKO is the output. Various passive circuits were connected between CKI and CKO and the results documented. Of the operational circuits, a subset was tested over temperature with the microcontroller only in the temperature chamber. A smaller subset was tested over temperature with both the microcontroller and the oscillator network in the temperature chamber.

The data with the oscillator network in the temperature chamber is obviously highly dependent on the particular components used. This data was taken with standard, inexpensive, uncompensated components. Neither high precision nor high stability components were used. This data is included only to provide the user with some very general indication of how the oscillator frequency may vary with temperature in a real system.

3.1 COP420/COP402

Except for the ROM, the COP420 and COP402 are equivalent devices. The internal circuitry of each device is identical. Therefore, data taken for one of the devices is equally

applicable to the other. The following discussion will refer to the COP420 but all such references apply equally well to the COP402. Similarly, the graphs for the COP420 apply to the COP402 and *vice versa*.

With the crystal option selected, the COP420 oscillator circuitry will readily oscillate with almost any circuit configuration between CKI and CKO. What difficulty there is lies in finding the network of the device. With the appropriate divide option selected, oscillator frequencies between 800 kHz and 4 MHz are valid for the COP420. No data was taken for any network that produced an oscillation frequency outside the valid range.

3.1.1 L, LC, and RLC Networks

Various L, LC, and RLC networks were connected with varying results. Certain networks produced results much more stable than the RC networks; others were no better than the RC networks. With a single inductor connected between CKI and CKO, frequencies between 1 MHz and 4 MHz were easily obtained. However, the input gate capacitance at CKI (typically 5 pF to 10 pF) and the series resistance of the inductance become factors that impact the oscillation frequency and its stability over temperature.

The addition of a capacitor between CKI and ground tends to reduce the effects of the internal gate capacitance. For the single L, single C network of this type, the capacitor value should be greater than about 50 pF to begin to effectively swamp out the effects of the input gate capacitance. As might be expected, LC combinations which had their resonant frequencies within the valid COP420 frequency range produced the best results.

The addition of another capacitor(s) to the basic two-component LC network, as shown in *Figure III.1*, produced very good results. Varying the capacitor values in these networks — especially those capacitors between CKI and ground and CKO and ground — provided a great deal of control over the oscillation frequency. In *Figure III.1*, varying C1 from 25 pF to 0.01 μF produced oscillation frequencies between about 3 MHz and 1.6 MHz (C2 = 25 pF, L = 56 μH). In *Figure III.2*, with C1 = 330 pF, L = 56 μH, and C2 = 27 pF, varying C3 between 10 pF and 0.003 μF produced oscillation frequencies between about 2 MHz and 1.1 MHz. Varying C2 in *Figure 111.3* produced a similar kind of control.

As the graphs indicate, various types of RLC networks were also tried. The range of possible usable circuits here is limited only by the user's imagination and his favorite type of RLC oscillator circuit. When their resonant frequency is within the valid frequency range of the COP420, LC and RLC networks can be a very effective substitute for a crystal. The only potential problem is that a good RLC, or even LC, oscillator circuit may not be a cost-effective substitute for a crystal in a COP420 system. The user will have to make that determination.

3.2 COP420L

The valid input frequency range for the COP420L, with the appropriate divide option selected, is between 200 kHz and 2.097 MHz. With the crystal option selected the COP420L oscillated much less readily than the COP420.

The LC networks gave outstanding results with the COP420L. With the simple two-component LC network shown in the graphs, holding C at 50 pF and varying L from 200 μH to 700 μH gave oscillation frequencies from about 2 MHz to 1 MHz. Holding L at 390 μH and varying C from 10 pF to 700 pF gave oscillation frequencies of about 2 MHz to 1.6 MHz. Similar results were obtained when a capacitor was placed in parallel with the inductance.

3.3 COP410L

The COP410L has a valid input frequency range of 200 kHz to 530 kHz.

The LC networks also gave very good results. With the simple LC network shown in the graphs, holding L at 4700 μH and varying C from 25 pF to 0.003 μF gave oscillation frequencies of about 460 kHz to 225 kHz.

3.4 GENERAL NOTES

With the crystal or inverter option selected on COPS microcontrollers, a wide variety of networks may be used in place of the ceramic resonator or crystal.

LC and RLC networks can be used in any of the devices. Appropriately designed, these networks will provide a stable oscillation frequency for the microcontroller. The user will have to allow for the variation of the external components with temperature when using these networks. The problems with networks such as these is that they may not be cost-effective alternatives to the crystal or resonator, especially if high stability, temperature compensated components are used. The user will have to make the determination of cost-effectiveness.

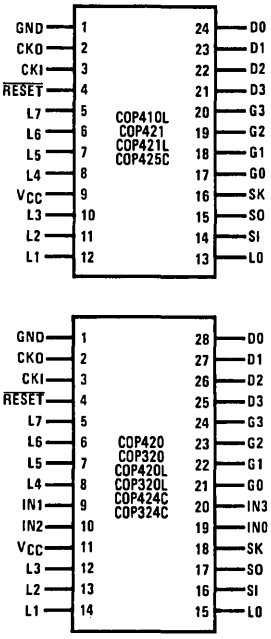
A final note is that all of these networks place a load on the CKO output. If the signal from CKO is needed elsewhere in the system and a circuit similar to one of those discussed in this document is used, it will probably be necessary to buffer the CKO output.

4.0 Conclusion

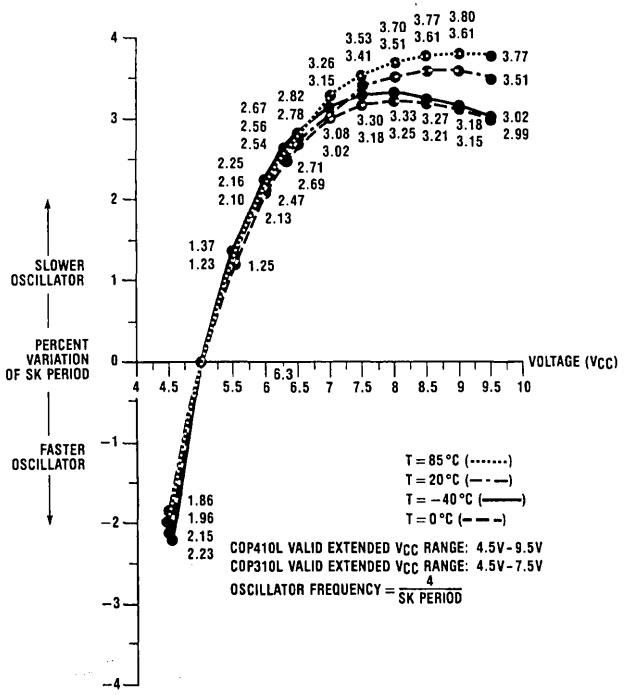
The networks described are generally simple and inexpensive and have all been observed to be functional.

The results obtained provide greater flexibility in the oscillator selection in a COPs system and gives the user some general indication as to what may be expected with the various circuits described.

COP Microcontroller Pinouts



TL/DD/6938-1



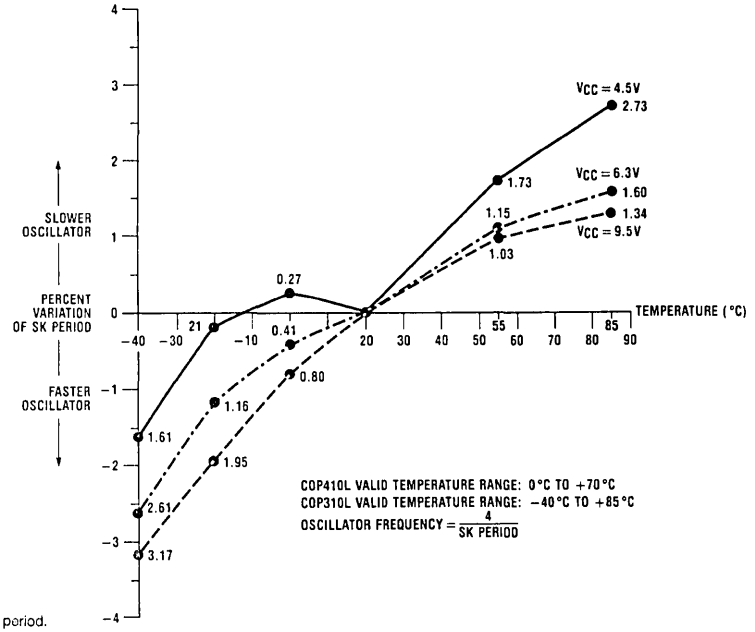
TL/DD/6938-2

Note 1: Base period at V_{CC} = 5.0V.

Note 2: Device variation only. Graph does not include RC variation with temperature.

Note 3: SK period = instruction cycle time.

FIGURE 1. COP310L/COP410L RC Oscillator Variation with V_{CC}

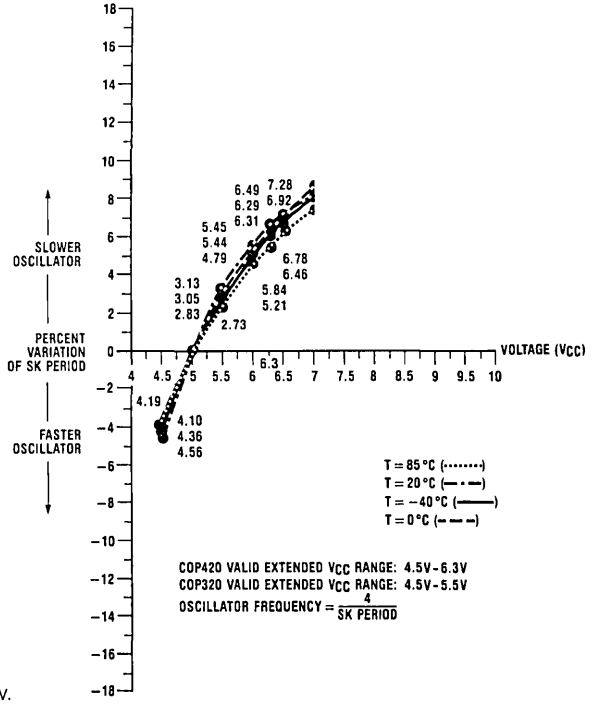


Note 1: 20°C = base period. TL/DD/6938-3

Note 2: Device variation only. Graph does not include RC variation with temperature.

Note 3: SK period = instruction cycle time.

FIGURE 2. COP310L/COP410L RC Oscillator Variation with Temperature

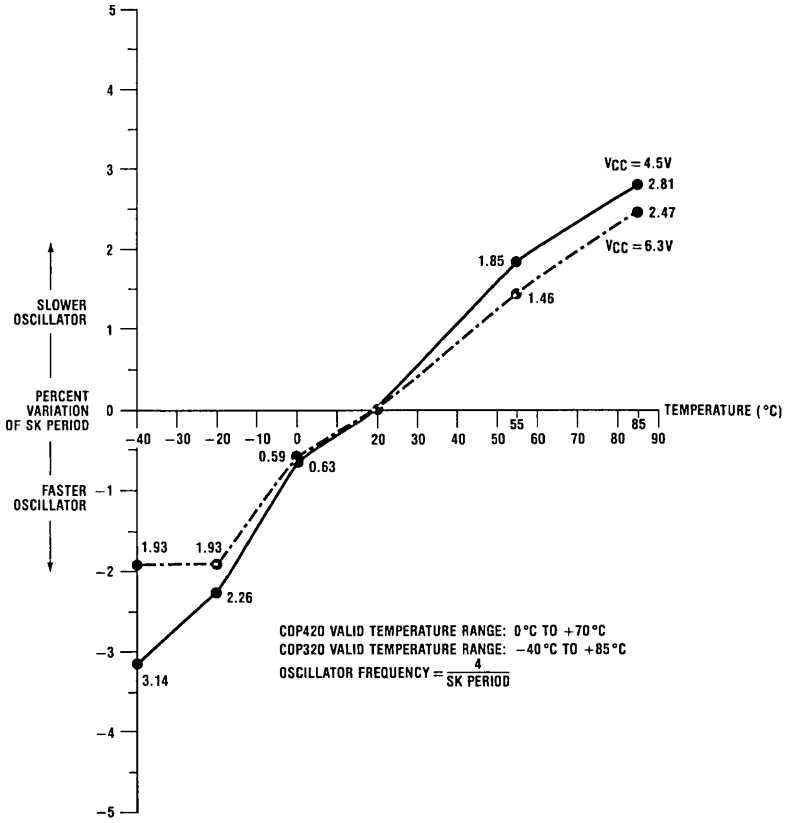


Note 1: Base period at V_{CC} = 5.0V. TL/DD/6938-4

Note 2: Device variation only. Graph does not include RC variation with temperature.

Note 3: SK period = instruction cycle time.

FIGURE 3. COP320/COP420 RC Oscillator Variation with V_{CC}



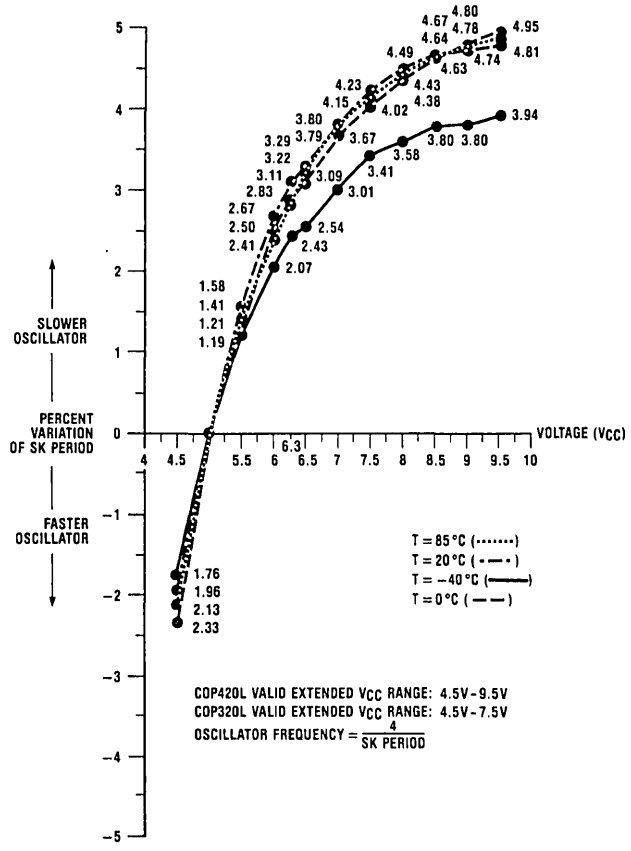
TL/DD/6938–5

Note 1: 20°C = base period.

Note 2: Device variation only. Graph does not include RC variation with temperature.

Note 3: SK period = instruction cycle time.

FIGURE 4. COP320/COP420 RC Oscillator Variation with Temperature

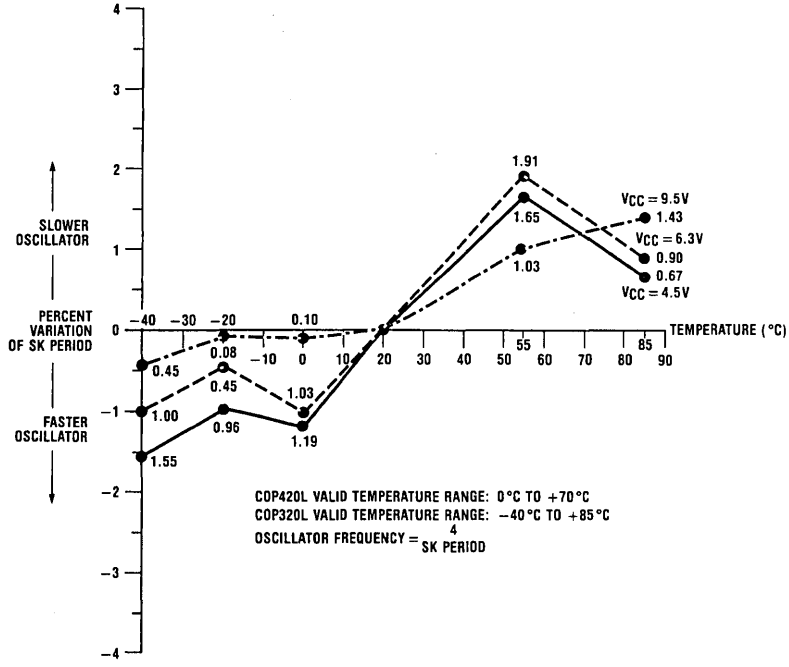


Note 1: Base period at V_{CC} = 5.0V.

Note 2: Device variation only. Graph does not include RC variation with temperature.

Note 3: SK period = instruction cycle time.

FIGURE 5. COP320L/COP420L RC Oscillator Variation with V_{CC}



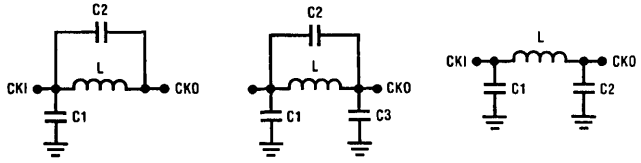
TL/DD/6938–7

Note 1: 20°C = base period.

Note 2: Device variation only. Graph does not include RC variation with temperature.

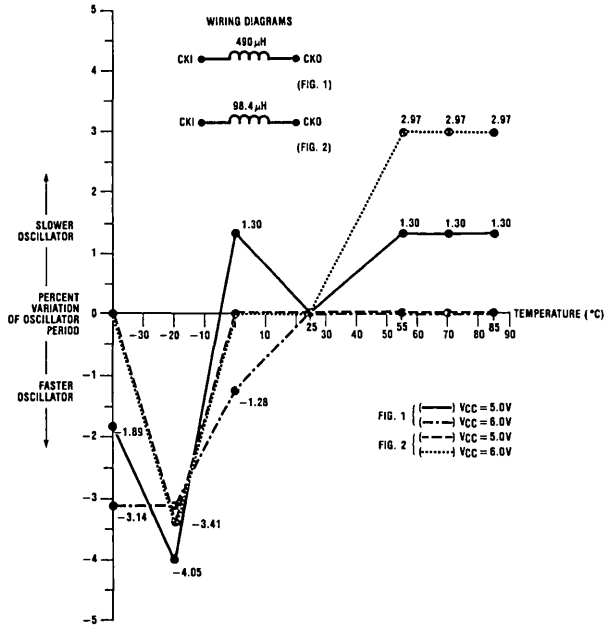
Note 3: SK period = instruction cycle time.

FIGURE 6. COP320L/COP420L RC Oscillator Variation with Temperature



TL/DD/6938–8

FIGURE III.1 **FIGURE III.2** **FIGURE III.3**

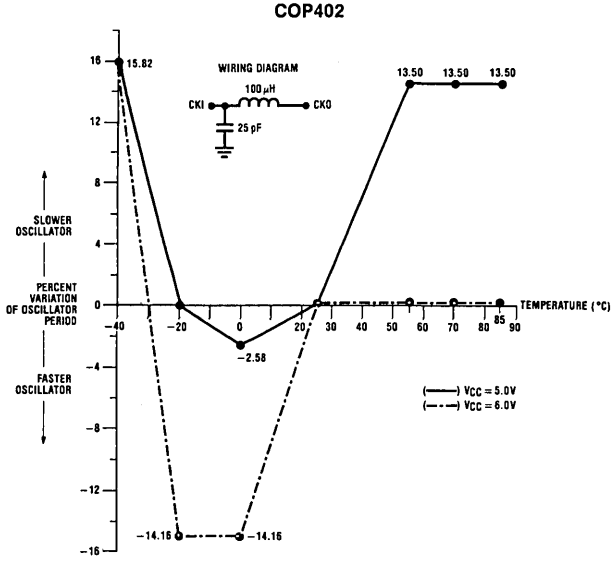


TL/DD/6938-9

Note 1: 25°C = base period.

Note 2: Device variation only. Graph does not include "L" variation with temperature.

FIGURE 7



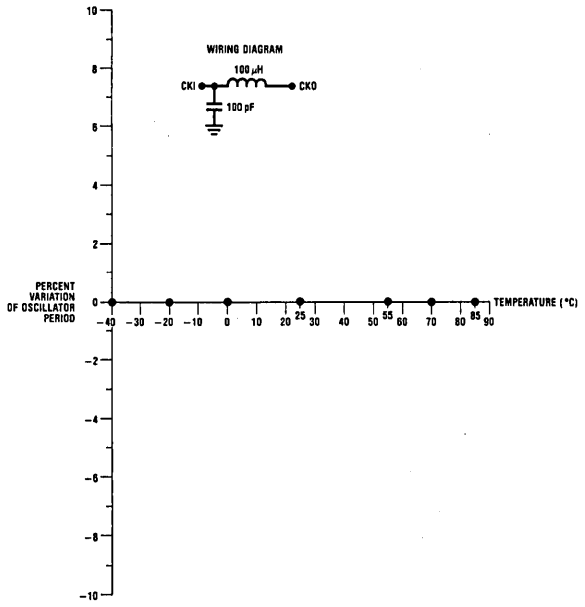
TL/DD/6938-10

Note 1: 25°C = base period.

Note 2: Device variation only. Graph does not include LC variation with temperature.

FIGURE 8

COP420



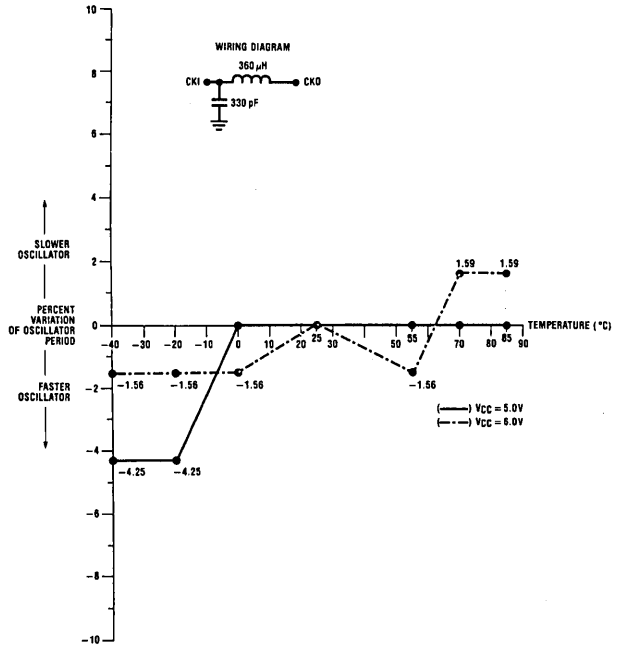
TL/DD/6938–11

Note 1: 25°C = base period.

Note 2: Device variation only. Graph does not include LC variation with temperature.
No measurable variation over temperature.

FIGURE 9

COP420



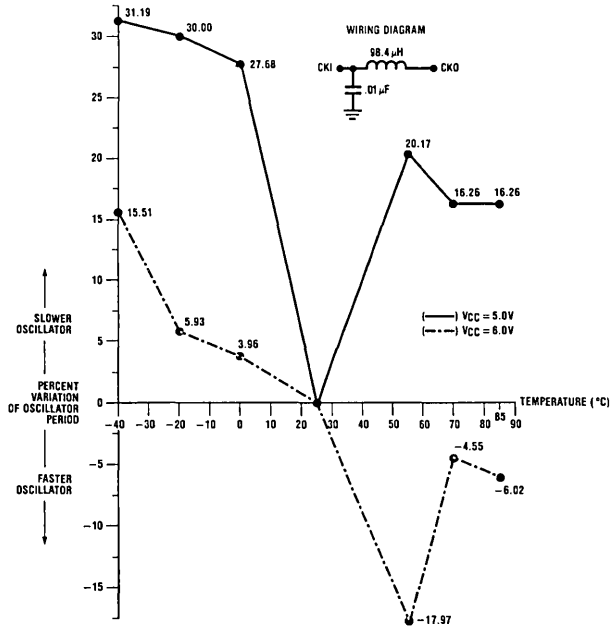
TL/DD/6938–12

Note 1: 25°C = base period.

Note 2: Device variation only. Graph does not include LC variation with temperature.

FIGURE 10

COP402



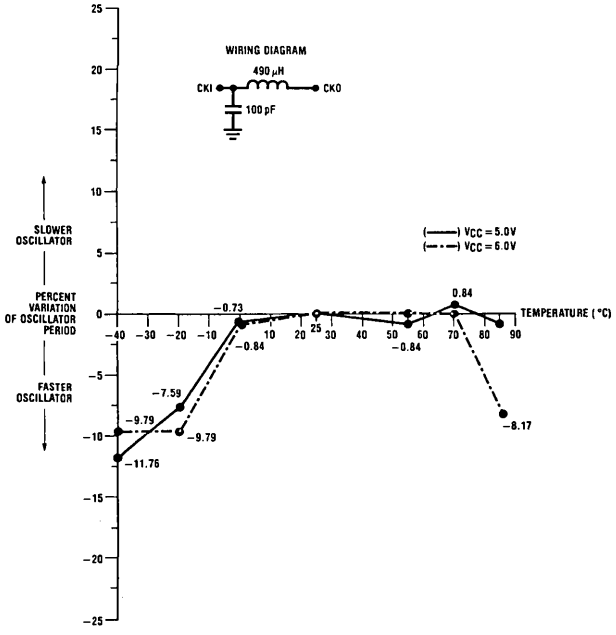
TL/DD/6938–13

Note 1: 25°C = base period.

Note 2: Device variation only. Graph does not include LC variation with temperature.

FIGURE 11

COP402



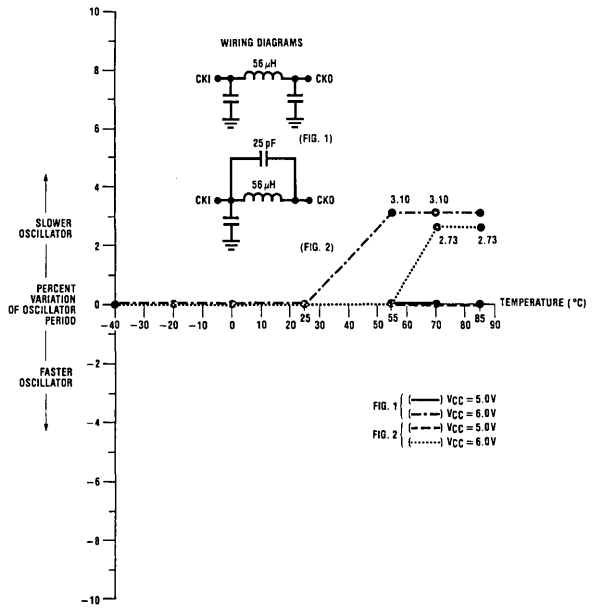
TL/DD/6938–14

Note 1: 25°C = base period.

Note 2: Device variation only. Graph does not include LC variation with temperature.

FIGURE 12

COP402



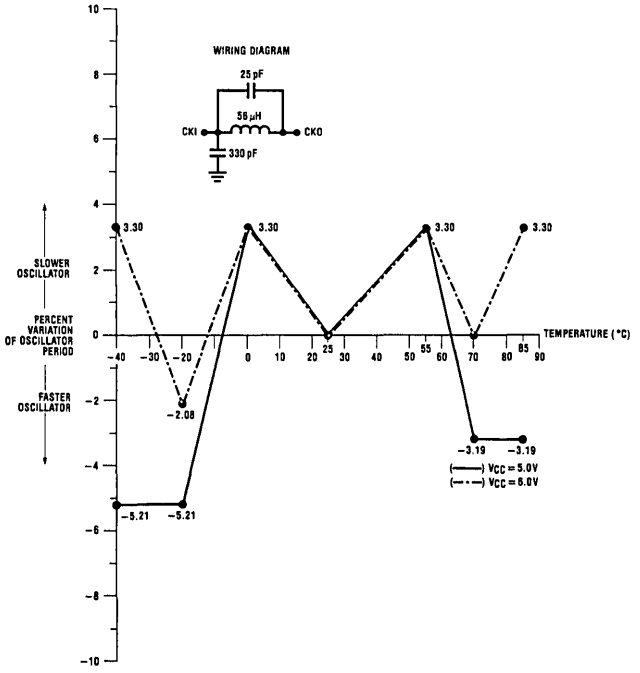
TL/DD/6938–15

Note 1: 25°C = base period.

Note 2: Device variation only. Graph does not include LC variation with temperature.

FIGURE 13

COP402



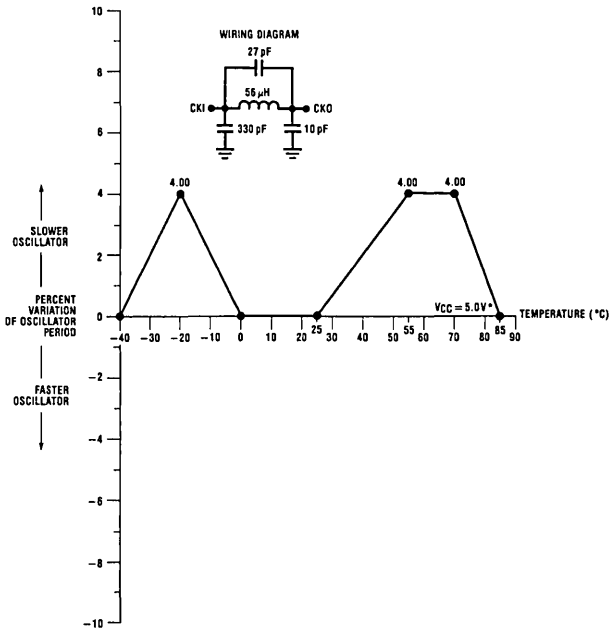
TL/DD/6938–16

Note 1: 25°C = base period.

Note 2: Device variation only. Graph does not include LC variation with temperature.

FIGURE 14

COP402



TL/DD/6938–17

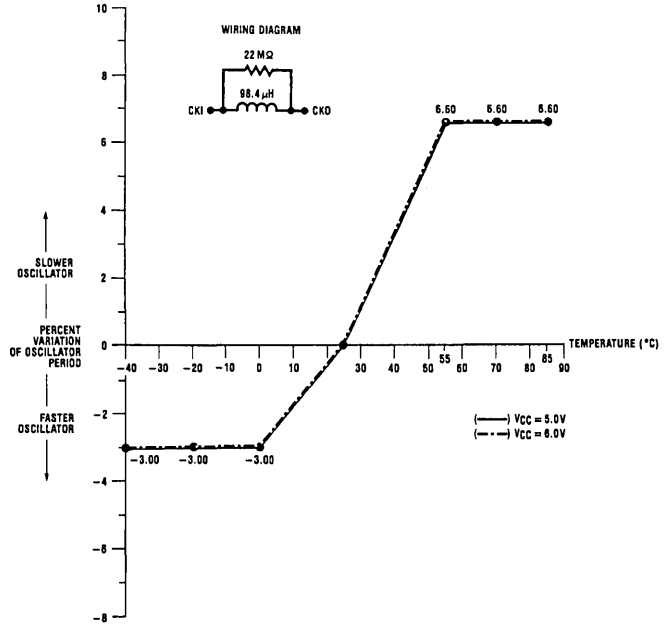
Note 1: 25°C = base period.

Note 2: Device variation only. Graph does not include LC variation with temperature.

*No variation at 6V.

FIGURE 15

COP402



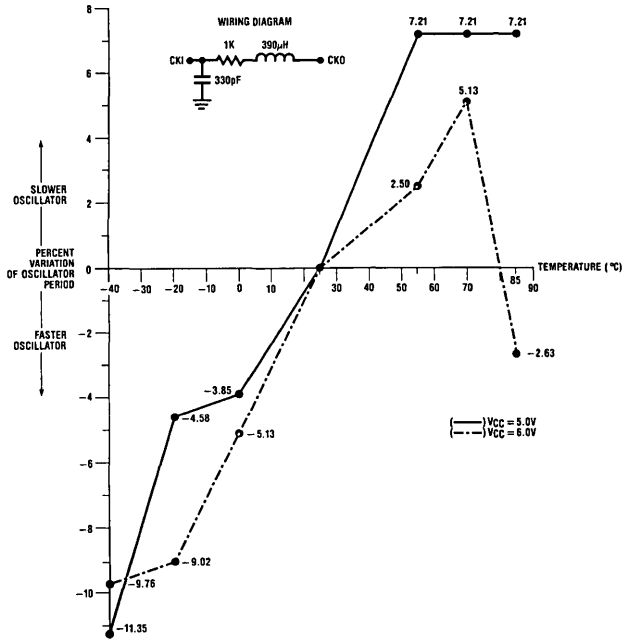
TL/DD/6938–18

Note 1: 25°C = base period.

Note 2: Device variation only. Graph does not include RL variation with temperature.

FIGURE 16

COP402



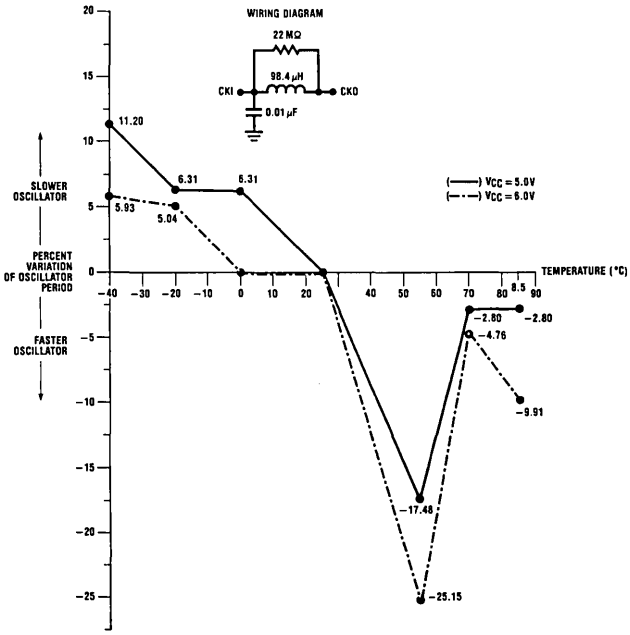
TL/DD/6938–19

Note 1: 25°C = base period.

Note 2: Device variation only. Graph does not include RLC variation with temperature.

FIGURE 17

COP402



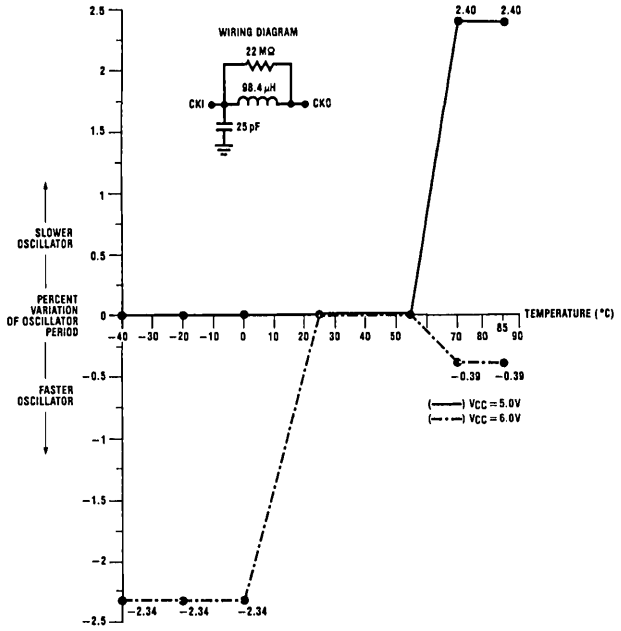
TL/DD/6938–20

Note 1: 25°C = base period.

Note 2: Device variation only. Graph does not include RLC variation with temperature.

FIGURE 18

COP402



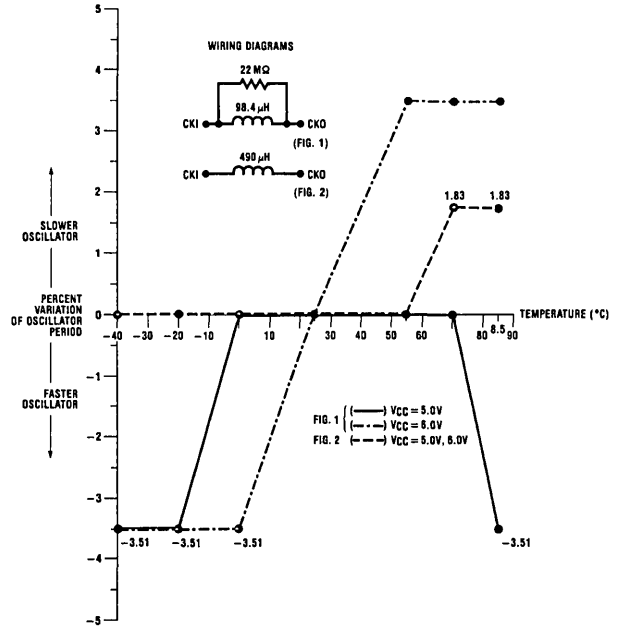
TL/DD/6938-21

Note 1: 25°C = base period.

Note 2: Device variation only. Graph does not include RLC variation with temperature.

FIGURE 19

COP402



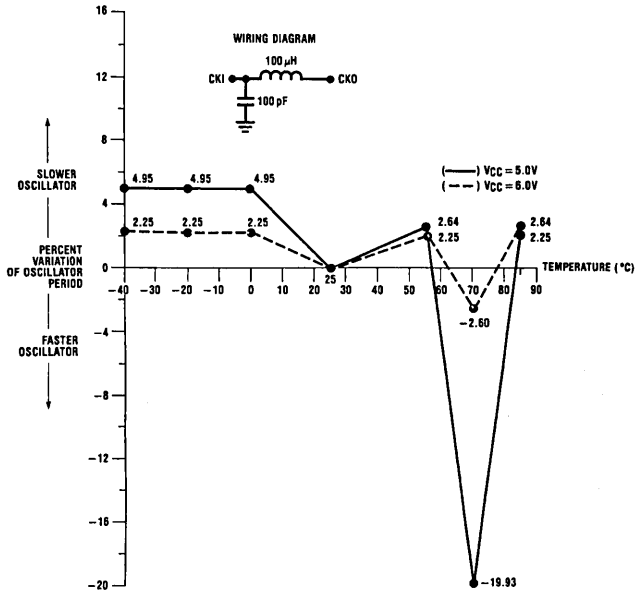
TL/DD/6938-22

Note 1: 25°C = base period.

Note 2: RL in oven with COP402.

FIGURE 20

COP420



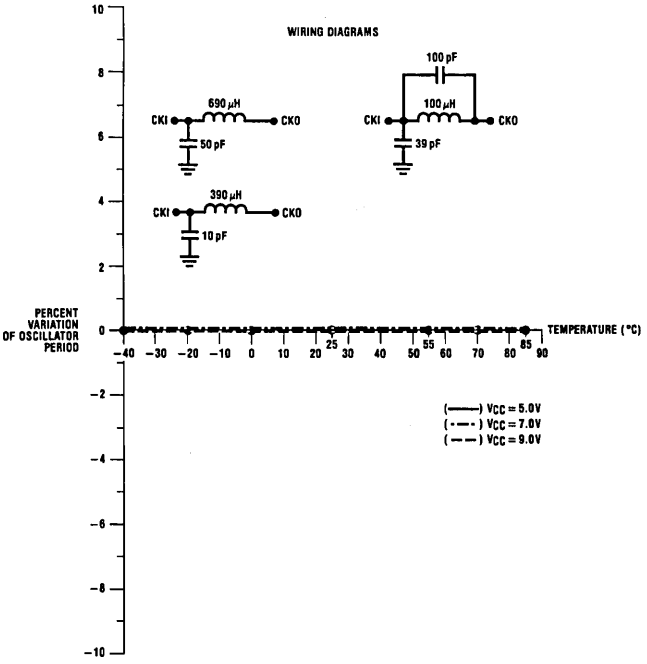
TL/DD/6938–23

Note 1: 25°C = base period.

Note 2: LC in oven with COP402.

FIGURE 21

COP420L



TL/DD/6938–26

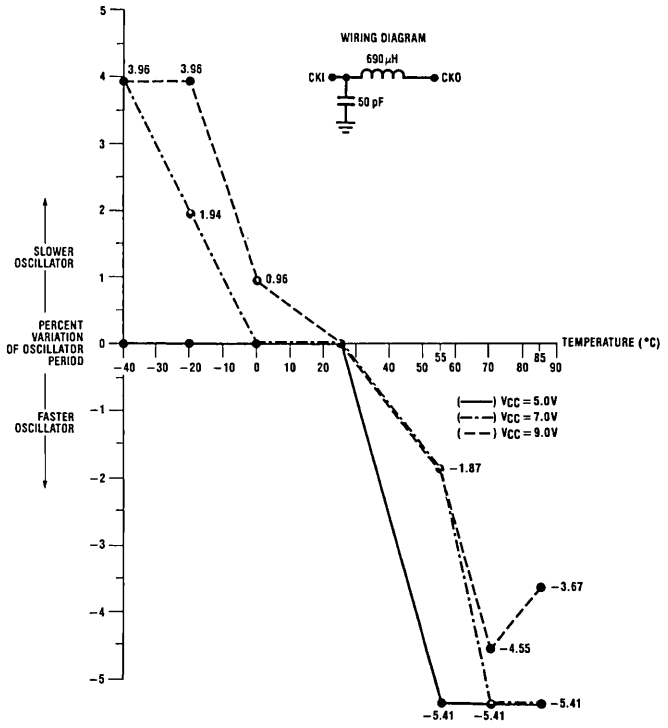
Note 1: No measurable variation for all three circuits above.

Note 2: 25°C = base period.

Note 3: Device variation only. Graph does not include LC variation with temperature.

FIGURE 22

COP420L



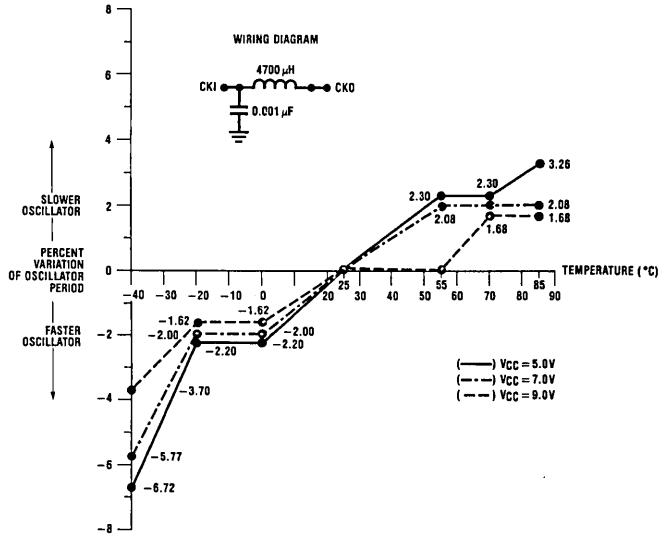
TL/DD/6938-28

Note 1: 25°C = base period.

Note 2: LC in oven with COP420L.

FIGURE 23

COP410L



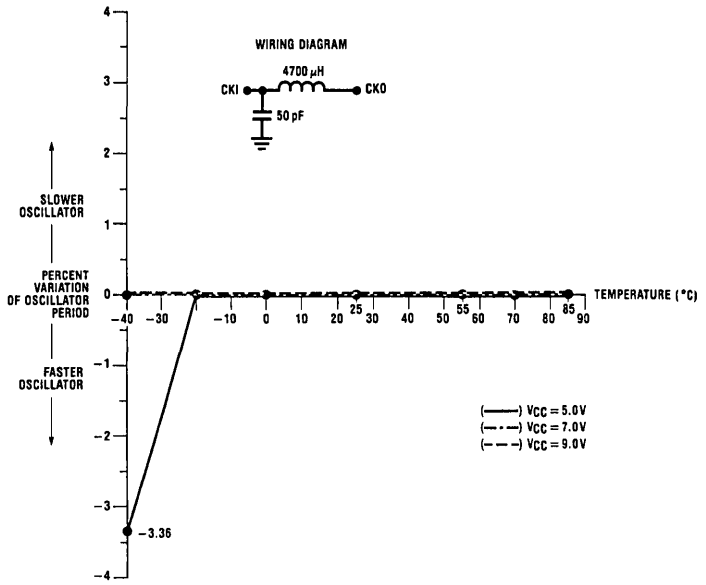
TL/DD/6938-29

Note 1: 25°C = base period.

Note 2: Device variation only. Graph does not include LC variation with temperature.

FIGURE 24

COP410L



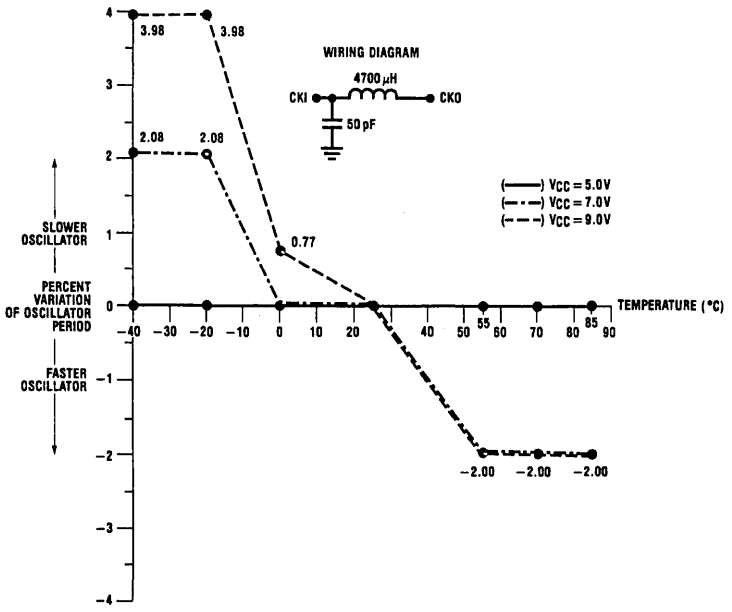
TL/DD/6938–30

Note 1: 25°C = base period.

Note 2: Device variation only. Graph does not include LC variation with temperature.

FIGURE 25

COP410L



TL/DD/6938–32

Note 1: 25°C = base period.

Note 2: LC in oven with COP410L.

FIGURE 26

Triac Control Using the COP400 Microcontroller Family

National Semiconductor Corp.
COP Note 6

Table of Contents

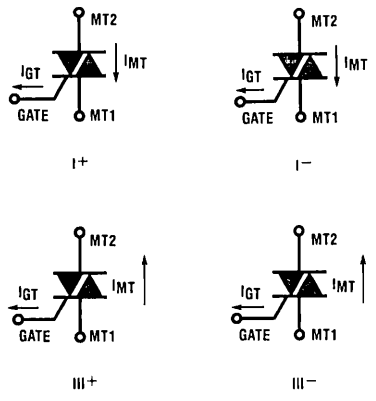
1.0 Triac Control

The COP400 single-chip controller family members provide computational ability and speed which is more than adequate to intelligently manage power control. These controllers provide digital control while low cost and short turnaround enhance COPS™ desirability. The COPS controllers are capable of 4 μs cycle times which can provide more than adequate computational ability when controlling 60 Hz line voltage. Input and output options available on the COPS devices can contour the device to apply in many electrical situations. A more detailed description of COPS qualifications is available in the COP400 data sheets.

The COPS controller family may be utilized to manage power in many ways. This paper is devoted to the investigation of low cost triac interfaces with the COP400 family microcontroller and software techniques for power control applications.

1.1 BASIC TRIAC OPERATION

A triac is basically a bidirectional switch which can be used to control AC power. In the high-impedance state, the triac blocks the principal voltage across the main terminals. By pulsing the gate or applying a steady state gate signal, the triac may be triggered into a low impedance state where conduction across the main terminals will occur. The gate signal polarity need not follow the main terminal polarity; however, this does affect the gate current requirements. Gate current requirements vary depending on the direction of the main terminal current and the gate current. The four trigger modes are illustrated in *Figure 1*.

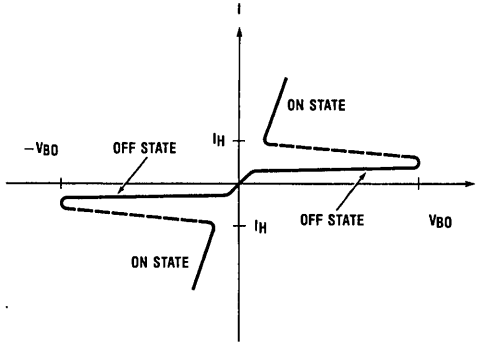


TL/DD/6939–1

FIGURE 1. Gate Trigger Modes. Polarities Referenced to Main Terminal 1.

The breakover voltage (V_{BO}) is specified with the gate current (I_{GT}) equal to zero. By increasing the gate current supplied to the triac, V_{BO} can be reduced to cause the triac to go into the conduction or on state. Once the triac has entered the on state the gate signal need not be present to sustain conduction. The triac will turn itself off when the main terminal current falls below the minimum holding current required to sustain conduction (I_H).

A typical current and voltage characteristic curve is given in *Figure 2*. As can be seen, when the gate voltage and the main terminal 2 (MT2) voltages are positive with respect to MT1 the triac will operate in quandrant 1. In this case the trigger circuit sources current to the triac (I+ MODE).



TL/DD/6939–2

FIGURE 2. Voltage-Current Characteristics

After conduction occurs the main terminal current is independent of the gate current; however, due to the structure of the triac the gate trigger current is dependent on the direction of the main terminal current. The gate current requirements vary from mode to mode. In general, a triac is more easily triggered when the gate current is in the same direction as the main terminal current. This can be illustrated in the situation where there is not sufficient gate drive to cause conduction when MT2 is both positive and negative. In this case the triac may act as a single direction SCR and conduction occurs in only one direction. The trigger circuit must be designed to provide trigger currents for the worst case trigger situation. Another reason ample trigger current must be supplied is to prevent localized heating within the pellet and speed up turn-on time. If the triac is barely triggered only a small portion of the junction will begin to conduct, thus causing localized heating and slower turn-on. If an insufficient gate pulse is applied damage to the triac may result.

1.2 TRIGGERING

Gate triggering signals should exceed the minimum rated trigger requirements as specified by the manufacturer. This is essential to guarantee rapid turn-on time and consistent operation from device to device.

Triac turn-on time is primarily dependent on the magnitude of the applied gate signal. To obtain decreased turn-on times a sufficiently large gate signal should be applied. Faster turn-on time eliminates localized heat spots within the pellet structure and increases triac dependability.

Digital logic circuits, without large buffers, may not have the drive capabilities to efficiently turn on a triac. To insure proper operation in all firing situations, external trigger circuitry might become necessary. Also, to prevent noise from disturbing the logic levels, AC/DC isolation or coupling techniques must be utilized. Sensitive gate triacs which require minimal gate input signal and provide a limited amount of main terminal current may be driven directly. This paper will focus on $120V_{AC}$ applications of power control.

1.3 ZERO VOLTAGE DETECTION

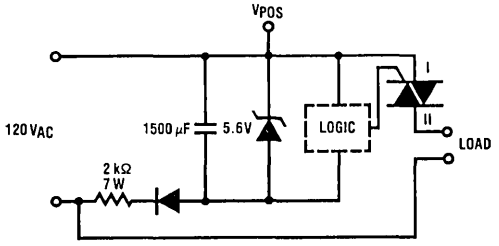
In many applications it is advantageous to switch power at the AC line zero voltage crossing. In doing this, the device being controlled is not subjected to inherent AC transients. By utilizing this technique, greater dependability can be obtained from the switching device and the device being switched. It is also sometimes desirable to reference an event on a cyclic basis corresponding to the AC line frequency. Depending on the load characteristics, switching times need to be chosen carefully to insure optimal performance. Triac controlled AC switching referenced to the AC 60 Hz line frequency enables precise control over the conduction angle at which the triac is fired. This enables the COPS device to control the power output by increasing or decreasing the conduction angle in each half cycle.

A wide variety of zero voltage detection circuits are available in various levels of sophistication. COPS devices, in most cases, can compensate for noisy or semi-accurate ZVD circuits. This compensation is utilized in the form of debounce and delay routines. If a noisy transition occurs near zero volts the COPS device can wait for a valid transition period specified by the maximum amount of noise present. Some software considerations are presented in the software section and are commented upon. The minimal detection circuit is shown in *Figure 9*.

1.4 DIRECT COUPLE

Isolation associated problems can be overcome by means of direct AC coupling. One such method is illustrated in *Figure 3*. This circuit incorporates a half-wave rectifier in conjunction with a filter capacitor to provide the logic power supply. The positive half-cycle is allowed to drop across the zener diode and be filtered by the capacitor. This creates a low cost line interface; however, only a limited supply current is available. In order to control the current capabilities of this circuit the series resistor must be modified. However, as more current is required, the power that must be dissipated in the series resistor increases. This increases the power dissipation requirements of the series resistor and the system cost. For applications which require large current sources an alternative method is advisable. In order to assure consistent operation, power supply ripple must be mini-

mized. COPS devices can be operated over a relatively wide power supply range. However, excessive ripple may cause an inadvertent reset operation of the device.

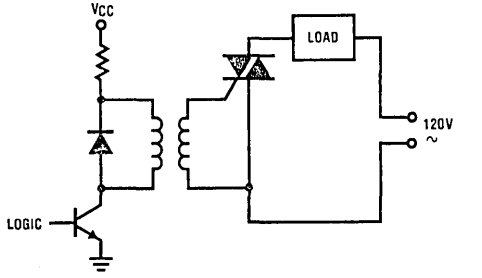


TL/DD/6939-3

FIGURE 3. AC Direct Couple

1.5 PULSE TRANSFORMER INTERFACE

Digital logic control of triacs is easily accomplished by triggering through pulse transformers or optical coupling. The energy step-up gained by using a pulse transformer should provide a more than adequate gate trigger signal. This complies with manufacturers' suggested gate signal requirements. Pulse transformers also provide AC/DC isolation necessary in control logic interfaces. Minimal circuit interface to the pulse transformer is required as shown in *Figure 4*. Optical coupling circuits provide isolation, and in some cases adequate gate drive capabilities.



TL/DD/6939-4

FIGURE 4. Pulse Transformer Interface

A logic controlled pulse is applied to the base of the transistor to switch current through the primary of the pulse transformer. The transformer then transfers the signal to the secondary and causes the triac to fire. The energy transfer that is now available on the secondary is more than adequate to turn on the triac in any of its operating modes. When the pulse transformer is switched off a reverse EMF is generated in the primary coil which may cause damage to the transistor. The diode across the primary serves to protect the collector junction of the switching transistor. Another major advantage is AC isolation; the gate of the triac is now completely isolated from the logic portion of the circuit.

1.6 FALSE TURN-ON

When switching an inductive load, voltage spikes may be generated across the main terminals of the triac which have the potential of a non-gated turn-on of the triac. This creates the undesirable situation of limited control of the system. In a system with an inductive load the voltage leads the current by a phase shift corresponding to the amount of inductance in the motor. As the current passes near zero, the voltage is at a non-zero value, offset due to the phase shift. When the principal current through the triac pellet decreases to a value not capable of sustaining conduction the triac will turn off. At this point in time the voltage across the terminals will instantaneously attain a value corresponding to the phase shift caused by the inductive load. The rapid decay of current in the inductor causes an L dI/dT voltage applied across the terminals of the triac. Should this voltage exceed the blocking voltage specified for the triac, a false turn-on will occur.

In order to avoid false turn-on, a snubber network must be added across the terminals to absorb the excess energy generated by this situation. A common form of this network is a simple RC in series across the terminals. In order to select the values of the network it is necessary to determine the peak voltage allowable in the system and the maximum dV/dT stress the triac can withstand. One approach to obtaining the optimal values for R_S and C_S is to model the effective circuit and solve for the triac voltage. The snubber in conjunction with the load can now be modeled as an RLC network. Due to the two storage elements (L motor, C snubber) a second order differential equation is generated. Rather than approach this problem from a computer standpoint it becomes much easier to obtain design curves generated for rapid solution of this problem. These design curves are available in many triac publications. (For instance, see RCA application note AN 4745.)

2.0 Software Techniques

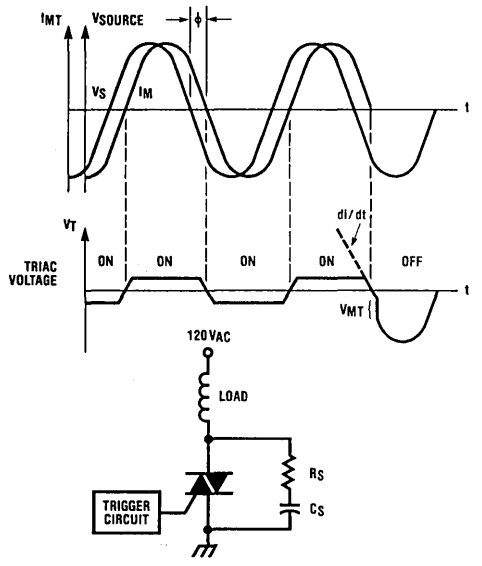
2.1 ZERO VOLTAGE DETECTION

In order to intelligently control triacs on a cyclic basis, an accurate time base must be defined. This may be in the form of an AC, 60 Hz sync pulse generated by a zero voltage detection circuit or a simple real time clock. The COP400 series microcontrollers are suited to accommodate either of these time base schemes while accomplishing auxiliary tasks.

Zero voltage detection is the most useful scheme in AC power control because it affords a real time clock base as well as a reference point in the AC waveform. With this information it is possible to minimize RFI by initiating power-on operations near the AC line voltage zero crossing. It is also possible to fire the triac for only a portion of the cycle, thus utilizing conduction angle manipulation. This is useful in both motor control and light intensity control.

Sophisticated zero voltage detection circuits which are capable of discriminating against noise and switch precisely at zero crossing are not necessary when used in conjunction with a COPS device. COPS software is capable of compensating for noisy or semi-accurate zero voltage detection circuits. This can be accomplished by introducing delays and debounce techniques in the software routines. With a given reference point in the AC waveform it now becomes easy

to divide the waveform to efficiently allocate processing time. These techniques are illustrated in the code listing at the end of this paper.



TL/DD/6939-5

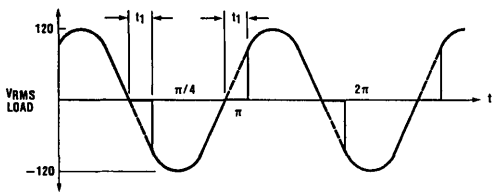
FIGURE 5. Current Lag Caused by Inductive Load, Snubber Circuit

2.2 PROCESSING TIME ALLOCATIONS

Half Cycle Approach

In order to accomplish more than triac timing, dead delay time must be turned into computation time. It appears that the controller is occupied totally by time delays, which leaves a very limited amount of additional control capability. There are, however, many ways to accomplish auxiliary tasks simultaneously.

On each half cycle an initial delay is incorporated to space into the cycle. This dead time may be put to use and very little voltage to the load is sacrificed. For example, if the load is switched on at $\pi/4$ RAD, the maximum applied RMS voltage to the load is $114V_{RMS}$ (assuming $V_{SUPPLY} = 120V_{RMS}$). This is illustrated in the figure below.



TL/DD/6939-6

FIGURE 6. Full Cycle Approach

If a delay of $\pi/4$ RAD (45 degrees) is inserted after each zero crossing detection the RMS voltage to the load can be determined in the following manner:

$$V_{LOAD} = \sqrt{\frac{(120\sqrt{2})^2}{(2)\pi}}(2)\int_{\pi/4}^{\pi} \sin^2(a)\,da$$

$$V_{LOAD} = \sqrt{\frac{(120\sqrt{2})^2}{(2)\pi}}(2)(1.428)$$

$$V_{LOAD} = 114.4\ V_{RMS}$$

$\pi/4$ RAD = 45 degrees @60 Hz t = 2.08 ms

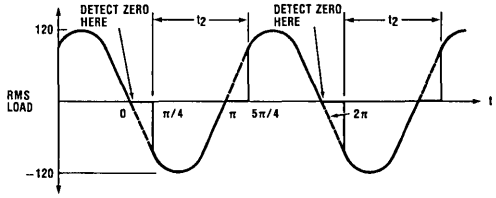
As can be seen the dead time on each half cycle can be 2.08 ms and the load will still see 114.4 V_{RMS} of a V_{SUPPLY} of 120 V_{RMS}. If this approach is implemented the initial delay of 2.08 ms can be used as computation time. The number of instructions which can be executed when operating at 4 μs instruction cycle time is:

2.08 ms/4 μs = 520 instructions

(130 instructions at 16 μs cycle time)

Full Cycle Approach

The methods of half cycle and full cycle triggering are very similar in procedure. The main difference is that all timing is referenced from only one (of the two) zero voltage detection transition in each full AC cycle. For most all applications, when varying the conduction angle it is desirable to fire at the same conduction angle each half cycle to maintain a symmetric applied voltage. In order to accomplish this the triac may be fired twice from one reference point. When applying this technique an 8.33 ms delay must be executed to maintain the symmetric applied voltage. This approach provides the most auxiliary computation time in that the 8.33 ms delay may be turned into computational time. The basic flow for this technique is illustrated below.



TL/DD/6939-7

FIGURE 7. Full Cycle Approach

In the above example the zero crossing pulse is debounced on the one-to-zero transition, thus marking the beginning of a full cycle. Once this transition has been detected, an ini-

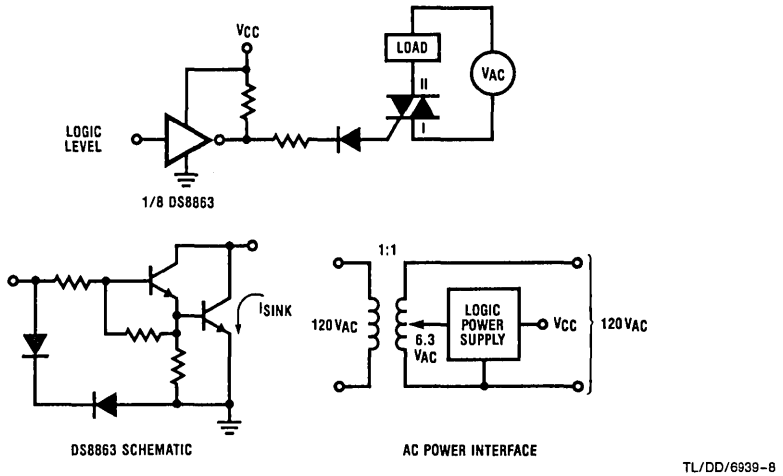


FIGURE 8. Steady State Triggering

tial delay of $\pi/4$ RAD is incorporated and the triac is fired. At this time exactly 8.33 ms is available until the triac need be triggered again. This will provide a symmetric voltage to the load only if the delay is 8.33 ms. During this period the number of instructions which can be executed when operating at 4 μs is:

$$8.33 \text{ ms}/4\ \mu\text{s} = 2082$$

(520 instructions at 16 μs)

An alternative approach may be to take the burden from the COPS device by using peripheral devices such as static display controllers, external latches, etc.

2.3 STEADY STATE TRIGGERING

It is possible to trigger a triac with a steady state logic level. This is accomplished by allowing the triac gate to sink or source current during the desired on-time. When utilizing this method it becomes easier to trigger the triac and leave it on for many cycles without having to execute code to retrigger. This approach is advantageous when the triac must be fired is for relatively long periods and conduction angle firing is not desired, thus more time is available to accomplish auxiliary tasks. A steady state on or off signal and external circuitry can accomplish triac firing and free the processor for other tasks. If it is desired to use a pulse transformer, an external oscillator must be gated to the triac to provide the trigger signal. A pulse train of 10 to 15 kHz is adequate to fire the triac each half cycle. This calls for external components and is relatively costly. If isolation associated problems can be tolerated or overcome (dual power supply transformers, direct AC coupling, etc.), a simple buffer may be utilized in triggering the triac. This method is illustrated in *Figure 8*. The National Semiconductor DS8863 display driver is capable of steady state firing of the triac. National offers many buffers capable of driving several hundred milliamps, which are suitable for driving triacs. On the market today there are many suppliers of sensitive gate triacs which may be triggered directly from a COPS device or in conjunction with a smaller external buffer.

The DS8863 display driver is capable of sinking up to 500 mA, which is adequate to drive a standard triac. In the off state the driver will not sink current. When a logic "1" is applied to the input the device will turn on. Keeping the device off (output "1") will prevent the triac from turning on because the buffer does not have the capability of sourcing current. A series resistor limits the current from the triac gate and the diode isolates the negative spikes from the gate. Since the drive circuit will only sink current in this configuration, the triac will be operating in the I- and III- modes.

3.0 Triac Light Intensity Control Code

The following code is not intended to be a final functional program. In order to utilize this program, modifications must be made to specialize the routines. This is intended to illustrate the method and is void of control code to command a response such as intensify or deintensify. The control is up to the user and full understanding of the program must be attained before modifications can be implemented.

This program is a general purpose light intensifying routine which may be modified to suit light dimmer applications. The delay routines require a 4.469 μs cycle time which can be attained with a 3.578 MHz crystal (CKI/16 option). This program divides the half cycle of a 60 Hz power line into 16 levels. Intensity is varied by increasing or decreasing the conduction angle by firing the triac at various levels. The program will increase the conduction angle to a maximum specified intensity in a fixed amount of time. The time required to intensify to the maximum level is dependent on the number of fire-times per level that is specified (FINO). This code illustrates a half cycle approach and relies on the parameters specified by the programmer in the control selection.

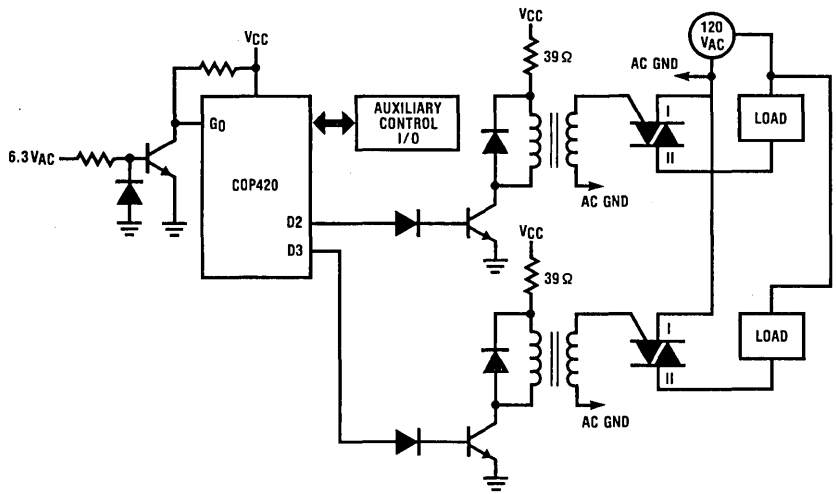
Zero crossings of the 60 Hz line are detected and software debounced to initiate each half cycle; thus the triac is serviced on every half cycle of the power line. A level/sublevel approach is utilized to vary the conduction angle and provide a prolonged intensifying period. The maximum intensity is specified by the "LEVEL" RAM location and time required to get to that level is specified by the "FINO" RAM location.

Once a level has been specified, the remaining time in the half cycle is then divided into sublevels. The sublevels are increased in steps to the maximum level. The "FINO" RAM location contains the number of times that the triac will be fired per sublevel, thus creating the intensity time base. There are 15 valid sublevels and up to 15 fire-times per sublevel. Both these parameters may be increased to provide better resolution and longer intensify periods. To make the triac de-intensity (dim) the sublevels need only to be decremented rather than incremented. If this is done, the conduction angle will start out at the maximum level and dim by means of stepping down the sublevels. When modifying this routine to incorporate more resolution or increased versatility, care must be taken to account for transfer of control instructions to and from the delay routines.

The following is a schematic diagram of the COPS interface to 120V_{AC} lamps. The program will intensify or de-intensify the lamps under program control.

3.1 TRIAC LIGHT INTENSIFY ROUTINE

This program intensifies a light source by varying the conduction angle applied to the load. The maximum level of intensity is stored in "LEVEL," and the time to get to that level is specified by "FIND." Both these parameters may be altered to suit specific applications. To cause the program to de-intensify the light source, the sublevels must be decremented rather than incremented.



TL/DD/6939-9

FIGURE 9. Triac Interface for COPS Program

```
; TRIAC LIGHT INTENSIFY ROUTINE
;
;
; THIS PROGRAM INTENSIFIES A LIGHT SOURCE BY VARYING THE
; CONDUCTION ANGLE APPLIED TO THE LOAD. THE MAX LEVEL
; OF INTENSITY IS STORED IN 'LEVEL' AND THE TIME TO GET TO
; THAT LEVEL IS SPECIFIED BY 'FIND'. BOTH THESE PARAMETERS
; MAY BE ALTERED TO SUIT SPECIFIC APPLICATIONS. TO CAUSE
; THE PROGRAM TO DE-INTENSIFY THE LIGHT SOURCE, THE
; SUBLEVELS MUST BE DECREMENTED RATHER THAN
; INCREMENTED.
;
;
        TEMP1   =1,0    ; TEMPORARY DELAY COUNTER
        FIND    =0,9    ; NUMBER OF FIRE TIMES
        LEVEL   =0,0    ; MAX LEVEL
        SUBLEV  =1,10   ; SUBLEVEL COUNT
        TEMP    =1,11   ; TEMPORARY DELAY COUNTER
;
; HERE THE OPERATING PARAMETERS ARE DEFINED AND LEVEL
; INITIATION IS SPECIFIED
;
        .FORM
        .PAGE   0
        CLRA            ; REQUIRED
CLRAM:  LBI     3,15    ; ROUTINE TO CLEAR ALL RAM
CLR;    CLRA
        XDS
        JP      CLR
        XABR
        AISC    15
        JP      BEGG
        XABR
        JP      CLR
;
; THIS SECTION INITIATES CONTROL ON POWER UP OR RESET
; AND SYNCHRONIZES THE COPS DEVICE TO THE 60 HZ AC LINE
;
BEGG:   OGI     15      ; OUTPUT 15 TO G PORTS TO PULL
                        ; UP ZERO CROSSER INPUT
        LBI     LEVEL   ; SPECIFY MAX LEVEL
        STII    7
        JSR     OUT     ; COPY TO TEMP1
BEG:    SKGBZ   0       ; SYNC UP TO 60 HZ
        JP      HI      ; READY NOW
        JP      BEG     ; WAIT TILL G IS 1
;
; THIS SECTION PROVIDES THE DEBOUNCE FOR THE ZERO
; VOLTAGE DETECTION INPUT AND COMPENSATES FOR THE
; OFFSET OF THE DETECTION CIRCUIT
;
HI:     SKGBZ   0       ; TEST GO FOR ZERO CROSS
        JP      HI      ; HIGH LEVEL
; GETS HERE ON FIRST TRANSITION
        CLRA            ; START OF DEBOUNCE DELAY
        AISC    1
        JP      .-1
; DID A LITTLE DELAY, IS IT STILL 0
        SKGBZ   0       ; TEST FOR 0
        JP      HI      ; FALSE ALARM
; MUST HAVE HAD SOME NOISE GO BACK AND WAIT FOR TRUE ZC
DOIT:   JMP     INT     ; VALID TRANSITION, SERVICE
                        ; TRIAC
LO:     SKGBZ   0       ; DEBOUNCE IN 0 TO 1
        JP      DDD     ; MAY HAVE SOMETHING THERE
        JP      LO      ; NO WAIT HERE FOR A BIT
DDD:    CLRA            ; GOING TO WAIT AND SEE
        AISC    1
        JP      .-1
        SKGBZ   0       ; WELL, DO WE HAVE A CLEAN
                        ; TRANSITION
        JP      DELL    ; YES, GO TO MAIN ROUTINE
        JP      LO      ; FALSE ALARM, TRY AGAIN
DELL:   CLRA            ; DO A DELAY TO COMPENSATE
DEL:    NOP
        NOP             ; FOR NON SYMMETRIC ZC
        NOP
        AISC
        JP      DEL     ; KEEP DELAY GOING
        JP      DOIT    ; GO TO MAIN ROUTINE

        .FORM
        .PAGE   1
;
;
; THIS IS THE MAIN ROUTINE FOR THE INTENSIFY/DE-INTENSIFY
; OPERATIONS. TRANSFER OF CONTROL TO THIS SECTION
; OCCURS AFTER ZERO VOLTAGE CROSSING EACH HALF CYCLE.
; THIS MAKE USE OF TEMP REGISTERS THUS PARAMETERS
; NEED NOT BE REDEFINED FOR EACH OPERATION.
;
;
INT:    CLRA
        ADT             ; DELAY INTO WAVEFORM
        LBI     TEMP    ; USE TEMP REG
        X
        JSRP    PORT    ; DO DELAY
POINT:  LDD     LEVEL   ; POINT TO LEVEL TO INITIATE
                        ; DELAY
                        ; DELAY TO MAX LEVEL
        XAD     TEMP    ; USE TEMP DIGIT TO DELAY
TAMP:   LBI     TEMP
        LD
        AISC    15      ; ARE WE AT THE LEVEL ?
        JP      ATLEV   ; MADE IT TO THE LEVEL
        X               ; NO
        JSRP    DE5     ; DO SERIES OF .5MS TO GET
                        ; THERE
        JP      TAMP    ; KEEP DOING IT
ATLEV:  LDD     SUBLEV  ; AT MAX FIRE LEVEL
        XAD     TEMP    ; INIT FOR SUBLEVEL DELAY
JK:     LBI     TEMP
        LD
        AISC    1       ; AT SUB LEVEL ?
        JP      TRE     ; NO DO DELAY
        JP      SBLEV   ; YES
TRE:    X
        JSRP    SPDL    ; VARIABLE DELAY
        JP      JK
SBLEV:  LBI     FIND
        JSRP    DEC     ; DEC FIRE NUMBER
        AISC    1       ; TEST IF FIND AT 15
MAXLEV: JMP     FIRE    ; NO KEEP FIRING AT THAT LEVEL
        LBI     SUBLEV  ; YES INC SUBLEVEL
        CLRA
        AISC    14      ; IS MAX SUBLEV REACHED
        SKE
        JP      THERE   ; NO INC SUBLEV
        JP      MAXLEV  ; YES FIRE IT
THERE:  JSRP    INC     ; GO TO NEXT SUBLEVEL
        LBI     FIND
        STII    14      ; SET FIRE TIME
        JP      MAXLEV  ; GO FIRE

        .FORM
        .PAGE   2
```

```
; SUBROUTINE PAGE
INC:    CLRA
        AISC    1
        JP      ADEX        ; GO ADD ONE TO DIGIT
DEC:    CLRA                ; 0 TO A
        COMP                ; CREATE A 15
ADEX:   ADD                 ; ADD A TO RAM
        X                   ; PUT BACK (D - 1 IN A NOW)
        RET
DE5:    LBI     0,10        ; DELAY ROUTINE
        CLRA                ; WILL BE REPLACED LATER
        AISC    3
        JP      .-1
        LD
        XIS
        JP      .-5
        RET                 ; DONE DELAY
FIRE:   LBI     0,15        ; PULSE D OUTPUT
        OBD
        NOP
        NOP
        LBI     0,0
        OBD
        SKBGZ   0           ; TEST WHICH DEBOUNCE IS
                            ; NEEDED
        JMP     HI          ; DEBOUNCE ONE TO ZERO
        JMP     LO          ; DEBOUNCE ZERO TO ONE
SPDL:   LBI     TEMP1       ; TEMP1 IS A TEMP REG
PORT:   LD                  ; VALUE IN TEMP1 DICTATES
        AISC    1           ; THE AMOUNT OF DELAY
        JP      FOY
OUT:    LBI     LEVEL       ; ALSO USED TO COPY LEVEL
        LD      1           ; RESTORE LEVEL
        X
        RET
FOY:    X
        JP      PORT
        .END
```

Testing of COPS™ Chips

National Semiconductor Corp.
COP Note 7

Table of Contents

This note will provide some insight into the test mode, the mechanics of testing, and the philosophy of how to implement a test of the COP-400 microcontrollers. Other than the obvious, (verifying that the part meets the specifications), the reason for the test must be considered. Somewhat different criteria may hold, depending on the objective. The manufacturer wafer sort or final test can differ from an incoming inspection at the user's plant, or a field reject test. The first two tests have limited interest as this is not a justification of the testing done on the part during manufacture. Rather, this is a guide for those doing user functional testing.

1.0 INTRODUCTION

Since the introduction of the very first semiconductor devices, testing has been a major problem and expense in their production and use. As the complexity has risen, testing has become a more significant factor. With today's single chip microcontrollers like the COPS devices this is particularly true as one has a complete computer system in a chip. In order to reduce the testing burden, the facilities to ease the testing have been built into the COPS devices. With the test ability built into the device for production test, the user need only follow set procedures to verify the chip at incoming inspection or field test.

2.0 PHILOSOPHY

The basic test philosophy requires that four major areas be exercised. These areas are:

1) Synchronize the device and tester.
2) Test the internal logic and I/O.
3) Test the RAM.
4) Verify the ROM program.

If the devices perform all of these four properly, the device is good. This is a reasonable assumption with a standard device that has a debugged test routine and is ROM programmed. A custom circuit just going into production might not have the accumulated test background. By attacking the problem on a "sum of the parts" approach, one need not do any exhaustive functional test on routine production parts. This will be a major gain where lengthy time consuming or time dependent routines are involved. If one attempts to do a functional test of the chip, a sequence that is unique to the application is needed. Thus, a test program must be written and debugged for each ROM pattern. Further, a test box/board must be designed, built, debugged, documented, and maintained for each one. If testing has been considered from the beginning, the chip will have built-in capabilities to exercise the various parts of it. The different functional parts and instructions are tested to verify proper operation at the voltage and frequency limits.

3.0 BUILT-IN TEST FEATURES

The first step in testing the COP400 devices is to understand the built-in test control features. This will involve the SI/O and the L lines. The SO pin has been designed to be the control node for testing. The pin will normally be in an active low state and when forced high externally, places the chip in the test mode. It should be noted that this output can sink considerable current and one should not force the pin to the V_{CC} rail. By limiting the voltage to the 2.0/3.0V range one can not damage the device where the application of a higher voltage could. When forced into the test mode the SI pin controls the sub mode of the chip. With SI high the data placed on the L port is used as an instruction. When SI is low (and the L output is enabled) the contents of the ROM will be dumped out through the L port. Certain other internal functions have been implemented to allow these modes but these are not part of the basic operation. Included in this category is the activation of the skip signal to prevent the program counter from jumping out of sequence by executing a program control instruction.

3.1 Sync Between Tester and DUT

In order to be able to test a COPS chip, the tester must be in sync with the device under test (DUT). By using an external oscillator the two may be run at the same frequency. This is true regardless of the option or type of oscillator chosen for the chip. Even the RC configuration may be overridden with an external signal that meets the level requirements. In addition to running at the same frequency, the chip and tester must be in sync on a bit basis. See *Figure 1.* The supportive features mentioned above include the condition of the SK signal being a bit (instruction) clock until stopped by software in the program. Hence, one can start the tests based on an edge change of SK. It is important that this be accurate because all data I/O changes will be relative to the SK timing (see the appropriate device data sheet).

It should also be noted that the oscillator frequency is programmed to a rate of 4-32 higher than SK. If one is building a test fixture for more than one device, some method must be available to enter this number. If one is testing a COP420 or COP421 near its upper limit it would be wise to do the SK sync operation at a lower rate and then increase the input frequency. This is desirable because the phase relationship is close to TTL propagation delays at the upper limit. Implementation of the area could be a preset counter that is gated on after a zero to one transition is seen on SK. Continual comparison could be made but once in sync, there should not be any need for the comparison as they should remain in sync.

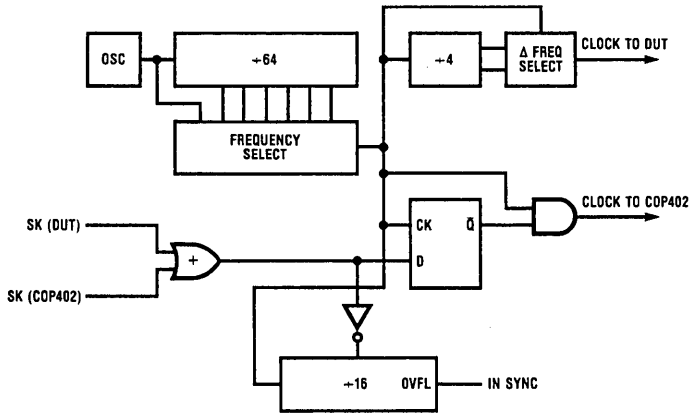
The basic use of this "sync counter" is to derive the proper timing for loading data and instructions into the chip and verify the outputs. The COP402 data sheet should be used as a guide for these times, modified properly for the L and C parts. For those designing testers, it is suggested that one not attempt to test worse case timing changes as these could be very difficult to implement. Like other parametric tests these should in general be left to the professional test equipment.

3.2 Internal Logic Test

With the device and the tester in sync, actual testing may begin. See the sequence control circuit of *Figure 2.* To place the chip into the test mode the SO output is pulled to a one level (between 2.0 and 3.0 volts). It should be pulled with a circuit that will limit the upper voltage to 3V as this output can have a significant current sink capability. On power up (or after reset) the SO line is set to a zero by the internal logic. An internal sense line will detect the forced condition and provide test control. A delay of 10 ms should be taken after power-up to allow the power on reset circuit to time out before instructions can be executed. If the reset pin is activated in mid-program for some reason, several instructions cycle times should be ignored to insure complete operation.

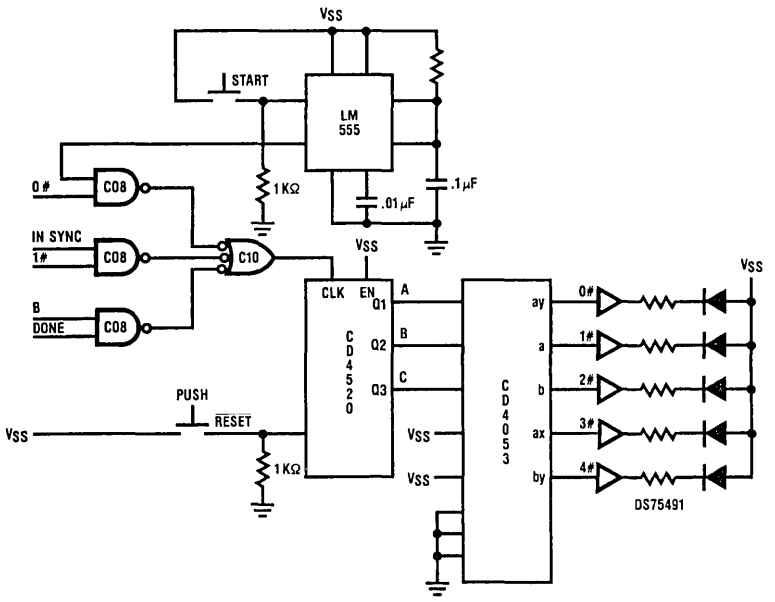
The tester should at this point force instructions into the L port. These instructions will be executed as if they were from the ROM. The sequence of the instructions is not particularly critical. Table I gives an example sequence. The main steps are to be able to detect an output change (OGI) early to verify connection/operation. It is much better to find a problem before going through the steps of loading RAM and then finding that the chip doesn't work. All instructions should be exercised although certain ones should be postponed. Enabling the Q register to the L port is an example. This would interfere with the insertion of instructions on the L port. Another problem is the SO test which could be set up with an XAS and then released from the test mode to check proper data output.

Certain commands will require more effort than others. To check the program counter during JMP's and sub-routine operation will require that known info at the new address be available. One should execute a JSRP at some known address and release the test mode to see that the operation in the subroutine (e.g., SC) is done and that a return is made to N + 1. At this point test mode can be re-established to continue the test. The main point to remember is to provide a positive indication of the success of that specific test.



TL/DD/6940–1

FIGURE 1. Tester Clock Generation and Synchronization Circuit



TL/DD/6940–2

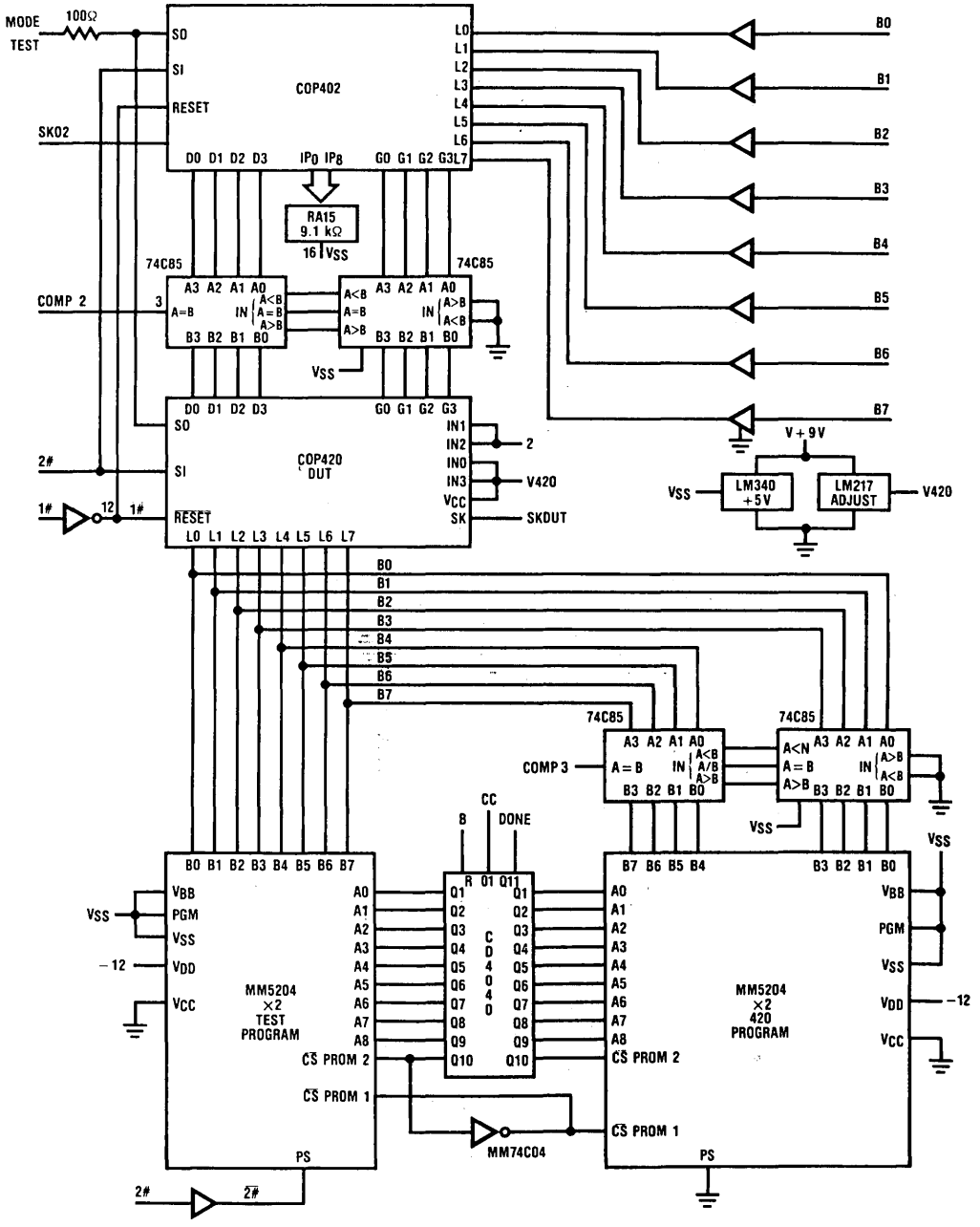
FIGURE 2. Tester Mode Sequencer

3.3 RAM Test

The verification of RAM is a part of the internal logic test, but is treated separately here. One must check both the RAM and its address register to find all faults. An example of this testing would be to load RAM with a string of STII commands. By then going back and reading this data to the outside (through an OMG instruction in a loop) the tester could verify both RAM and address were functional. One could then load RAM with all 6's and 9's (or 5's and 10's) sequentially to insure that all bits were functional and adjacent bits not shorted. Other similar tests could be run at the discretion of the user to do further testing. All of these tests would utilize the output of data via the G ports to validate the data. See the comparator circuit *Figure 3.*

3.4 ROM Dump

Successful operation of the internal logic tests and RAM will lead to the final test phase, ROM comparison. In order to check the ROM contents, the ROM dump mode must be entered. One should force a JMP to an address near the end of the ROM space (3FF for a 420 chip, 1FF for a 410). A desirable point might be 3FA. The program counter will step ahead on each instruction cycle unless a program control is executed. The next step is to load the Q register with a non-conflicting value so that the enabling of the L outputs will not destroy the second byte of the LEI instruction as control is passed into the ROM dump mode. After going to this address, one should execute an enable of the L lines to the output port (LEI 4). Having done this the external buffers should be disabled and the SI pin taken low. This will allow data out and remove potential level conflicts. By letting the PC step ahead to address zero one can then begin the byte by byte comparison of data. In this mode the controller is not executing the code because the skip line is enabled throughout the sequence. By halting a counter on a failure, one could determine the questionable address.



TL/ DD/6940–3

FIGURE 3. Functional Logic and RAM Comparison Circuit

TABLE I. Typical Test Sequence

INSTRUCTION	RESULT	COMMENTS
NOP	NO CHANGE	CHECK NOP & ALLOW TRANSIENT
		CYCLE FOR MODE
OGI 9	G(0 > 9)	NOT ON 410L/411L
OGI 6	G(9 > 6)	REVERSE ALL G STATES
STII 8		SET UP 0,0 FOR FUTURE
LBI 3,13		B TO NEW POSITION (3, 13)
OBD	D(0 > 13	CHECK D
CLRA		MAKE SURE A = 0
XABR		3 > A; 0 > Br
CAB		MOVE 3 to Bd
OBD	D(13 > 3)	CHECK XABR CAB & D CHANGE
CLRA		I
AISC 2		IFORCE A > 2
CAB		2 > Bd
OBD	D(3 > 2)	VERIFY 2 FROM A > Bd
STII 7		7 > 0.2 & Bd > 3
OBD	D(2 > 3)	STII INCREMENTS Bd
CAB		SEE THAT A STILL THE SAME
OMG	G(6 > 7)	OMB & RAM CHECK
CLRA		
CAB		B(0,0)
OMG	G(7 > 8)	TIE IN RAM, A & G OPERATION
SMB 0		SMB INST. CHECK
OMG	G(8 > 9)	:
SMB 1		:
OMG	G(9 > 11)	:
RMB 0		:
RMB 3		:
X		:0 > 0,0;2 > A
CAB		A = 2 > B
OMG	G(11 > 7)	OUTPUT M(0,2)
LD 1		M(0,2) > A; B > 1,2
XAD 0,0		A(7) < -> M(0,0) 2
AISC 15		AISC CHECK; A = 1
LDD 0,0		CHECK SKIP OF 2 BYTE INST.
X		STORE 1
OMB	G(7 > 1)	VERIFY
LD 0		COPY1,2 BACK TO A
ADT		ADD TEN
XDS		LEAVE 11 IN 1,2;GO 1, 1 WITH 1
XDS		LEAVE 1 IN 1,1;GO 1,0 W ?
OBD	D(2 > 0)	CHECK Bd MOVEMENT
STII 5		5 > 1,0;Bd TO 1,1
CBA		CHECK B > A
AISC 3		AISC CHECK 4 >A

INSTRUCTION	RESULT	COMMENTS
XDS		1 > A; 4 > 1,1
OMG	G(1 > 5)	FROM 1,0
XDS		5 > A; 1 > 1,0; Bd < 15 SKIP
LDD 0,0		SKIPPED !
OBD	D(0 > 15)	
AISC 4		9 > A
X		9 > 15
OMG	G(5> 9)	
CLRA		
COMP		ONES TO A
XOR		FLIP MEMORY
XIS		6 > 1,15; 9 > A; Bd > 1,0
LDD 0,0		SKIP
SKE		
LB 1,2		SKIP 2 WORD LBI (NOT IN 410)
OBD	D(15 > 0)	VERIFY WORD
SKE		11 NOT = 9
LBI 1,0		BACK TO 1,0
SMB 2		:
SKE		:
RMB 2		:
SKE		:CHECK BIT
SMB 3		:MANIPULATIONS
SKE		:
LDD 0,0		:
X 3		Bd > 2,0
XAD 1,1		9 > 1,1; 4 > A
XIS 1		4 > 2,0; Bd > 3,1
ING		INPUT G PORT
X		STORE

INSTRUCTION	RESULT	COMMENTS
CLRA		
ASC		CHECK ADD WITH CARRY
SC		CHECK SET CARRY
SKC		CHECK SKIP ON CARRY
LDD 0,0		
X		STORE A
OMG	G = 9	NO CHANGE
CLRA		
ASC		
X		
OMG	G(9 > 10)	CARRY ADDS ONE TO MEMORY
CAMQ		STORE A & M IN Q; 10,9
XDS		9 > 3,1; 10 > A; Bd > 3,0
X		STORE 9 IN 3,0
OMG	G(10 > 9)	
LD 2		9 > A; Bd > 1,0

INSTRUCTION	RESULT	COMMENTS
OMG	G(9 > 1)	
LD 3		1 > A; Bd > 2,0
OMG	G(1 > 2)	
ADD		ADD WITHOUT CARRY
X		STORE 3 IN 2,0
SC		
LDD 0,0		7 > A
CASC		CHECK CASC
SKC		
X		STORE 12
OMG	G(2 > 12)	
CLRA		:
AISC 3		:
X		:
SC		:CHECK
SKC		:SKC/SC
X		:
OMG	G(12 > 3)	
RC		:
SKC		:CHECK
X		:RC
OMG	G(3 > 12)	:
LBI 0,0		:CHECK
LBI 1,15		;SEQUENTIAL LBI'S
LBI 2,7		ALSO SKIPPED (LBI 2,7 NOT IN 410)
OMG	G(2 > 7)	
CQMA		LOAD CONSTANTS FROM Q
OMG	G(7 > 9)	CHECK
X		:
OMG	G(9 > 10)	:
LEI 1		
XAS		STORE A - > S (9)
CLRA		
AISC 7		:
SKGBZ 0		:
X		:CHECK
OMG		:
SKGBZ 1		:
X		;G BIT
OMG	G (10 > 7)	:
SKGBZ 2		:
X		:
OMG	G(7 > 10)	:TESTS
SKGBZ 3		:
X		:
OMG	G(10 > 7)	:

INSTRUCTION	RESULT	COMMENTS
SKGZ		
X		:CHECK
OMG	G(7 > 10)	:
OGI 0	G(10 > 0)	:G TEST
SKGZ		:
X		:
OMG	G(0 > 10)	:
SKMBZ 0		
X		CHECK MEMORY BIT TESTS
OMG		NO CHANGE
SKMBZ 1		

TABLE I. Typical Test Sequence (Continued)

INSTRUCTION	RESULT	COMMENTS
X		
OMG	G(10 > 7)	NO SKIP
SKMBZ 2		
X		WON'T SKIP
OMG	G(7 > 10)	
INIL		SEE THAT L LATCHES RESET
ININ		ASSUME G − > I
SKE		
X1		Br > 1
OMG		SHOULD BE EQUAL
INIL		:
X		:
SKMBZ 3		:
OBD	D(15 > 0)	:INIL TEST
OGI 1		:
LBI 3,11		:
OGI 0		:
INIL		:
X		:
SKMBZ 0		:
OBD	D(0 > 11)	:
NOP		
XAS		:
X		:XAS TEST
OMG	G(10 > 9)	:

INSTRUCTION	RESULT	COMMENTS
LBI 0,0		LOAD RAM WITH
STII 7		CONSTANTS USING
STII 14		STII
STII 5		
STII 12		
STII 3		
STII 10		
STII 1		
STII 8		
STII 15		
STII 6		
STII 13		
STII 4		
STII 11		
STII 2		
STII 9		
STII 0		
LBI 1,0		
STII 7		
STII 14		
STII 5		
STII 12		
STII 3		
STII 10		
STII 1		
STII 8		
STII 15		
STII 6		
STII 13		
STII 4		
STII 11		
STII 2		
STII 9		
STII 0		
LBI 2,0		
STII 7		
STII 14		
STII 5		
STII 12		
STII 3		
STII 10		
STII 1		
STII 8		
STII 15		
STII 6		
STII 13		

INSTRUCTION	RESULT	COMMENTS
STII 4		
STII 11		
STII 2		
STII 9		
STII 0		
LBI 3,0		
STII 7		
STII 14		
STII 5		
STII 12		
STII 3		
STII 10		
STII 1		
STII 8		
STII 15		
STII 6		
STII 13		
STII 4		
STII 11		
STII 2		
STII 9		
STII 0		

INSTRUCTION	RESULT	COMMENTS
LBI 0,0		CHECK FOR RAM DATA
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT

INSTRUCTION	RESULT	COMMENTS
LBI 1,0		CHECK FOR RAM DATA
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT

TABLE I. Typical Test Sequence (Continued)

INSTRUCTION	RESULT	COMMENTS
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT

INSTRUCTION	RESULT	COMMENTS
LBI 1,0		CHECK FOR RAM DATA
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT

INSTRUCTION	RESULT	COMMENTS
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT

INSTRUCTION	RESULT	COMMENTS
LBI 3,0		CHECK FOR RAM DATA
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT
OMG		OUTPUT DATA
LD		:
XIS		:MOVE TO NEXT DIGIT

INSTRUCTION	RESULT	COMMENTS
JMP X	INITIALIZE—SELECT ADDRESS X FOR OGI OR OMG (SELECT LBI FOR KNOWN DATA)	
RELEASE TEST MODE	OBD (SELECT B FOR KNOWN CONDITION) CHECKS JMP	

4

TABLE I. Typical Test Sequence (Continued)

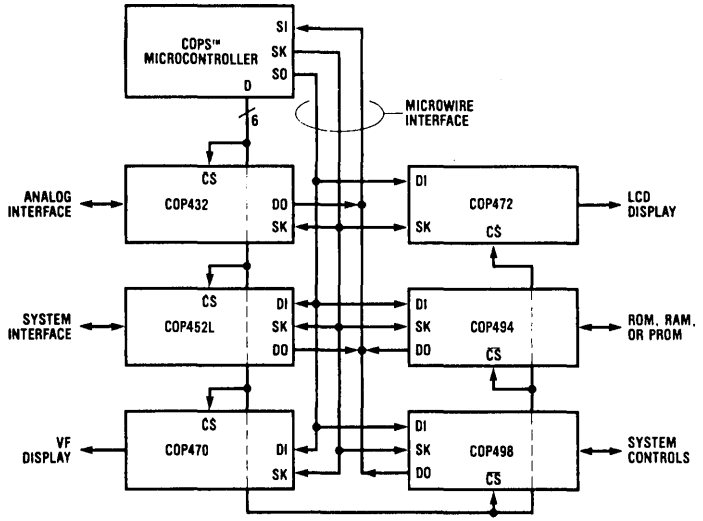
INSTRUCTION	RESULT COMMENTS
SET TEST MODE	
JP X-2	:
JSR Y	CHECK JP & JSR
RELEASE TEST MODE	"Y" SHOULD CHANGE THE OUTPUT CONDITIONS OF "X"
EXECUTE CODE (Y)	IF AT ALL POSSIBLE
SET TEST MODE	
RET	
RELEASE TEST MODE	
EXECUTE "X" AGAIN	VERIFIES RET
SET TEST MODE	
JP X-2	
JSRP Z	CHECK JSRP & RETSK
RELEASE TEST MODE	
EXECUTE CODE	"Z" SHOULD CHANGE "X" OUTPUT CONDITIONS
SET TEST MODE	
RETSK	DON'T CHANGE Z CONDITIONS — RETSK
RELEASE TEST MODE	
EXECUTE	" "
SET TEST MODE	
LOAD A & M TO	FIND VALUE OF ADDRESS IN BLOCK (4 PAGES)
VALUE OF ADDRESS	AT OR JUST BEFORE AN OUTPUT
TO GO TO	CHANGE SET A & M TO ADDRESS
OUTPUT CHANGE	OF "VALUE"
JID	CHECKS JID
RELEASE TEST MODE	
EXECUTE OUTPUT	
SET TEST MODE	LOAD A & M WITH A UNIQUE ADDRESS
LOAD A & M	SUCH THAT CONTENTS OF THAT ADDRESS WILL BE SEEN ON G
LQID	
X064	;OR USE THIS CAUSE THE DATA COMES ;FROM YOUR TESTER ANYWAY
CQMA	
OMG	LQUID & CQMA CHECKED
X	
OMG	"
INL	:
OMG	G − > 2 INL TEST (COPY OF 2nd BYTE)
X	
OMG	G − > E :

This test sequence is not to be taken as a recommended test routine and is only shown as an example of what might be done to test various COPS parts. It is also advisable to approach measurements in the test mode with some caution. As stated earlier, one can force a large current into the SO node to place the chip in the test mode. Not only can this current do damage if unlimited, but it can also cause local current overloading such that some I/O conditions may be adversely affected. Obviously this will be more pronounced at higher V_{CC} voltages. A specific example is that the L output current sink test should only be tested at a V_{OUT} of 0.4V and 0.36 mA as the more stringent tests can exceed power limits when combined with the SO current.

MICROWIRE™

National's super-sensible MICROWIRE serial data exchange standard allows interfacing to any number of specialized peripherals using an absolute minimum number of valuable I/O pins; this leaves more I/O lines available for system interfacing and/or may permit the COPS controller to be packaged in a smaller (and even lower cost) package. (MICROWIRE peripherals may also be used with non-COPS controllers). For further applications information, refer to COPS Briefs 8 and 9. MICROWIRE makes sense.

The example below illustrates the power and versatility of MICROWIRE via an extreme example—using one of each type of peripheral with a single controller.



TL/DD/6940–4

COP431 SERIES, 8-BIT A/D CONVERTERS

The COP431 series is an 8-bit successive approximation A/D converter with a serial I/O and configurable input multiplexer with up to 8 channels. The serial I/O is configured to comply with the NSC MICROWIRE serial data exchange standard for easy interface to the COPS family of processors, and can interface with standard shift registers or other μPs.

The 2, 4 or 8 channel multiplexers are software configured for single-ended or differential inputs as well as channel assignment.

The differential analog voltage input allows increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

COP452L FREQUENCY/COUNTER PERIPHERAL

The COP452L contains 2 independent 16-bit counter/register pairs, and is well suited to a wide variety of tasks involving the measurement and/or generation of times and/or frequencies. Included are multiple tones, precise duty cycles, event counting, waveform measurement, "white noise" generation, and A-D/D-A conversions. An on-chip zero-crossing detector can trigger a pulse with a programmed delay and duration.

COP470 V.F. DISPLAY DRIVER

The COP470 is designed to directly drive a multiplexed Vacuum Fluorescent display. Data is loaded serially and held in internal latches. The COP470 has an on-chip oscillator to multiplex four digits of eight segment display, and may be cascaded and/or stacked to drive more digits, more segments, or both.

With the addition of external drivers, the COP470 also provides a convenient means of interfacing to a large-digit LED display.

COP472-3 LIQUID CRYSTAL DISPLAY CONTROLLER

The COP472-3 Liquid Crystal Display (LCD) Controller drives a multiplexed liquid crystal display directly. Data is loaded serially and is held in internal latches. The COP472-3 contains an on-chip oscillator and generates all the multilevel waveforms for backplanes and segment outputs on a triplex display. One COP472-3 can drive 36 segments multiplexed as 3 × 12 (4½ digit display). Two COP472-3 devices can be used together to drive 72 segments (3 × 24) which could be an 8½ digit display.

COP494 256-BIT SERIAL ELECTRICALLY ERASABLE PROGRAMMABLE MEMORY

The COP494 is a 256-bit non-volatile memory. The device contains 256 bits of read/write memory divided into 16 registers of 16 bits each. Each register is serially read or written by the COP400 Family Controller. Written information is stored in a floating gate cell with at least 10 years of retention.

COP498/COP499 LOW POWER CMOS RAM AND TIMER

The COP498 low power CMOS Random-Access Memory and Timer is an external memory and timer chip with the simple MICROWIRE serial interface. The device contains 256 bits of read/write memory divided into 4 registers of 64 bits each. The COP498 also contains a crystal-based timer for timekeeping purposes, and can provide a "wake-up" signal to turn on a COPS controller.

The COP498 can be used for low power standby memory and can also be used for low power operation by turning the controller off and on, on a duty cycle basis.

The COP499 Low Power CMOS Random-Access Memory is an external memory and switch chip with the simple MICROWIRE serial interface. The device contains 256 bits of read/write memory divided into 4 registers of 64 bits each. The COP499 also contains circuitry that enables the user to turn a controller on and off while maintaining the integrity of the memory.

Current Consumption in NMOS COPS™ Microcontrollers

National Semiconductor Corp.
Application Brief 3
Len Distaso

Current consumption in the N-channel COPS microcontrollers is a function of manufacturing process variation and three operating condition parameters: temperature, voltage, and frequency. The aforementioned process variation swamps all other variations. Of the operating condition parameters, temperature is by far the most significant. This application brief is intended to provide the user with a guide to approximate the worst-case current consumption of the NMOS COPS microcontroller at a given set of operating conditions and to approximate the current variation with respect to temperature, voltage, and frequency.

Note that this is a guide only. Some approximations in the equations have been made. Only the current values found in the various device data sheets are guaranteed. Values derived by the techniques described here are neither guaranteed nor tested.

PROCESS VARIATION

If a user were to measure the current in two identical COPS microcontrollers under identical operating conditions (i.e., same temperature, voltage, and frequency), the results would probably be different. The reason for this difference is variation in the manufacturing process within its valid range. This variation can be quite substantial; a range of about 3 to 1 can be expected. This variation is essentially a device-to-device variation and basically not related to the operating conditions of the device. The three operating condition parameters (temperature, voltage, and frequency) affect current in the manner described below.

The values for current consumption in the various device data sheets are worst-case maximum values and assume that the processing parameters are at the end of the valid range which will produce maximum current consumption in the device.

THE EFFECT OF FREQUENCY

The frequency effect on current consumption is primarily a device design consideration. The higher the intended operating frequency, the higher the maximum current. However, once the device is designed in this process for a given maximum frequency, there is little variation with operating frequency. To be sure, there is some variation. As might be expected, current consumption is greater at higher frequencies. The variation is, however, slight—typically less than 5%.

THE EFFECT OF VOLTAGE

The operating voltage of the microcontroller has a slightly greater effect on current consumption than the operating current. Current consumption increases with increasing operating voltage. On examining the MOS device equations, one finds that the device current is proportional to the square of a voltage term:

$$I \propto (V_{GS} - V_T)^2$$

where:

I = device current

V_{GS} = device gate to source voltage

V_T = device threshold voltage.

In the N-channel COPS devices, current is consumed primarily by the load devices. Most of these devices, though not all, are depletion mode devices with the gate and source tied together. Thus, V_{GS} is 0. Therefore, the primary mechanism for current consumption as related to voltage is variation in V_T. The depletion mode load devices in the COPS NMOS microcontrollers have geometries (length is much greater than width) which tend to minimize variations in threshold voltage. There are additional second order effects related to operating voltage, such as effective channel lengths shortening due to increased voltage, which affect current consumption. These effects, however, do not have a major impact on current consumption. Note also that the threshold voltage is affected by process variation. This is one of the areas where the process variation contributes to the device-to-device variation in current consumption. The user can typically expect to see a 5% to 10% variation in current due to operating voltage with the maximum current consumption occurring at maximum operating voltage.

THE EFFECT OF TEMPERATURE

Of the three operating parameters affecting current consumption in the NMOS COPS microcontrollers, temperature has by far the greatest impact. The relationship is given by the following simplified, empirical equation:

$$I(T) = I_O(T/T_O)^{-3/2}$$

where:

T_O = reference junction temperature in °K

T = device junction temperature in °K

I_O = device current at temperature T_O

$I(T)$ = device current at temperature T.

Although this equation is for a single transistor, it can be applied to the entire microcontroller since all the devices are made with the same process and will exhibit the same

characteristics. It should also be noted that the temperatures involved are device junction temperatures. The junction temperature is essentially a function of two items:

$T_j = F(T_A, \theta_{jA})$

where:

T_j = junction temperature

T_A = ambient temperature

θ_{jA} = package thermal characteristic.

The preceding relationship indicates that the package for the device will affect current because the package affects junction temperature. This should not come as a surprise. One need only consider the differences between ceramic and plastic packages to find support for this claim.

For purposes of discussion, it will be assumed that junction temperature is given by the following:

$T_j = T_A + 25°K$

where T_j and T_A are as defined previously. Note that this is an approximation. It is not necessarily true for all packages, or any package. The relationship between junction temperature and ambient temperature is also not necessarily linear. However, the approximation is reasonable and provides a workable framework.

Substituting the junction temperature relationship into the current equation, the following equation results:

$$I(T_A) \cong I_O \left(\frac{T_A + 25}{T_{AO} + 25}\right)^{-3/2}$$

where:

T_{AO} = reference ambient temperature, °K

T_A = ambient temperature, °K

I_O = current at ambient temperature T_{AO}

$I(T_A)$ = current at ambient temperature T_A.

AN EXAMPLE

The COP320L has a specified maximum current of 10 mA. In this process, maximum current occurs at minimum temperature, which is −40°C in this case. It is desired to find the maximum current at 25°C. Therefore,

$T_{AO} = -40°C = 233°K$

$T_A = 25°C = 298°K$

$I_O = 10$ mA

$I(T_A)$ to be determined

$$I(T_A) \cong I_O \left(\frac{T_A + 25}{T_{AO} + 25}\right)^{-3/2}$$

$$\cong 10 \text{ mA } (323/258)$$

$$\cong 7.14 \text{ mA.}$$

Thus the maximum current for the COP320L at 25°C is approximately 7 mA.

CONCLUSION

A means is provided to the user to approximate the current variation of the NMOS COPS microcontroller over its valid operating range. A given device will consume its maximum current at maximum operating voltage, maximum operating frequency, and minimum operating ambient temperature. Conversely, minimum current will be consumed at minimum operating voltage, minimum operating frequency, and maximum operating ambient temperature.

The user should remember that this document is intended as a guide only. The values produced here are reasonable but they are approximations and are not guaranteed values. The user should also remember that the equations and methods discussed here do not involve process variation. The numbers calculated approximate the worst-case maximum current values at a given set of operating conditions. The user should be prepared to see a wide range of values over the course of volume production.

Further Information on Testing of COPS™ Microcontrollers

National Semiconductor Corp.
Application Brief 4
Len Distaso

COP Note 7 describes the basic approach and philosophy for testing COPS microcontrollers. This application brief is intended to complement and expand COP Note 7. It is assumed that the reader is familiar with and has access to COP Note 7.

TEST MODE

On COPS microcontrollers, test mode is entered by forcing the SO output to a logic "1" when it should otherwise be a logic "0". The easiest way to do this is to hold the COPS device in reset, hold the $\overline{\text{RESET}}$ pin low, and pull SO up to a logic "1" level. WARNING: *Do not force more than 3.0V on SO, as damage to the device may occur.* SO should be forced to approximately 2.5V to guarantee entry into test mode and to protect the device from damage.

Once the device is in test mode, the state of the SI input controls the type of test. SI at a logic "1" (high level) conditions the device to accept instructions from an external source via the L port. In test mode, when SI is high, the internal ROM is disabled. SI at a logic "0" (low level) forces the device to dump the internal ROM to the L port where the user can read and verify the ROM contents.

INSTRUCTION INPUT

With the device in test mode and SI at a logic "1", the microcontroller will read the data at the L port as instructions. The instructions must be presented at the beginning of each cycle time and must remain valid during the whole cycle time. The chip SK output is the instruction cycle clock in test mode and can be used as the timing reference. *Figure 1* indicates the timing for instruction input using the chip's SK output as the reference. A new instruction must be valid at the L inputs within approximately 200 ns of the rising edge of SK. The user should make every effort to make this time (t2 in *Figure 1*) as short as possible.

It is possible to create an external SK signal which more closely duplicates the internal SK. This requires building a divider from CKI and synchronizing the resultant signal with the device under test. This is significant because it is the internal version of the SK signal which is the master timing signal for the microcontroller. The short time from the rising edge of the SK output to instruction valid is necessary because the actual objective is to provide new instructions at the rising edge, or close to it, of the internal timing signal. If the user creates the external timing signal, the 200 ns time is not applicable. A new instruction, or ROM word, would be presented at each rising edge of the external signal. A method for generating and using this external SK is described in COP Note 7.

ROM DUMP

With SI at logic "0" in test mode, the microcontroller will dump the ROM to the L port. ROM will be dumped sequentially, one word at a time, starting at whatever value the program counter contains. A new ROM word appears at the L lines every falling edge of the chip SK signal. The output timing (t1 in *Figure 1*) is the L output timing as found in the various device data sheets. The device will remain in ROM dump mode as long as SI is at logic "0" in test mode. The program counter will wrap around from the maximum address to 000 and ROM dump will continue.

To get a ROM dump, the user cannot simply enter test mode and force SI to logic "0". Some conditioning of the device is necessary. This requires that the user first go into instruction input mode and set up the device. The suggested sequence is as follows:

1. Enter test mode—pull $\overline{\text{RESET}}$ low, force SO to about 2.5V.
2. Force SI to logic "1" and force 0s on L lines—RESET still low.
3. Force RESET high and input the following sequence to the device:

```
CLRA
JMP    3FC   (modify for ROM size)
LQID
O44H
LEI    4
NOP
```

4. During the NOP, change SI from high to low as shown in *Figure 2.* The ROM dump should start at address 000H at the time shown in *Figure 2.*

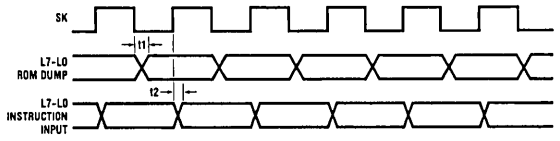
Figure 3 presents a general timing diagram for the entire sequence above. The jump instruction (JMP 3FC) in the sequence is used merely to position the program counter so that the ROM dump will begin at a specified location. That jump will be modified to reflect different ROM sizes or different desired starting locations for the ROM dump.

CHANGING BETWEEN INSTRUCTION INPUT AND ROM DUMP

The change from instruction input to ROM dump is accomplished according to the timing in *Figure 2.* It is necessary to do this to perform a valid ROM dump. However, it is not recommended to go the other direction, from ROM dump to instruction input, "on the fly". The instruction input mode should only be entered while the device is reset, $\overline{\text{RESET}}$ line low, to guarantee proper timing.

CONCLUSION

With COP Note 7 and this application brief, the user should be able to create a workable functional test for his COPS microcontroller. The relative timing is presented here and general techniques and sequences are provided in COP Note 7.

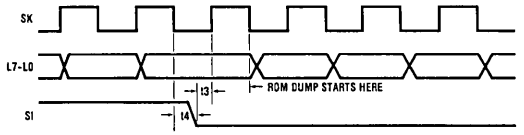


t1 = L output timing (t_{PD1}, t_{PD0}) as found in data sheet

t2 ~ 200 ns max

TL/DD/5146-1

FIGURE 1. Basic Test Mode Timing

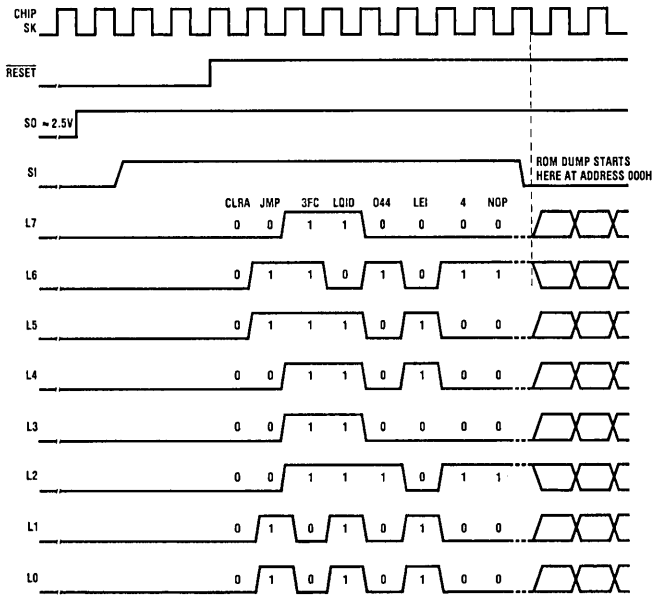


t4 ≈ t3 ~ 1 μs min for 4 μs devices

t4 ≈ t3 ~ 4 μs min for 16 μs devices

TL/DD/5146-2

FIGURE 2. Timing for Changing from Instruction Input to ROM Dump—Test Mode



TL/DD/5146-3

FIGURE 3. Relative Timing for Suggested Sequence to Generate ROM Dump

COPS™ Interrupts

National Semiconductor Corp.
Application Brief 6
Jim Murashige

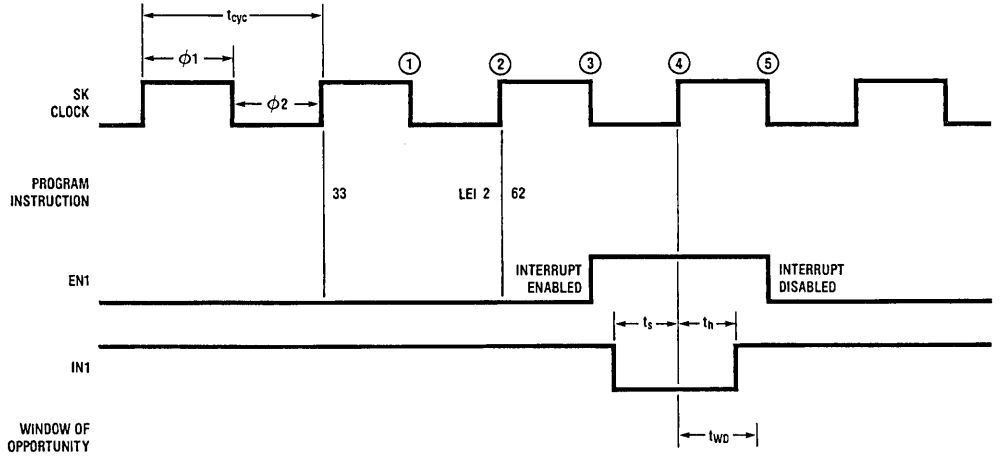
This brief describes in detail the timing requirements pertinent to COPS interrupts. *Figure 1* shows a typical enable-interrupt sequence in relation to the SK (Instruction Cycle) Clock. The SK clock is actually derived afrom the $\phi 1$ clock which is 180° out of phase with the $\phi 2$ clock. It is the $\phi 1$ and $\phi 2$ clocks to which all operation is referenced but for our purposes the SK will suffice. Program instructions are read on a rising $\phi 1$ edge and executed during the $\phi 1$, $\phi 2$ cycle time. Here we see the EN register interrupt enable bit EN2 being set with an LEI instruction. Interrupts are actually enabled on the $\phi 2$ leading edge of the second byte of the instruction point ③. Timing for an INTERRUPT DISABLE is essentially the same.

The interrupt line is sampled on the leading edge of $\phi 1$ as shown and interrupts are recognized if the minimum setup and hold times shown are satisfied. Note that the guaranteed times are longer than the typicals. The interrupt signal conditioning circuitry contains a falling edge detection circuit (a one shot) which requires that in addition to meeting the setup and hold times, the enable interrupt bit EN1 must have been turned on sometime before the end of the WINDOW of OPPORTUNITY shown. If not, the interrupt will be missed and another high to low IN1 transition will be required. EN1 is automatically disabled upon interrupt recognition at point ⑤. Note that although the interrupt is recognized at point ④ it will not be acted upon until all successive transfer of control instructions are executed as defined in the data sheets.

Because of gate delays it is doubtful that if an interrupt had been generated in time to meet the leading $\phi 1$ edge at point ② that the EN1 enable bit would have been on in time to meet the WINDOW of OPPORTUNITY.

By doing a worst case analysis one can see that in order to guarantee reception of an asynchronous interrupt IN1 must remain low for at least 2 instruction cycles. The analysis is as follows. Assuming that interrupts had been enabled prior to point ①, if the interrupt arrives a little after point ① it will not satisfy the minimum setup requirements bringing us up to a point ⑤ our total elapsed time becomes ⑤ − ① = 2 t_{CYC}.

In a dual COPS the interrupt sequence is the same except that now an instruction cycle time is made up of both a Processor X and a Processor Y instruction execution cycle. With one $\phi 1$ and $\phi 2$ clock per processor execution cycle itne instruction cycle time is made up of 2 $\phi 1$'s and $\phi 2$'s. Therefore 1 instruction cycle time in a dual COPS is equivalent to 2 instruction cycle times in a single COPS as far as $\phi 1$'s, $\phi 2$'s and interrupts are concerned.



TL/DD/5180-1

FIGURE 1. COP Interrupt Diagram

Parameter	Min	Typ	Max
t_s	½ t_{CYC}	200 ns	
t_n	½ t_{CYC}	200 ns	
t_{wo}	$-\infty$	½ t_{CYC} − 600 ns	0

Protecting Data in the NMC9306/COP494 and NMC9346/COP495 Serial EEPROMs

National Semiconductor Corp.
Application Brief 15
Asim Bajwa

The NMC9306/COP494 and NMC9346/COP495 are non-volatile serial access memories with the following salient features:

- Low cost
- Single supply read/write/erase operation (5V ± 10%)
- TTL compatible
- MICROWIRE™ compatible I/O
- 16 × 16 serial read/write memory (NMC9306/COP494)
  64 × 16 serial read/write memory (NMC9346/COP495)
- Self-timed programming cycle (NMC9346/COP495 only)
- Ready/busy status signal during programming (NMC9346/COP495 only)
- Read-only mode

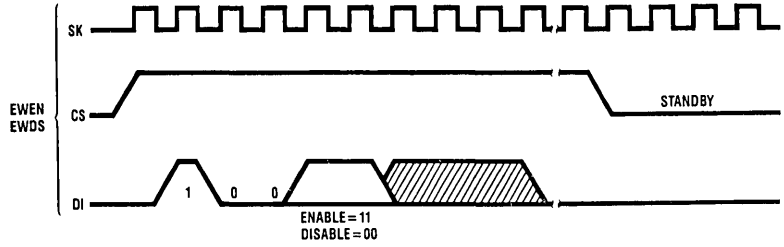
The read-only mode is provided to prevent accidental data disturb, especially during V_{CC} power up, power down or excessive noise on the I/O or power supply pins.

Executing the EWDS instruction (*Figure 1*) activates this mode by disabling the programming modes and the high voltage pump. The READ instruction is not affected and can be executed as usual. However, all programming instructions (ERASE, WRITE, ERAL and WRAL) are ignored until the EWEN instruction is executed to enable programming.

On V_{CC} power up the device is designed to automatically enter the read-only mode to avoid accidental data loss due to power up transients. Putting the device in the read-only mode before powering down V_{CC} avoids spurious programming during power down.

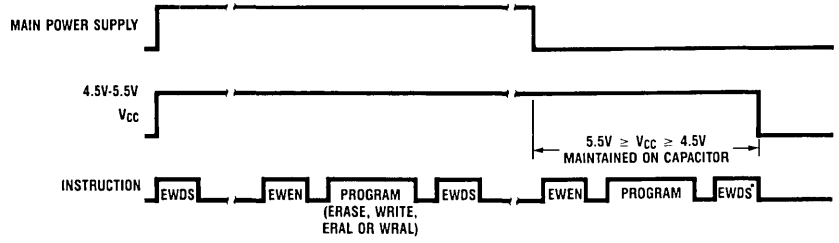
The following guidelines are presented and should be incorporated into the user's designs to achieve the maximum possible protection of stored data *(Figure 2)* :

1) The device powers up in the read-only mode. However, as a backup, the EWDS instruction should be executed as soon as possible after V_{CC} to the EEPROM is powered up to ensure that it is in the read-only mode.
2) Immediately preceding a programming instruction (ERASE, WRITE, ERAL or WRAL), the EWEN instruction should be executed to enable the device for programming; the EWDS instruction should be executed immediately following the programming instruction to return



TL/D/7085–1

FIGURE 1. EWEN, EWDS Instruction Timing



TL/D/7085–2

*EWDS must be executed before V_{CC} drops below 4.5V to prevent accidental data loss during subsequent power down and/or power up transients.

FIGURE 2. Typical Instruction Flow for Maximum Data Protection

the device to the read-only mode and protect the stored data from accidental disturb during subsequent power transients or noise.

3) Special care must be taken in designs in which programming instructions are initiated to store data in the EEPROM after the main power supply has gone down. This is usually accomplished by maintaining V_{CC} for the EEPROM and its controller on a capacitor for a sufficient amount of time (approximately 50 ms, depending on the clock rate) to complete these operations. This capacitor must be large enough to maintain V_{CC} between 4.5 and 5.5 volts for the total duration of the store operation, INCLUDING the execution of the EWDS instruction immediately following the last programming instruction. FAILURE TO EXECUTE THE LAST EWDS INSTRUCTION BEFORE V_{CC} DROPS BELOW 4.5 VOLTS MAY CAUSE INADVERTENT DATA DISTURB DURING SUBSEQUENT POWER DOWN AND/OR POWER UP TRANSIENTS.

COPS™ Peripheral Chips

National Semiconductor Corp.
Application Brief 28

There are several I/O peripheral chips that are compatible with the COPS microcontroller by communicating through the serial I/O port.

Two different sets of timing employed by them are shown in *Figure 2*. A brief description of the electrical characteristics of each chip is given below.

COP452 FREQUENCY/COUNTER PERIPHERAL

The COP452 is fabricated using N-channel silicon-gate MOS technology. Containing 2 independent 16-bit counter/register pairs, it is well suited to a wide variety of tasks involving the measurement and/or generation of times and/or frequencies. Included are multiple tones, precise duty cycles, event counting, waveform measurement, "white noise" generation, and A-D/D-A conversions. An on-chip zero-crossing detector can trigger a pulse with a programmed delay and duration.

COP470 V.F. DISPLAY DRIVER

The COP470 is designed to directly drive a multiplexed Vacuum Fluorescent display. Data is loaded serially and held in internal latches. The COP470 has an on-chip oscillator to multiplex four digits of eight segment display, and may be cascaded and/or stacked to drive more digits, more segments, or both.

With the addition of external drivers, the COP470 also provides a convenient means of interfacing to a large-digit LED display.

COP472 LIQUID CRYSTAL DISPLAY CONTROLLER

The COP472 Liquid Crystal Display (LCD) Controller is fabricated using CMOS technology. It drives a multiplexed liquid crystal display directly. Data is loaded serially and is held in internal latches. The COP472 contains an on-chip oscillator and generates all the multilevel waveforms for backplanes and segment outputs on a triplex display. One COP472 can drive 36 segment multiplexed as 3 x 12 (4½ digit display). Two COP472 devices can be used together to drive 72 segments (3 x 24) which could be an 8½ digit display.

COP494 256-Bit Serial Electrically erasable programmable memory. The COP494 is a 256-bit non-volatile memory. The device contains 256 bits of Read/Write memory divided into 16 registers of 16 bits each. Each register is serially read or written by the COP400 controller. Written information is stored in a floating gate cell with at least 10 years retention.

COP498/COP499 LOW POWER CMOS RAM AND TIMER

The COP498 low power CMOS Random-Access Memory and Timer is an external memory and timer chip with the simple MICROWIRE™ serial interface. The device contains 256 bits of read/write memory divided into 4 registers of 64 bits each. The COP498 also contains a crystal based timer for timekeeping purposes, and can provide a "wake-up" signal to turn on a COPS controller.

The COP498 can be used for low power standby memory and can also be used for low power operation by turning the controller off and on, on a duty cycle basis.

The COP499 Low Power CMOS Random-Access Memory is an external memory and switch chip with the simple MICROWIRE serial interface. The device contains 256 bits of read/write memory divided into 4 registers of 64 bits each. The COP499 also contains circuitry that enables the user to turn a controller on and off while maintaining the integrity of the memory.

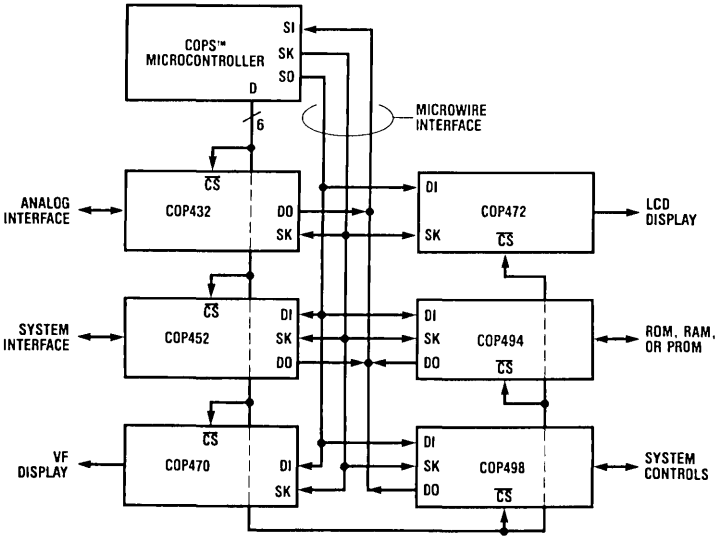


FIGURE 1

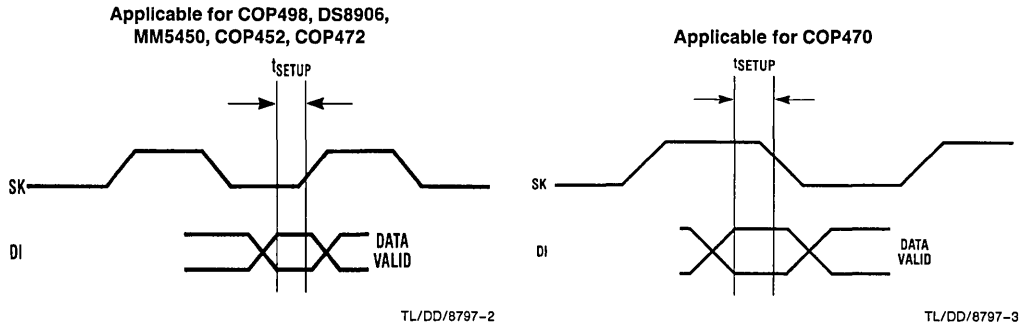


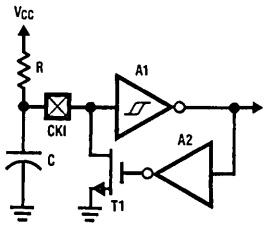
FIGURE 2. Serial Input Data Timing

A Users Guide to COPS™ Oscillator Operation

National Semiconductor Corp.
Application Note 326
Jim Murashige

The following discussion is an overview of the COPS oscillator circuits meant to give the reader a working knowledge of the circuits. Although the descriptions are very general and light on detail; a background in complex frequency analysis is necessary. For additional information the references cited should be consulted as well as the many works on oscillator theory.

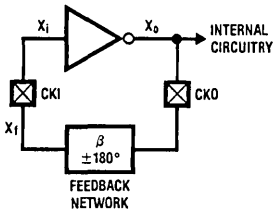
There are 2 basic circuits from which all of the COPS oscillator options are provided. (See option lists in individual data sheets.) The first and simplest in description is the astable one shot of *Figure 1* which gives us our RC oscillator option. A1 and A2 are inverters with A1 possessing a Schmitt trigger input. T1 is a large N channel enhancement MOS FET. Operation with the external R-C shown is as follows. Assuming C is initially discharged the CKI pin is low forcing T1 off. As C charges through R the trigger point of A1 is eventually reached at which time T1 is turned on discharging C and beginning a new cycle. Although almost any combination of R-C could be chosen, we would ideally like to have as short a discharge time as possible thereby eliminating the high variability in T1 drain current from device to device as a timing factor. For this reason R is chosen very large and C very small. This choice also leads to minimum R-C power dissipation. For the CKI Schmitt trigger clock input option the T1 MOS FET is merely mask disabled from the oscillator circuit.



TL/DD/5139-1

FIGURE 1. R-C Oscillator

The second oscillator circuit is the classic phase shift oscillator depicted in *Figure 2*. Found not only on COPS but on most other microprocessor circuits it is the simplest oscillator in terms of component complexity but the most difficult to analyze.



TL/DD/5139-2

FIGURE 2. Phase Shift Oscillator

The conditions under which the circuit will oscillate are described by the Barkhausen Criterion which states that oscillation will occur at the frequency for which the total loop phase shift from x_i to x_f is 0° or a multiple of 360° (i. e., x_f is identical to x_i). In addition the total loop gain must be > 1 to insure self propagation. The inverting amplifier shown between x_i and x_o provides 180° of phase shift thus leaving the feedback network to supply the other ±180°. The feedback network can be comprised of active or passive components but highly effective oscillators are possible using only passive reactive components and the general configuration of *Figure 3*.

If you work out the feedback loop equations for *Figure 3* it can be shown that in order to achieve ±180° phase shift:

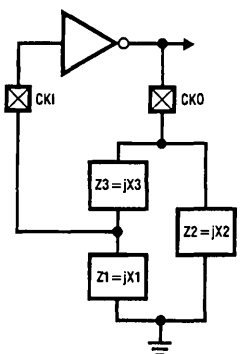
$$X1 + X2 + X3 = 0 \qquad (1)$$

X1 and X2 must both be inductors or capacitors (2)

therefore X3 is inductive if X1 is capacitive and vice versa

if X1 and X2 are capacitors it is a Colpitts Oscillator

X1 and X2 are inductors it is a Hartley Oscillator



TL/DD/5139-3

FIGURE 3. Typical Feedback Configuration

The Colpitts configuration is commonly shown in microprocessor oscillator circuits (*Figure 5*) with the inductive X3 replaced by a crystal for reasons we shall soon see. The equivalent electrical model of a crystal is shown in *Figure 4b* and a plot of its Reactance versus Frequency shown in *Figure 4c*. R-L-C represent the electro-mechanical properties of the crystal and C_0 the electrode capacitance. There are 2 important points on the reactance curve labeled f_a and f_b.

$$\text{At } f_a = \frac{1}{2\pi}\sqrt{\frac{1}{LC}}$$

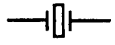
the crystal is at series resonance with L and C canceling each other out leaving only a nonreactive R for 0 phase shift. This mode of operation is important in oscillator circuits where a non-inverting amplifier is used and 0° phase shift must be preserved.

$$\text{At } f_b = \frac{1}{2\pi}\sqrt{\frac{1}{LC} + \frac{1}{LC_C}}$$

which is just a little higher than f_a the crystal is at parallel resonance and appears very inductive or capacitive. Note that the cyrstal will only appear inductive between f_a and f_b and that it becomes highly inductive very quickly. In addition f_b is only a fraction of a percent higher than f_a. Therefore the only time that the crystal will satisfy the X3 = −(X1 + X2) condition in the Colpitts configuration of *Figure 5* is when the circuit is oscillating between f_a and f_b. The exact frequency will be the one which gives an inductive reactance large enough to cancel out:

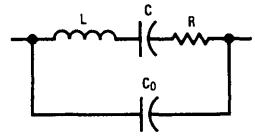
$$X1 + X2 = \frac{1}{\omega C1} + \frac{1}{\omega C2} = \frac{1}{\omega}\left[\frac{1}{C1} + \frac{1}{C2}\right] = \frac{1}{2\pi f}\left[\frac{1}{C_L}\right]$$

Therefore by varying C1 or C2 we can trim slightly the oscillator frequency.



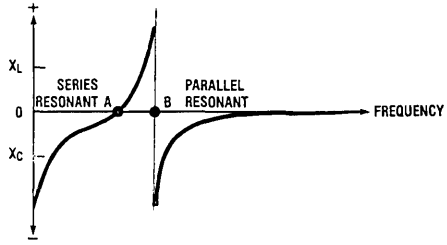
TL/DD/5139-4

a. Circuit Symbol



TL/DD/5139-5

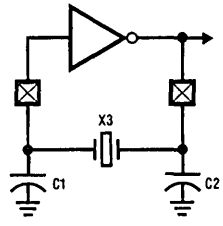
b. Electrical Equivalent



TL/DD/5139-6

c. Reactance Versus Frequency

FIGURE 4. Quartz Crystal

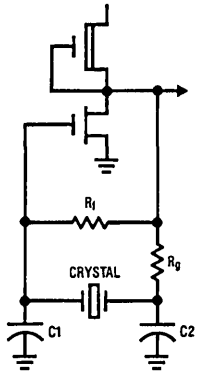


TL/DD/5139-7

FIGURE 5. Colpitts Oscillator

The Q of a circuit is often bounced around in comparing different circuits and can be viewed graphically here as the slope of the reactance curve between f_a and f_b. Obviously the steeper the curve the smaller the variation in f necessary to restore the Barkhausen Phase Shift Criterion. In addition a lower Q (more R) means that the reactance curve won't peak as high at f_b, necessitating a smaller X1 + X2. When selecting crystals the user should be aware that the frequency stamped on the cans are for either parallel or series resonance, which, although very close, may matter significantly in the particular application.

An actual MOS circuit implementation of *Figure 5* is shown in *Figure 6*. It consists of a MOS inverter with depletion load and the crystal π network just presented. External to the COPS chips are the R_f and R_g resistors. R_f provides bias to the MOS inverter gate $V_g = V_o$. Since the gate draws no current R_f can be very large (MΩ) and should be, since we do not wish it to interact with the crystal network. R_g increases the output resistance of the inverter and keeps the crystal from being over driven.

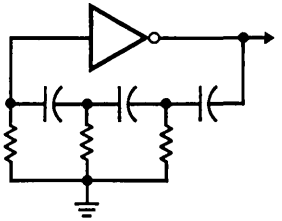


TL/DD/5139-8

FIGURE 6. MOS Oscillator

Of course the feedback network doesn't have to have the configuration of *Figure 3* and can be anything so long as the Barkhausen Phase Shift Criterion is satisfied. One popular configuration is shown in *Figure 7* where the phase shift will be 180°

$$\text{at } f = \frac{1}{(2\pi RC\sqrt{6})}$$

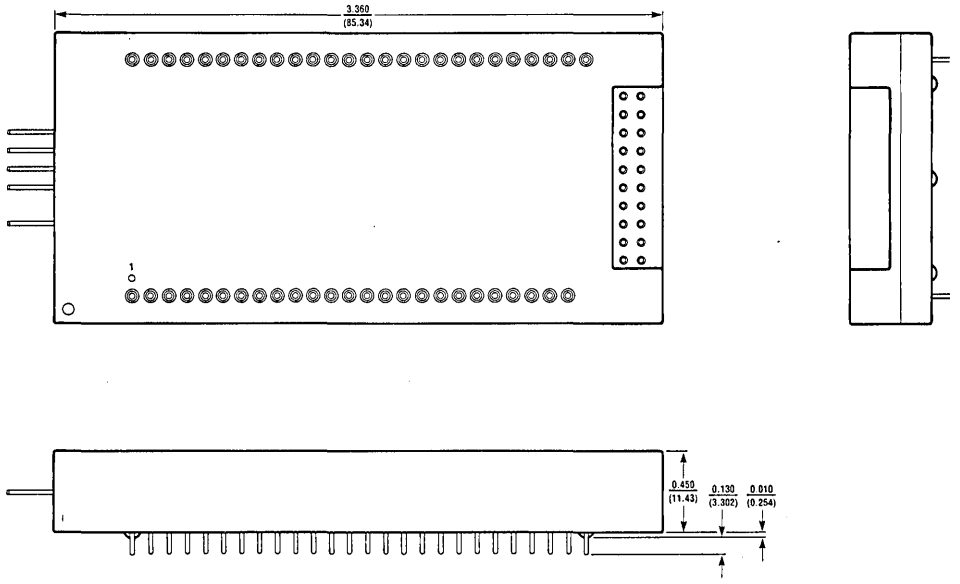


TL/DD/5139-9

FIGURE 7. R-C Phase Shift Oscillator

REFERENCES

1. Crystal/INS8048 Oscillator, AN-296, March 1982, National Semiconductor
2. Oscillator Characteristics of COPS Microcontrollers, CN-5, Feb. 1981, National Semiconductor
3. Integrated Electronics, Chapter 14, Millman and Halkias 1972
4. Handbook of Electronics Calculations, Chapter 9, Kaufman and Seidman 1979
5. 1982 COPS Microcontroller Databook, National Semiconductor



TL/DD/5139-10

Implementing an 8-Bit Buffer in COPS™

National Semiconductor Corp.
Application Note 329
David Pointer

Sometimes a COP microcontroller must input and/or output 8-bit data; for instance, when handling ASCII data. In some applications, the processor must also provide temporary storage for 8-bit data before it is output. The COP instruction set and RAM structure lend themselves very nicely to providing a 32 digit, 8-bit buffer for a solution to these applications.

Such a large buffer is possible using a COP440 or a COP444L. The other members of the COP400 family with half as much RAM as these two would provide a 16 digit 8-bit buffer using the techniques described in this example.

Four adjacent RAM registers (16 digits each) are required. Referring to *Figure 1*, registers 4, 5, 6, and 7 are used for the buffer. Each RAM location contains 4 bits, so 2 locations will be used to store a byte of data. But these RAM locations are not adjacent to each other. You will note that the MSD of digit number 0A hex is in RAM location (4, A) while the LSD of the same digit is in RAM location (6, A).

The 2 RAM locations CHARM and CHARL are used for temporary storage of an 8-bit value.

In addition, 4 RAM locations are used for buffer pointers: those labelled IPM and IPL are the MSD and LSD of the input pointer, and those labelled OPM and OPL are the MSD and LSD of the output pointer. Each pointer's function is to store an 8-bit counter whose value ranges from 00 hex thru 1F hex. The input pointer's value is used for storing the temporary storage buffer contents into the digit with the same number. For example, if the input pointer equals 14 hex, then the contents of CHARM would be stored in RAM location (5, 4) and the contents of CHARL would be stored in RAM location (7, 4). The output pointer's value is used for retrieving a digit from the buffer and putting it in CHARM and CHARL. For instance, if the output pointer equals 05 hex, then the contents of RAM location (4, 5) would be transferred to CHARM and the contents of RAM location (6, 5) would be transferred to CHARL.

A simple example of one possible application of the buffer is flowcharted in *Figure 2*. In this example, data is input to CHARM and CHARL, then stored in the buffer. An output device (a printer) is checked to see if it is ready to receive data. If it is, data is brought out of the buffer and put in CHARM and CHARL for output to the printer.

Pages 3 and 4 contain a listing of the subroutines needed to perform the data transfers in the 32-digit, 8-bit buffer.

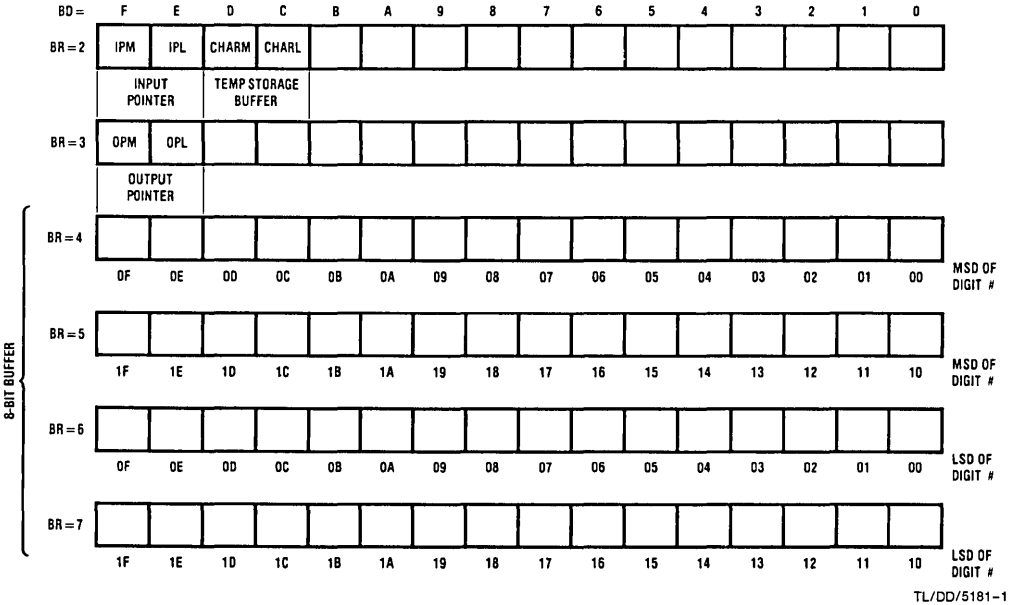
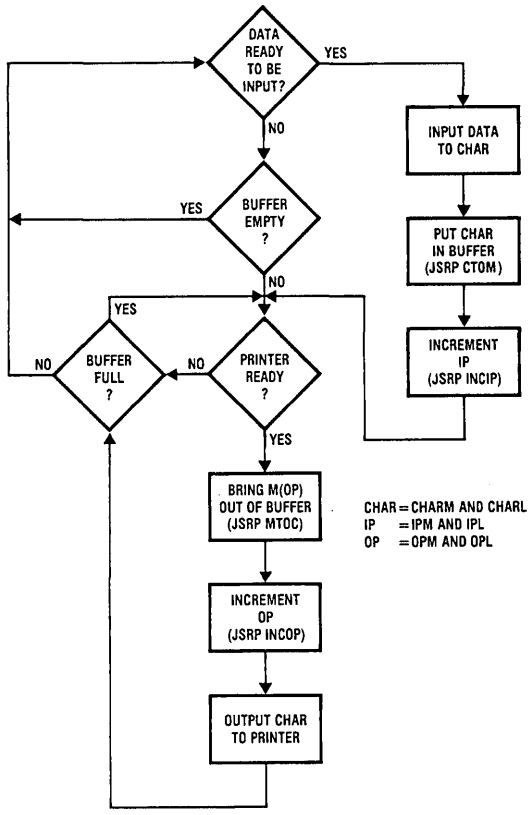


FIGURE 1. 8-Bit Buffer RAM Map



TL/DD/5181-2

FIGURE 2. Buffer Example Flowchart

```
COP CROSS ASSEMBLER    PAGE:  1
BUFFER

  1            ;*************************************
  2            ;***                               ***
  3            ;***  8-BIT RAM BUFFER SUBROUTINES  ***
  4            ;***                               ***
  5            ;*************************************
  6            ;THESE ARE SUBROUTINES FOR IMPLEMENTING A 32 BYTE
  7            ;BUFFER IN A COP440 OR COP444L RAM  9/3/82
  8     01BC   .CHIP 444
  9            .TITLE BUFFER
 10     002D   CHARM   =       2,13            ;TEMPORARY STORAGE BUFFER MSD
 11     002C   CHARL   =       2,12            ;TEMPORARY STORAGE BUFFER LSD
 12     002F   IPM     =       2,15            ;INPUT POINTER MSD
 13     002E   IPL     =       2,14            ;INPUT POINTER LSD
 14     003F   OPM     =       3,15            ;OUTPUT POINTER MSD
 15     003E   OPL     =       3,14            ;OUTPUT POINTER LSD
 16 000 00             CLRA
 17     0080   .PAGE 2
 18            ;MTOC IS A SUBROUTINE THAT TRANSFERS M(OPM) AND M(OPL) TO
 19            ;CHARM AND CHARL
 20 080 233E   MTOC:    LDD     OPL             ;LOAD LSD OUTPUT POINTER
 21 082 50              CAB                     ;WHICH IS BD
 22 083 233F            LDD     OPM              ;LOAD MSB OUTPUT POINTER FOR B
 23 085 54              AISC    4                ;MAKE BR EQUAL 4 OR 5
 24 086 12              XABR
 25 087 25              LD      2               ;LOAD M(OPM), MAKE BR = 6 OR 7
 26 088 23AD            XAD     CHARM           ;M(OPM) TO CHARM
 27 08A 05              LD                       ;LOAD M(OPL)
 28 08B 23AC            XAD     CHARL           ;M(OPL) TO CHARL
 29 08D 48              RET
 30            ;
 31            ;
 32            ;CTOM IS A SUBROUTINE THAT TRANSFERS CHARM AND CHARL TO
 33            ;M(IPM) AND M(IPL)
 34 08E 232E   CTOM:    LDD     IPL             ;LOAD LSD INPUT POINTER
 35 090 50              CAB                     ;WHICH IS BD
 36 091 232F            LDD     IPM             ;LOAD MSD INPUT POINTER FOR BR
 37 093 54              AISC    4               ;MAKE BR = 4 OR 5
 38 094 12              XABR
 39 095 232D            LDD     CHARM           ;LOAD MSD TEMP STORAGE
 40 097 26              X       2               ;TO M(OPM), MAKE BR = 6 OR 7
 41 098 232C            LDD     CHARL           ;LOAD LSD TEMP STORAGE
 42 09A 06              X                       ;TO M(OPL)
 43 09B 48              RET
 44            ;
 45            ;
```

```
COP CROSS ASSEMBLER    PAGE:  2
BUFFER

   46              .FORM
   47              ;INCREMENTS INPUT POINT OR OUTPUT POINTER, ROLLS OVER
   48              ;AT 1F HEX
   49 09C 2D       INCIP:  LBI     IPL             ;POINT TO LSD OF POINTER
   50 09D 3D       INCOP:  LBI     OPL
   51 09E 22               SC                      ;C=1 FOR INCREMENT
   52 09F 00               CLRA
   53 0A0 30               ASC                     ;INCREMENT RAM VALUE
   54 0A1 44               NOP                     ;NEGATES SKIP CONDITION
   55 0A2 04               XIS                     ;STORE AND POINT TO (X,F)
   56 0A3 00               CLRA
   57 0A4 30               ASC                     ;PROPAGATE CARRY, IF ANY, TO MS
   58 0A5 44               NOP
   59 0A6 06               X                       ;STORE
   60 0A7 45               RMB     1               ;ROLL OVER AT X'1F
   61 0A8 48               RET
   62              ;
   63              ;
   64              .END


COP CROSS ASSEMBLER    PAGE:  3
BUFFER

CHARL  002C     CHARM  002D     CTOM   008E *   INCIP  009C *
INCOP  009D *   IPL    002E     IPM    002F     MTOC   0080 *
OPL    003E     OPM    003F
NO ERROR LINES
   42  ROM WORDS USED
COP   444  ASSEMBLY
SOURCE CHECKSUM = C6A5
INPUT FILE    6:RBUFFC. SRC  VN:    5
```

Designing with the NMC9306/COP494 a Versatile Simple to Use E² PROM

National Semiconductor Corp.
Application Note 338
Masood Alavi

This application note outlines various methods of interfacing an NMC9306/COP494 with the COPS™ family of microcontrollers and other microprocessors. *Figures 1–6* show pin connections involved in such interfaces. *Figure 7* shows how parallel data can be converted into a serial format to be inputted to the NMC9306; as well as how serial data outputted from an NMC9306 can be converted to a parallel-format.

The second part of the application note summarizes the key points covering the critical electrical specifications to be kept in mind when using the NMC9306/COP494.

The third part of the application note shows a list of various applications that can use a NMC9306/COP494.

GENERIC CONSIDERATIONS

A typical application should meet the following generic criteria:

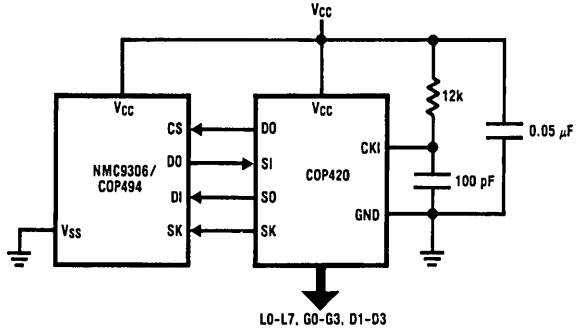
1. Allow for no more than 10,000 E/W cycles for optimum and reliable performance.
2. Allow for any number of read cycles.
3. Allow for an erase or write cycle that operates in the 10–30 ms range, and not in the tens or hundreds of ns range as used in writing RAMs. (Read vs write speeds are distinctly different by orders of magnitude in E²PROM, not so in RAMs.)
4. No battery back-up required for data-retention, which is fully non-volatile for at least 10 years at room-ambient.

SYSTEM CONSIDERATIONS

When the control processor is turned on and off, power supply transitions between ground and operating voltage may cause undesired pulses to occur on data, address and control lines. By using WEEN and WEDS instructions in conjunction with a LO-HI transition on CS, accidental erasing or writing into the memory is prevented.

The duty cycle in conjunction with the maximum frequency translates into having a minimum Hi-time on the SK clock. If the minimum SK clock high time is greater than 1 μs, the duty cycle is not a critical factor as long as the frequency does not exceed the 250 kHz max. On the low side no limit exists on the minimum frequency. This makes it superior to the COP499 CMOS-RAM. The rise and fall times on the SK clock can also be slow enough not to require termination up to reasonable cable-lengths.

Since the device operates off of a simple 5V supply, the signal levels on the inputs are non-critical and may be operated anywhere within the specified input range.



TL/D/5286–1

FIGURE 1. NMC9306/COP494 — COP420 Interface

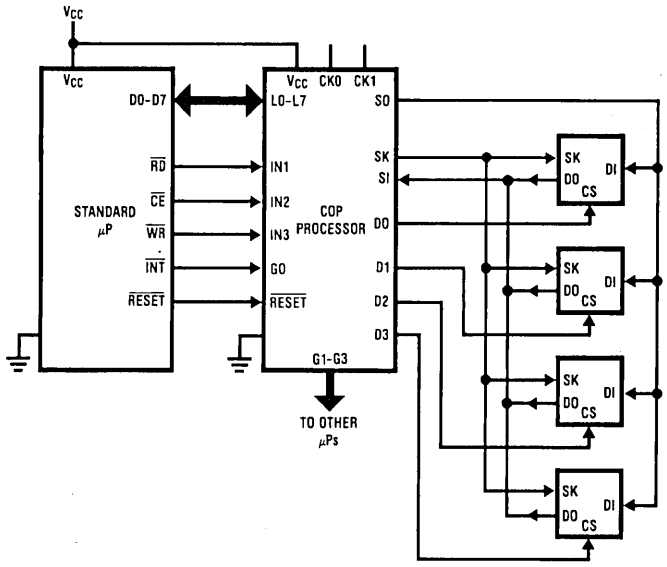
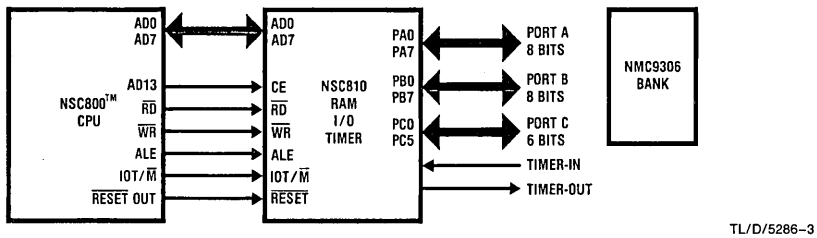


FIGURE 2. NMC93O6 — Standard μP Interface Via COP Processor

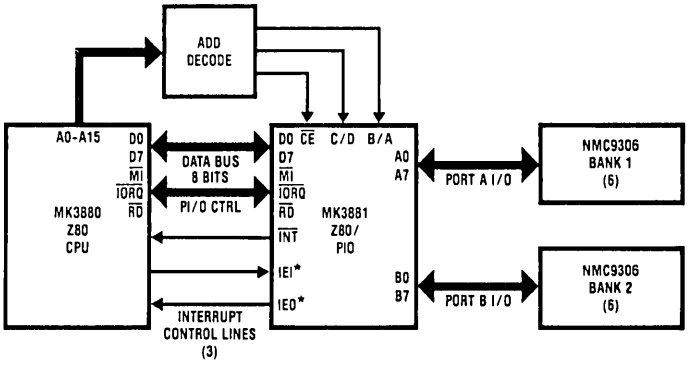


PA0 → SK
PA1 → DI/DO
} Common to all 9306's
PA2–7 → 6CS for 6– 9306's

*** SK is generated on port pins by bit-set and bit-clear operations in software. A symmetrical duty cycle is not critical.**

*** CS is set in software. To generate 10–30 ms write/erase the timer/counter is used. During write/erase. SK may be turned off.**

FIGURE 3. NSC800™ to NMC9306 Interface (also Valid for 8085/8085A and 8156)

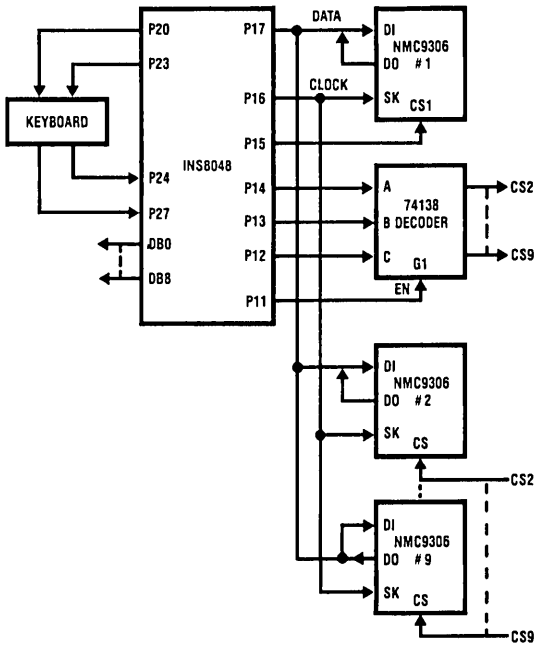


TL/D/5286–4

Z80-P10	9306	
A0	SK	Common to all 9306's (Bank 1)
A1	DI/DO	
A2-A7	CS1-CS6	

* Only used if priority interrupt daisy chain is desired
* Identical connection for Port B

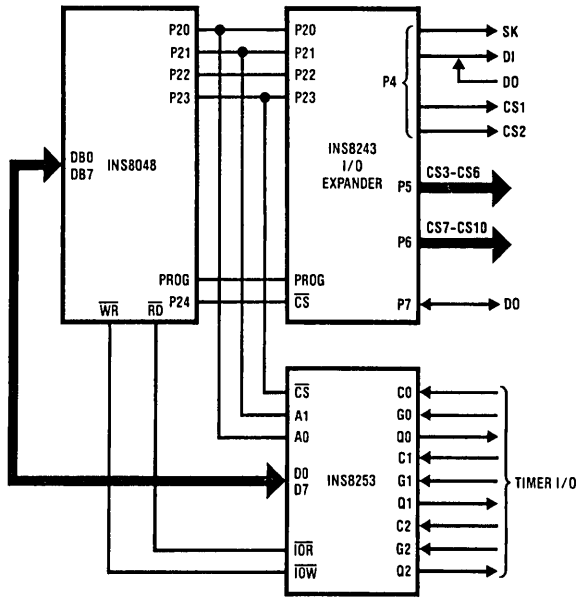
FIGURE 4. Z80 — NMC9306 Interface Using Z80-PIO Chip



TL/D/5286–5

* SK and DI are generated by software. It should be noted that at 2.72 μs/instruction. The minimum SK period achievable will be 10.88 μs or 92 kHz, well within the NMC9306 frequency range.
* DO may be brought out on a separate port pin if desired.

FIGURE 5. 48 Series μP — NMC9306 Interface



Expander outputs

	DI SK	(COMMON)
Port 4	CS1	
	CS2	
Port 5-6	CS3–CS10	
Port 7	DO (COMMON)	

FIGURE 6. 8048 I/O Expansion

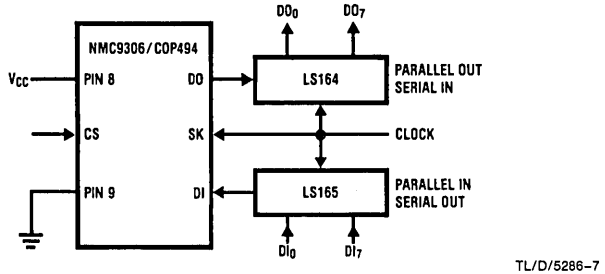
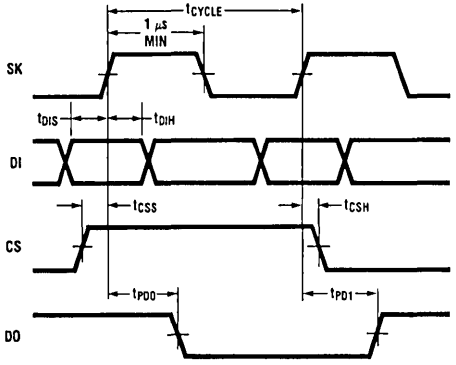


FIGURE 7. Converting Parallel Data into Serial Input for NMC9306/COP494



TL/D/5286–8

Min	Max
t_{CYCLE} 0	250 kHz
DUTY CYCLE 25%	75%
t_{DIS} 400	ns
t_{D1H} 400	ns
t_{CSS} 200	ns
t_{CSH} 0	ns
t_{PD0}	2 μs
t_{PD1}	2 μs

FIGURE 8. NMC9306/COP494 Timing

THE NMC9306/COP494

Extremely simple to interface with any μP or hardware logic. The device has six pins for the following functions:

Pin 1	CS*	HI enabled
Pin 2	SK	Clock input for data bit maneuvering
Pin 3	DI	For instruction or data input
Pin 4	DO**	For data read TRI-STATE® otherwise
Pin 5	GND	
Pin 8	V_{CC}	For 5V power
Pins 6–7	No Connect	No termination required

* Following an E/W instruction feed, CS is also toggled low for 10 ms (typical) for an E/W operation. This internally turns the VPP generator on (HI-LO on CS) and off (LO-HI on CS).

** DI and DO can be on a common line since DO is TRI-STATED when unselected DO is only on in the read mode.

USING THE NMC9306/COP494

The following points are worth noting:

1. SK clock frequency should be in the 0–250 kHz range. With most μPs in the 1–11 MHz range this is easily achieved when implemented in software by bit-set and bit-clear instructions, which take 4 instructions to execute a clock or a frequency in the 100 kHz range for standard μP speeds. Symmetrical duty cycle is irrelevant if SK HI time is ≥ 2 μs.
2. CS low period following an E/W instruction must not exceed the 30 ms max. It should best be set at typical or minimum spec of 10 ms. This is easily done by timer or a software connect. The reason is that it minimizes the 'on time' for the high V_{PP} internal voltage, and so maximizes endurance. SK-clock during this period may be turned off if desired.
3. All E/W instructions must be preceded by EWEN and should be followed by an EWDS. This is to secure the stored data and avoid inadvertent erase or write.
4. A continuously 'on' SK clock does not hurt the stored data. Proper sequencing of instructions and data on DI is essential to proper operation.
5. Stored data is fully non-volatile for up to ten years independent of V_{CC}, which may be on or off. For all practical purposes any number of read cycles have no adverse effects on data retention.
6. Up to 10,000 E/W cycles/register are possible. Under typical conditions, this number may actually approach 1 million. For applications requiring a large number of cycles, redundant use of internal registers beyond 10,000 cycles is recommended.
7. Data shows a fairly constant E/W Programming behavior over temperature. In this sense E^2PROMs supersede EPROMs which are restricted to room temperature programming.
8. As shown in the timing diagrams, the start bit on DI must be set by a ZERO - ONE transition following a CS enable (ZERO - ONE), when executing any instruction. ONE CS enable transition can only execute ONE instruction.
9. In the read mode, following an instruction and data train, the DI can be a don't care, while the data is being outputted i.e., for next 17 bits or clocks. The same is true for other instructions after the instruction and data has been fed in.
10. The data-out train starts with a dummy bit 0 and is terminated by chip deselect. Any extra SK cycle after 16 bits is not essential. If CS is held on after all 16 of the data bits have been outputted, the DO will output the state of DI till another CS LO-HI transition starts a new instruction cycle.
11. When a common line is used for DI and DO, a probable overlap occurs between the last bit on DI and start bit on DO.
12. After a read cycle, the CS must be brought low for 1 SK clock cycle before another instruction cycle can start.

INSTRUCTION SET

Commands	Opcode	Comments
READ	10000A3A2A1A0	Read Register 0–15
WRITE	11000A3A2A1A0	Write Register 0–15
ERASE	10100A3A2A1A0	Erase Register 0–15
EWEN	111000 0 0 1	Write/Erase Enable
ENDS	111000 0 1 0	Write/Erase Disable
*** WRAL	111000 1 0 0	Write All Registers
ERAL	111000 1 0 1	Erase All Registers

All commands, data in, and data out are shifted in/out on rising edge of SK clock.

Write/erase is then done by pulsing CS low for 10 ms.

All instructions are initiated by a LO-HI transition on CS followed by a LO-HI transition on DI.

READ — After read command is shifted in DI becomes don't care and data can be read out on data out, starting with dummy bit zero.

WRITE — Write command shifted in followed by data in (16 bits) then CS pulsed low for 10 ms minimum.

ERASE
ERASE ALL — Command shifted in followed by
WRITE ALL — Pulsing CS low for 10 ms.
WRITE
ENABLE/DISABLE — Command shifted in.

*** (This Instruction is not speced on Data sheet.)

The following is a list of various systems that could use a NMC9306/COP494

A. Airline terminal
Alarm system
Analog switch network
Auto calibration system
Automobile odometer
Auto engine control
Avionics fire control

B. Bathroom scale
Blood analyzer
Bus interface

C. Cable T.V. tuner
CAD graphics
Calibration device
Calculator—user programmable
Camera system
Code identifier
Communications controller
Computer terminal
Control panel
Crystal oscillator

D. Data acquisition system
Data terminal

E. Electronic circuit breaker
Electronic DIP switch
Electronic potentiometer
Emissions analyzer
Encryption system
Energy management system

F. Flow computer
Frequency synthesizer
Fuel computer

G. Gas analyzer
Gasoline pump

H. Home energy management
Hotel lock

I. Industrial control
Instrumentation

J. Joulemeter

K. Keyboard -softkey

L. Laser machine tool

M. Machine control
Machine process control
Medical imaging
Memory bank selection
Message center control
Mobile telephone
Modem
Motion picture projector

N. Navigation receiver
Network system
Number comparison

O. Oilfield equipment

P. PABX
Patient monitoring
Plasma display driver
Postal scale
Process control
Programmable communications
Protocol converter

Q. Quiescent current meter

R. Radio tuner
Radar dectector
Refinery controller
Repeater
Repertory dialer

S. Secure communications system
Self diagnostic test equipment
Sona-Bouy
Spectral scanner
Spectrum analyzer

T. Telecommunications switching system
Teleconferencing system
Telephone dialing system
T.V. tuner
Terminal
Test equipment
Test system
TouchTone dialers
Traffic signal controller

U. Ultrasound diagnostics
Utility telemetering

V. Video games
Video tape system
Voice/data phone switch

W. Winchester disk controller

X. X-ray machine
Xenon lamp system

Y. YAG—laser controller

Z. Zone/perimeter alarm system

A Study of the Crystal Oscillator for CMOS-COPS™

National Semiconductor Corp.
Application Note 400
Abdul Aleaf

INTRODUCTION

The most important characteristic of CMOS-COPS is its low power consumption. This low power feature does not exist in TTL and NMOS systems which require the selection of low power IC's and external components to reduce power consumption.

The optimization of external components helps decrease the power consumption of CMOS-COPS based systems even more.

A major contributor to power consumption is the crystal oscillator circuitry.

Table I presents experimentally observed data which compares the current drain of a crystal oscillator vs. an external squarewave clock source.

The main purpose of this application note is to provide experimentally observed phenomena and discuss the selection of suitable oscillator circuits that cover the frequency range of the CMOS-COPS.

Table I clearly shows that an unoptimized crystal oscillator draws more current than an external squarewave clock. An RC oscillator draws even more current because of the slow rising signal at the CKI input.

Although there are few components involved in the design of the oscillator, several effects must be considered. If the requirement is only for a circuit at a standard frequency which starts up reliably regardless of precise frequency stability, power dissipation and etc., then the user could directly consult the data book and select a suitable circuit with proper components. If power consumption is a major requirement, then reading this application note might be helpful.

WHICH IS THE BEST OSCILLATOR CIRCUIT?

The Pierce Oscillator has many desirable characteristics. It provides a large output signal and drives the crystal at a low power level. The low power level leads to low power dissipation, especially at higher frequencies. The circuit has good short-term stability, good waveforms at the crystal, a frequency which is independent of power supply and temperature changes, low cost and usable at any frequency. As compared with other oscillator circuits, this circuit is not disturbed very much by connecting a scope probe at any point in the circuit, because it is a stable circuit and has low impedance. This makes it easier to monitor the circuit without any major disturbance. The Pierce oscillator has one disadvantage. The amplifier used in the circuit must have high gain to compensate for high gain losses in the circuitry surrounding the crystal.

TABLE I

A. Crystal oscillator vs. external squarewave COP410C change in current consumption as a function of frequency and voltage, chip held in reset, CKI is ÷4.
I = total power supply current drain (at V_{CC}).

Crystal

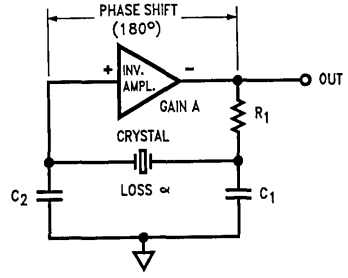
V_{CC}	f_{CKI}	Inst. cyc. time	I μA
2.4V	32 kHz	125 μs	8.5
5.0V	32 kHz	125 μs	83
2.4V	1 MHz	4 μs	199
5.0V	1 MHz	4 μs	360

External Squarewave

V_{CC}	f_{CKI}	Inst. cyc. time	I
2.4V	32 kHz	125 μs	4.4 μA
5.0V	32 kHz	125 μs	10 μA
2.4V	1 MHz	4 μs	127 μA
5.0V	1 MHz	4 μs	283 μA

WHAT IS A PIERCE OSCILLATOR?

The Pierce is a series resonant circuit, and its basic configuration is shown below.



TL/DD/8439–1

FIGURE 1

For oscillation to occur, the Barkhausen criteria must be met: (1) The loop gain must be greater than one. (2) The phase shift around the loop must be 360°.

Ideally, the inverting amplifier provides 180°, the R_1C_1 integration network provides a 90° phase lag, and the crystal's impedance which is a pure resistance at series resonance together with C_2 acts as a second integration network which provides another 90° phase lag. The time constants of the two RC phase shifting networks should be made as big as possible. This makes their phase shifts independent of any changes in resistance or capacitance values. However, big RC values introduce large gain losses and the selected amplifier should provide sufficient gain to satisfy gain requirement. CMOS inverters or discrete transistors can be used as amplifiers. An experimental evaluation of crystal oscillators using either type of amplifier is given within this report.

CRYSTAL OSCILLATORS USING CMOS-IC

The use of CMOS-IC's in crystal oscillators is quite popular. However, they are not perfect and could cause problems. The input characteristics of such IC's are good, but they are limited in their output drive capability.

The other disadvantage is the longer time delay in a CMOS-inverter as compared to a discrete transistor. The longer this time delay the more power will be dissipated. This time delay is also different among different manufacturers.

As a characteristic of most CMOS-IC's the frequency sensitivity to power supply voltage changes is high. As a group, IC's do not perform very well when compared with discrete transistor circuits.

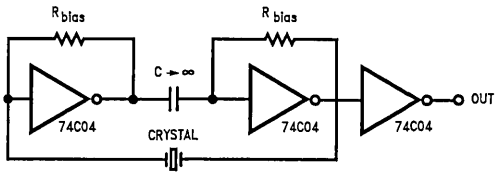
But let us not be discouraged. Low component count which leads to low cost is one good feature of IC oscillators.

As a rule, IC's work best at the low end of their frequency range and poorest at the high end.

Several types of crystal oscillators using CMOS-IC's have been found to work satisfactorily in some applications.

CMOS—TWO INVERTER OSCILLATOR

The two inverter circuit shown in *Figure 2* is a popular one. The circuit is series resonant and uses two cascaded inverters for an amplifier.



TL/DD/8439-2

FIGURE 2

Each inverter has a DC biasing resistor which biases the inverter halfway between the logic "1" and "0" states. This will help the inverters to amplify when the power is applied and the crystal will start oscillation.

The 74C family works better as compared with other CMOS-IC's. Will oscillate at a higher frequency and is less sensitive to temperature changes. The CMOS-COPS data sheet states that a crystal oscillator will typically draw 100 μA more than an external clock source. However, the crystal oscillator described above will draw approximately as much current as an external squarewave clock. The experimental data presented below shows the comparison:

Chip held in Reset, $V_{CC} = +5.0V$

f = 455 kHz, COP444C, CKI is ÷8

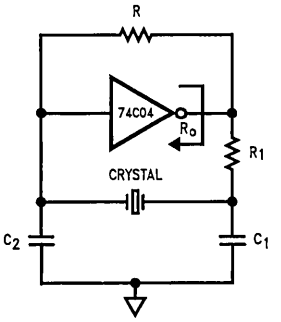
Instruction cycle time = 17.5 μs

I = total power supply (V_{CC}) current drain

Oscillator Type	I (current drain)
Crystal Osc. (data sheet)	950 μA
Crystal Osc. (two inverter)	810 μA
Ext. Clock	790 μA

PIERCE IC OSCILLATOR

Figure 3 shows a Pierce oscillator using CMOS inverter as an amplifier.



TL/DD/8439-3

FIGURE 3

The gain of CMOS inverter is low, so the resistor R_1 should be made small. This reduces gain losses. The output resistance of the inverter (Ro) can be the integrating resistor for the RoC_l phase lag network.

Omitting R_1 or with a small value of R_1, the crystal will be driven at a much higher voltage level. This will increase power dissipation.

For lower frequencies (i.e., 32 kHz), R_1 must be large enough so that the inverter won't overdrive the crystal. Also, if R_1 is too large we won't get an adequate signal back at the inverter's input to maintain oscillation. With large values of R_1 the inverter will remain in its linear region longer and will cause more power dissipation. Typically for 32 kHz, R_1 should be constrained by the relation.

$$\frac{1}{2\pi R_1 C_1} \ll 32\ \text{kHz}$$

At higher frequencies, selection of R_1 is again critical. In order to drive a heavy load at high frequency, the amplifier output impedance must be low. In order to isolate the oscillator output from C_1 so it can drive the following logic stages, then R_1 should be large. But again, R_1 must not be too large, otherwise it will reduce the loop gain.

The value of R_1 is chosen to be roughly equal to the capacitive reactance of C_1 at the frequency of operation, or the value of load impedance Z_L.

Where $Z_L = \frac{X_{c1}^2}{R_L}$

$R_L = R_S$ = series resistance of crystal

The small values of C_1 and C_2 will help minimize the gain reduction they introduce.

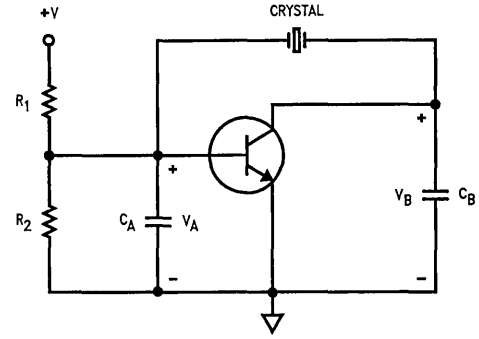
typically: $C_1 = C_2 = 220$ pF at 1 MHz

$C_1 = C_2 = 330$ pF at 2 MHz

DISCRETE TRANSISTOR OSCILLATOR

As mentioned earlier, a discrete transistor circuit performs better than an IC circuit. The reason for this is that in a discrete transistor circuit it is easier to control the crystal's source and load resistances, the gain and signal amplitude.

A discrete transistor circuit has shorter time delay, because it uses one or two transistors. This time delay should always be minimized, since it causes more power dissipation and shifts frequency with temperature changes. *Figure 4* shows a basic Pierce oscillator using a transistor as an amplifier.



TL/DD/8439-4

FIGURE 4

The basic phase shift network consists of C_{A_1}, C_{B_2} and the crystal which looks inductive and is series resonant with C_{A_1} and C_{B_1}. The phase shift through the transistor is 180° and the total phase shift around the loop is 360°. The condition of a unity loop gain must also be satisfied.

$$\frac{V_A}{V_B} = -\left(\frac{C_B}{C_A}\right)$$

$$\frac{V_A}{V_B} = -\left(\frac{X_{CA}}{X_{CB}}\right)$$

For oscillation to occur, the transistor gain must satisfy the relation

$$G\left(\frac{V_A}{V_B}\right) \geq 1$$

where $G = -g_{fe}Z_L$

g_{fe} is the transconductance of the transistor

Z_L is the load seen by the collector

$$Z_L = \frac{X_B^2}{Re}, \quad X_B = -\frac{1}{WC_B}$$

Re is the crystal's effective series resistance.

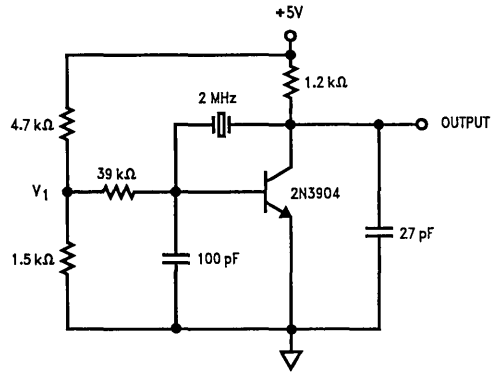
The crystal's drive level

$$P_d = \frac{V_{B2}Re}{X_{B2}}$$

This drive level should not exceed the manufacturer's spec.

Certain biasing conditions might cause collector saturation. Collector saturation increases oscillator's dependence on the supply voltage and should be avoided.

The circuit of *Figure 5* has been tested and has a very good performance.



TL/DD/8439-5

FIGURE 5

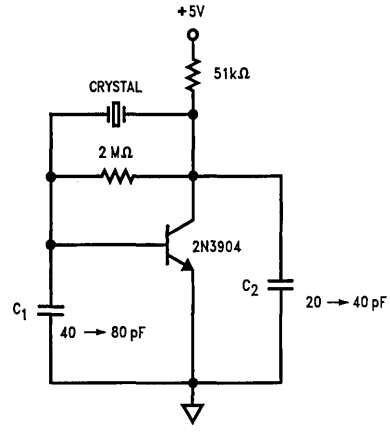
This circuit will oscillate over a wide range of frequencies 2–20 MHz.

Voltage $(V_1) = \frac{(5)(1.5)}{1.5 + 4.7} = 1.21V$

Base Current $= \frac{1.21\text{-}V_{BE}}{39k} = 15.6\ \mu A$

At Saturation ($V_{CE} = 0$)

$I_{C\,(SAT)} = \frac{5}{1.2} = 4.2$ mA



TL/DD/8439-6

FIGURE 6

Having 15.6 μA of base current, for saturation to occur

$$h_{FE} = \frac{4.2\ mA}{15.6\ \mu A} = 269$$

The DC beta for 3904 at 1 mA is 70 to 210, so no problem with saturation, even at lower supply voltages.

The current consumption (power supply V_{CC} current drain) of COP444C using the above oscillation circuit is around 267 μA.

The circuit of *Figure 6* is another configuration of discrete transistor oscillator.

The performance of above circuit is also good. The only drawback is that it does not provide larger output signal.

CONCLUSION

As discussed within this report, a discrete transistor circuit gives better performance than an IC circuit. However, oscillators using discrete transistors are more expensive than those using IC's when assembly labor costs are included. So, the selection of either circuit is a trade-off between better performance and cost.

The data and circuits presented here are intended to be used only as a guide for the designer. The networks described are generally simple and inexpensive and have all been observed to be functional. They only provide greater flexibility in the oscillator selection for CMOS-COPS systems.

Selecting Input/Output Options On COPS™ Microcontrollers

National Semiconductor Corp.
Application Note 401
Abdul Aleaf

INTRODUCTION

There are a variety of user selectable input and output options available on COPS when the ROM is masked. These options are available to help the user tailor the I/O characteristics of the Microcontroller to the application. This application note is intended to provide the user a guide to the options: What are they? When and how to use which ones? The paper is generally written without reference to a specific device except when examples are given. It must be remembered that any given generic COPS Microcontroller has a subset of all the possible options available and that a given pin might not have all possible options. A reference to the device data sheet will determine which options are available for a specific device and a specific pin of that device.

INPUT/OUTPUT OPTIONS

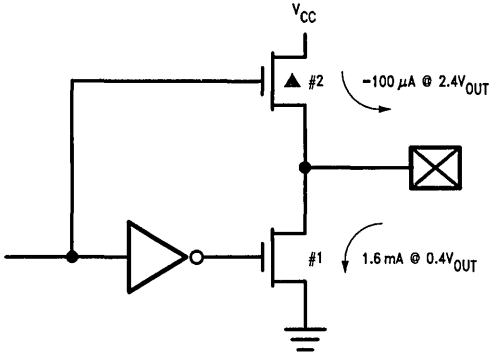
Table I summarizes the I/O capability of NMOS-COPS, in general. However, some of the options have different configuration in CMOS-COPS. Data sheets provide information on the I/O options associated with the CMOS-COPS.

I. OUTPUTS

The following discussion provides detailed information on the capabilities of the mask-programmable output options available on COPS.

A. STANDARD OUTPUT

This option is a simple, straightforward, logic compatible output used for simple logic interface. It is available on SO, SK and all D and G outputs, It is recommended to be used as a default option for all but SO, SK outputs.

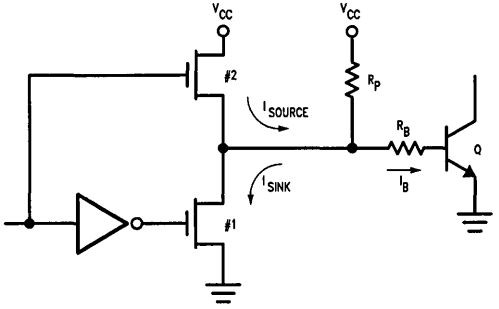


TL/DD/8440–1

FIGURE 1. Standard Output

Figure 1 shows the standard output configuration. The enhancement mode device to ground is good at sinking current (sinks 1–2 mA) and is compatible with the sinking requirement of 1 TTL load (1.6 mA at 0.4V). It will meet the "low" voltage requirement of CMOS logic. All output options use this device (device #1) for current sinking. On the other hand, the relatively high impedance depletion-mode device (device #2) to V_{CC} provides low current sourcing capability (100 μA at 2.4V). This pullup is sufficient to provide the source current for a TTL high level and will go to V_{CC} to meet the "high" voltage requirements of CMOS logic. An external resistor to V_{CC} may be required to interface to other external devices requiring higher sourcing capability.

An interface example to a common emitter NPN transistor is given below:



TL/DD/8440–2

FIGURE 2

R_B is needed to limit transistor's base current if $I_{source} > I_{B(max)}$.

R_p helps generate base drive if the I_{source} is not sufficient. The disadvantage of R_p is the introduction of more power dissipation. The temperature effects on the reverse saturation current I_{CBO} causes I_C to shift. I_{CBO} approximately doubles for every 10°C temperature rise. The effect of changes in I_{CBO} reduces off state margin and increases power dissipation in the off state.

However, in a typical device, the current supplied by R_p will swamp out any effects on I_{CBO}. Another parameter found to be decreasing linearly with temperature is V_{BE}:

$$\Delta V_{BE} = V_{BE_2} - V_{BE_1} = -k(T_2 - T_1)$$

where $k \approx 2$ mV/°C, T in °C.

Now let's consider a practical example:

LOW SOURCE CURRENT OUTPUT:

Standard output, COP420, device #2.

The selected transistor is 2N3904.

DESIGN CONSIDERATIONS:

a. Q is in saturation during ON-state.

b. Q's collector current I_C = 100 mA

TABLE I

	Default	Standard	Push-Pull	High Sink	Very High Sink	LED	Hi-Current LED	TRI-STATE® Push-Pull	Hi Current TRI-STATE Push-Pull	Open Drain
SO	Push-Pull	Logic Compatible; Non MICROWIRE™	MICROWIRE Higher Drive, Faster X'sition							External Pull Up
SK	Push-Pull	Logic Compatible; Non MICROWIRE	MICROWIRE Higher Drive Faster Transition							External Pull Up
D	Standard	Logic Compatible		L Parts Only 15 mA	L Parts Only 30 mA					External Pull Up, Standard, Hi Sink or V.H.S. Pull Down
G	Standard	Logic Compatible; Inputs		L Parts Only 15 mA	L Parts Only 30 mA					External Pull-Up, Standard, Hi Sink or V.H.S. Pull Down
L	Standard	Logic Compatible; Inputs, TRI-LEVEL				Hi Source 1.5 mA TRI-LEVEL	L Parts Only Higher Source 3 mA TRI-LEVEL	MICROBUS™ Meets TRI-STATE Spec. TRI-LEVEL	L Parts Only Meets TRI-STATE Spec. TRI-LEVEL	External Pull Up TRI-LEVEL
H	Standard	Logic Compatible Inputs								External Pull Up
R	Standard	Logic Compatible; Inputs, TRI-LEVEL	Higher Drive Faster Transition TRI-LEVEL					Meets TRI-STATE Spec TRI-LEVEL		External Pull Up TRI-LEVEL

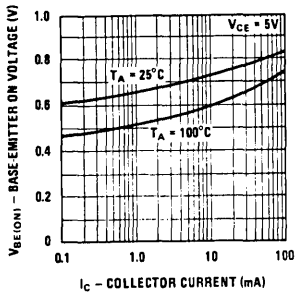
c. Assuming a "forced" of 10 for Q. This is a standard value for β to insure saturation.

For an I_C = 100 mA, β = 10, we have $I_B \geq 10$ mA. The low current standard output certainly cannot provide $I_B \geq 10$ mA. Therefore, a pullup resistor (R_p) is required.

d. Now we need to select the minimum allowed value for R_p. The sinking ability of COPS output will determine R_p. We must sink the pullup current to a $V_{OUT} < V_{BE}$ in order to hold Q off. Also, note that

$$\frac{\Delta V_{BE}}{\Delta T} = -2 \text{ mV/°C}.$$

e. Assuming the worst case is at V_{CC} (max) and High-temperature (let $\Delta T = 20°C \Rightarrow \Delta V_{BE} = -40$ mV). From $V_{BE(ON)}$ Vs. I_C curve, *Figure 3*:



TL/DD/8440-3

FIGURE 3. 2N3904 I/V

at 100 mA, 25°C, $V_{BE} \cong 0.85$V.

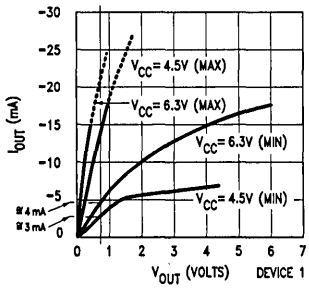
So, our $V_{BE(45°C)} = 0.85 - 0.04 \cong 0.81$V.

There is not margin here for process V_{BE} variations so we can allow 200 mV of slope,

V_{BE} = 0.61V (worst case)

f. Having V_{BE} = 0.61V, we go to COPS sink graph and draw a vertical line at $V_{OUT} = V_{BE} = 0.61$V. *Figure 4* below:

Output Sink Current



TL/DD/8440-4

FIGURE 4

This will tell us, at $V_{out} = V_{BE}$, how much current can be sinked to keep Q "OFF". The intersection of V_{CC} = 6.3 (MIN) and V_{BE} = 0.61V gives us I_{sink} = 4 mA.

g. Now calculate R_p.

$$R_p \geq \frac{6.3 - 0.61}{4} \text{k} \geq 1.42\text{k}$$

$$\text{the actual standard } R_p (\pm 10\%) = \frac{1.42}{0.9}$$

$$= 1.6\text{k} \pm 10\%$$

h. Using the value of R_p, let's calculate the current through R_p at V_{CC} = 4.5V(MIN).

$$I_{RP} = \frac{4.5 - 0.61}{1.42} \text{mA} = 2.74 \text{ mA}$$

Which is less than sink ability of device (3 mA from *Figure 4*) at V_{CC} = 4.5V, V_{out} = 0.61V.

i. Now calculate the available source current. Here we use $V_{BE(max)}$ which is the worst case, and low temperature.

Let T (ambient) = 10°C.

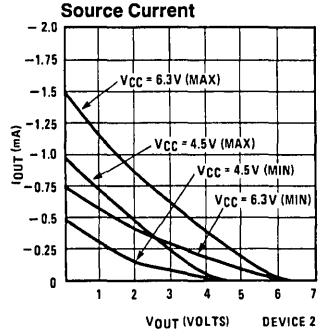
From V_{BE} vs. I_C curve, *Figure 3*:

$V_{BE} \cong 0.83$V at 25°C

$V_{BE} \cong 0.83 + 2 \text{ mv/°C} \times 15 = 0.86$V at 10°C.

Using this value of V_{BE}, we go to COP420 Standard Output source current curve (*Figure 5*), and draw a vertical line at V_{BE} = 0.86V. The intersection of this line and V_{CC} = 4.5(MIN) gives an I_{source} = 325 μA.

Standard Output Source Current



TL/DD/8440-5

FIGURE 5

This is low but typical of N-channel low current standard output.

Contribution of R_p

$$I_{RP} = \frac{4.5 - 0.86}{\underbrace{(1.6)\,(1.1)}_{R_{p(max)}}} = 2.07 \text{ mA}$$

$I_B(min) \cong 2.07 + 0.325 = 2.3$ mA

This is our worst case base drive, but we needed 10 mA.

What can we do to get the base drive we need?

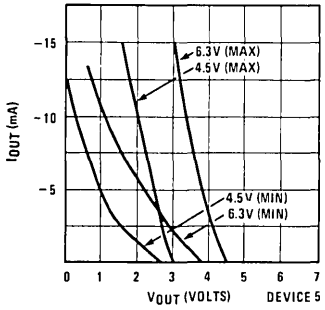
1. We can use above design and allow Q to come out of saturation. The disadvantage is that Q's power dissipation increases.

2. Or use a Darlington configuration (Process 05). In such a configuration only first stage of Darlington can be saturated (not output stage). This will introduce a slightly higher power dissipation. Note that for a process 05 transistor, the forced β is 1000.

3. Use a high source type output such as TRI-STATE output. If we draw a vertical line at V_{BE} = 0.86, we get a source current of ≅ 6 mA at V_{CC} = 4.5(MIN) *Figure 6*, which gives us a worst case

$$I_{B(min)} = 8.07 \text{ mA.}$$

TRI-STATE Output Source Current



TL/DD/8440-6

FIGURE 6

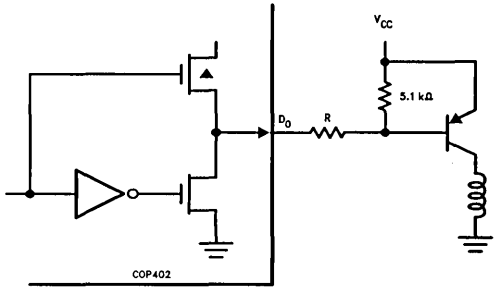
CAUTION! On TRI-STATE graph the intersection of $V_{out} = B_{BE}$ = 0.86V and V_{CC} = 6.3V(MAX) curve (*Figure 6*) would result in an $I_{B(Max)}$ = 50–60 mA, which is way too much to handle. In this case there is a need for a series current limiting R_B to kill some of the worst case $I_{B(max)}$.

4. There is a high current Standard-L option on some COPS (i.e., COP4XL, L-port) which provides sufficient source current.

5. N-channel output can generally sink better than source. PNP transistor can be used instead of NPN. The same analysis applies and in general will show better overdirve capabilities.

As shown in *Figure 7*, the D_0 output which has a standard output option, is driving the base of the PNP transistor. Assuming V_{CC} = 4.5V (for COP402), V_{BE} = 1.0V, and a worst case base drive requirement of 3.0 mA. We see that we must supply 200 μA to the base-emitter resistor to turn the transistor on:

$$1.0V/5.1k = 200\ \mu A$$



TL/DD/8440-7

FIGURE 7. PNP Drive

From the output sink current curve on the COP402 data sheet, we find that, at 1.0V the D-line can sink 3.2 mA. To calculate the value of the current limiting resistor,

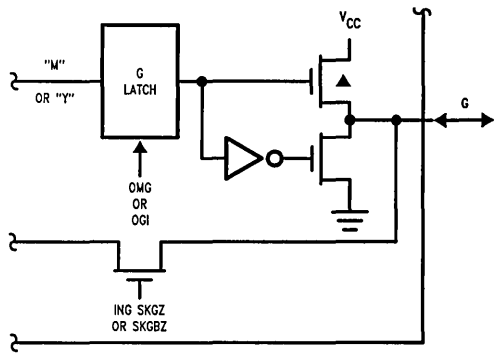
$$R = (V_{CC} - V_{BE} - V_{D0})/I$$

When V_{CC} = 6.3V, the D0 output can sink more than enough current at 0.3V, and if the V_{BE} = 0.7V, we can calculate the maximum D_0 output current:

$$I = (V_{CC} - V_{BE} - V_D)/R$$
$$= (6.3 - 0.7 - 0.3)/780 = 6.3 \text{ mA.}$$

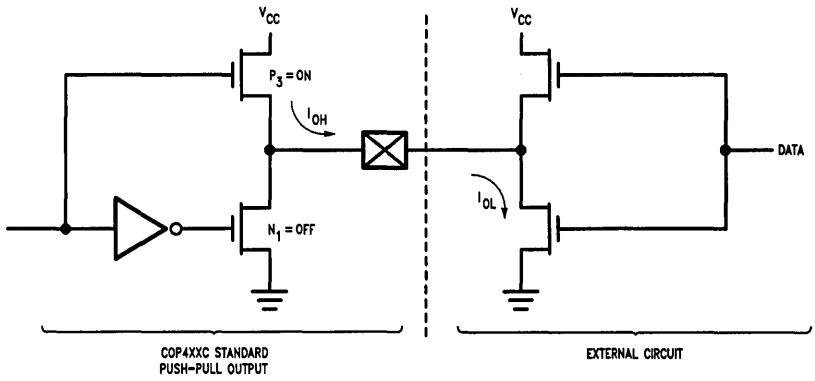
Using the Standard Output Option for Bidirectional I/O (G-port)

The standard output is good at sinking current, but rather weak at sourcing it. Therefore, by using the Standard Drive configuration and outputting 1's to the port, an external source may easily overdrive the port drivers with the added bonus of a built-in pullup. While the depletion-mode device provides sufficient current for a TTL high level, yet can be pulled low by an external source, thus allowing the same pin to be used as an input and output. Data written to the ports is statically latched and remains unchanged until rewritten. As inputs the lines are non-latching (*Figure 8*).



TL/DD/8440-8

FIGURE 8. G Port Characteristics



TL/DD/8440-9

FIGURE 9

When writing a "0" to the port, the enhancement-mode device to ground overcomes the high pullup and provides TTL current sinking capability. While writing a "1" the depletion-mode device behaves as internal pullup maintaining the "1" level indefinitely. In this situation, an input device capable of overriding the small amount of current supplied by the pull-up device can be read. This feature provides maximum user flexibility in selecting input/output lines with minimum external components.

In CMOS-COPS the low current push-pull output has even much weaker source current capability and this make it easier to be overriden.

Referring to *Figure 9*.

Note that $I_{OL} > I_{OH}$, otherwise transistors or buffers must be used.

For COP424C/444C, standard push-pull

@ $V_{CC} = 4.5V$, $V_{out} = 0V$, $I_{OH(min)} = 30\ \mu A$

$I_{OH(max)} = 330\ \mu A$

@ $V_{CC} = 2.4V$, $V_{out} = 0V$, $I_{OH(min)} = 6\ \mu A$

$I_{OH(max)} = 80\ \mu A$

While in NMOS (COP420L), Standard output:

@ $V_{CC} = 4.5V$, $V_{OH} = 2.0V$, $I_{OH(min)} = 30\ \mu A$

$I_{OH(max)} = 250\ \mu A$

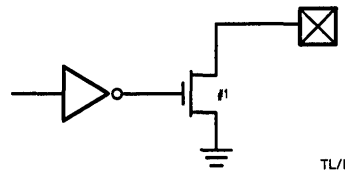
@ $V_{CC} = 6.3V$, $V_{OH} = 2.0V$, $I_{OH(min)} = 75\ \mu A$

$I_{OH(max)} = 480\ \mu A$

As we see, both in CMOS and NMOS it is easier to override I_{OH}. Note that the standard output option is available with standard, high, or very high sink current capability ("L" parts only). The pulldown device is bigger for the high/very high current standard output. The sourcing current is the same. These three choices provide some control over current capability.

B. OPEN-DRAIN OUTPUT

This option uses the same enhancement-mode device to ground as the standard output with the same current sinking capability. It does not contain a load device to V_{CC}, allowing external pullup as required by the user's application. The sinking ability of device #1 determines the minimum allowed external pullup. The analysis discussed earlier for Standard Output options equally applies here. Available on SO, SK, and all D, G, and L outputs.



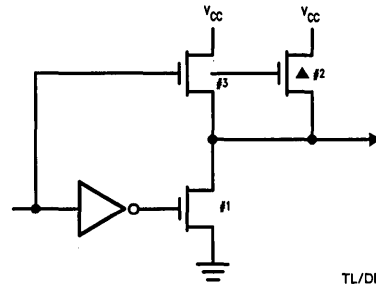
TL/DD/8440-10

FIGURE 10. Open-Drain Output

The open-drain option makes the ports G and L very easy to drive when they are used as inputs. This option is commonly used for high noise margin inputs, unusual logic level inputs as from a diode isolated keyboard, analog channel expansion, and direct capacitive touch-panel interface. Available with standard, high or very high sink capability ("L" parts only).

C. PUSH-PULL OUTPUT

The push-pull output differs from the standard output configuration in having an enhancement-mode device in parallel with the depletion-load device to V_{CC}, providing greater current sourcing capability (better drive) and faster rise and fall times when driving capacitive loads.



TL/DD/8440-11

FIGURE 11. Push-Pull Output

If a push-pull output is interfaced to an external transistor, a current-limiting resistor must be placed in series with the base of the transistor to avoid excessive source current flow out of the push-pull output. This option is generally for MICROWIRE Serial Data exchange.

It is available on SO, SK only and is recommended to be used as a default option for these outputs. A few points must be kept in mind when using SO, SK for MICROWIRE interface.

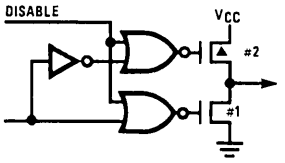
The data sheet specifies the propagation delay for a certain test condition (i.e., V_{CC} = 5V, V_{OH} = 0.4V, Loading = 50 pF, etc.).

In practice, actual delay varies according to actual input capacitive loading (typical 7–10 pF per IC input), total wire capacitance and PCB stray capacitance connected to the SI input. Thus, if actual total capacitive loading is too large to satisfy the delay time relationships ($t_d = t_{SK} - t_r$; t_d = actual delay time, t_{Sk} = the instruction cycle time, t_r = the finite SK rise time), either slow down SK cycle time or add a pullup resistor to speed up SK "0" to "1" transition or use an external buffer to drive the large load. Besides the timing requirement, system supply and fan-out/fan-in requirements have to be considered, too.

If devices of different types are connected to the same serial interface, the output driver of the controller must satisfy all the input requirements of each device. Briefly, for devices that have incompatible input levels or source/sink requirements to exchange data, external pullups or buffers are necessary to provide level shifting or driving. Unreliable operation might occur during data transfer, otherwise. For a 100 pF load, a standard COPS Microcontroller may use a 4.7k external resistor, with the output "low" level increased by less than 0.2V. For the same load the low power COPS may use a 22k resistor; with the SO, SK output "low" level increased by less than 0.1V.

D. STANDARD L OUTPUT

Same as Standard Output, but may be disabled. Available on L-outputs only.



TL/DD/8440–12

FIGURE 12. Standard L Output

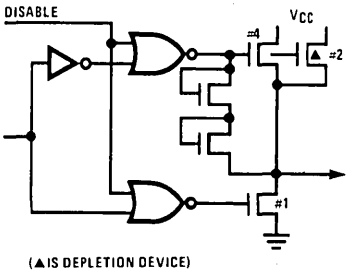
When this option is implemented on the L-port and the L-drivers are disabled to use the L lines as inputs, the disabled depletion-mode device cannot be relied on to source sufficient current to pull an input to a logic high. There are two ways to use L lines as inputs (having standard L option):

The first method requires that the drivers be disabled. In this case the lines are floating in an undefined state. The external circuitry must provide good logic levels both high and low to the input pins. The inputs are then read by the INL instruction. The second method is similar to the technique used for the G-port. The drivers are enabled and a "1" must be written to the Q register.

The external circuitry will then be required only to pull the lines low to a logic "0". The line will pull up to a "1" itself. The INL instruction is used as before to read the lines.

E. LED DIRECT DRIVE OUTPUT

In this configuration, the depletion-load device to V_{CC} is paralleled by an enhancement-mode device to V_{CC} to allow for the greater current sourcing capacity required by the segments of an LED display. Source current is clamped to prevent excessive source current flow.



TL/DD/8440–13

FIGURE 13. LED (L output) NMOS-COPS

This configuration can be disabled under program control by resetting bit 2 (EN^2) of the enable register to provide simplified display segment blanking.

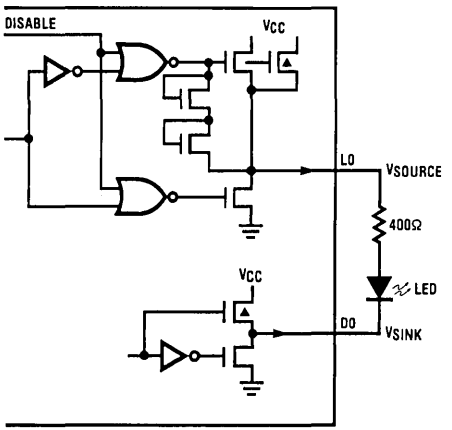
However, while both enhancement-mode devices are turned off in the disabled mode, the depletion-load device to V_{CC} will still source up to 0.125 mA. As in the case of Standard L output, again this current is not sufficient to pull an input to a logic "1".

The drivers must be disabled and the lines must be pulled high and low externally, whenever they are used as inputs.

Example #1:

When COPS outputs are used to drive loads directly, the power consumed in the outputs must be considered in the maximum power dissipation of the package.

Figure 14 shows an LED segment obtaining its source current from L_0 output and D_0 sinking the current. In this configuration all the power required to drive the LED with the exception of the portion consumed by the LED itself, is consumed within the chip. Assuming COP404L is the driving device:



TL/DD/8440–14

FIGURE 14. LED Drive

If we assume the V_{source} is not inserted, the device has a V_{CC} of 9.5V, and that the voltage drop across the LED is 2.0V.

We can calculate the power dissipation in these outputs. The minimum current that D_0 can sink at 1.0V is 35 mA (COP404L data sheet). L_0 can source up to 35 mA at 3.0V. Therefore, the power dissipation for the L_0 output could be: (9.5 −3.0) (0.035) = 227 mW. The power in the D_0 output is (1)(0.035) = 35 mW.

Now let us calculate the current limiting resistor. Referring to COP404L L_0–L_7 output source current curves, at V_{CC} =9.5V the minimum current curve peaks at I = 6.0 mA and V_{source} = 4.8V. The current curve is actually very flat between 4.0 and 5.0 volts. For maximum current, we need to set the voltage on the L pin equal to 4.8V at 6.0 mA. The D line will sink this current at 0.4V. Therefore, the resistor and LED must make up the difference:

$$V_I = V_D + IR + V_{LED}$$
$$4.8 = 0.4 + 0.006R + 2.0$$
$$R = 400\Omega$$

At the other end of the curve, when the L line sources the maximum current, assume the LED and the D line will have the same voltage drop.

$$V_I = 0.4 + IR + 2.0$$
$$V_I = 2.4 + IR$$

From the current curve, we see that at 6.4V the L line will source 10 mA. Therefore: V_I = 2.4 + (0.01) (400) = 6.4V.

Example #2:

Let's consider a different configuration.

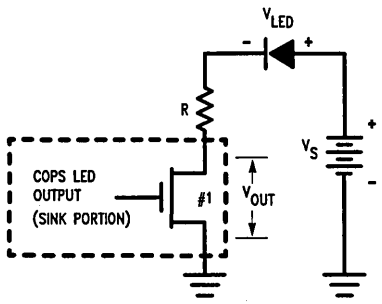


FIGURE 15. LED DRIVE

Now we calculate the series current limiting resistor R. The circuit has two non-linear devices to be considered; the output device and the LED.

The LED in this example is NSC5050. Looking at I/V curve, the device has a threshold 1.6V. Also, note that for V_{LED} > 1.6V the I/V curve is very linear (*Figure 17*).

Because of this, the LED characteristic can be modeled as a sharp threshold device with a non-zero source resistance (normally I/V curve is LOG looking).

From ON part of curve,

$$R_S = \frac{1.9 - 1.7}{0.05} = 4\Omega$$

We can neglect R_S as well (only $R_S \ll R$). Our model is simply a voltage source for the LED when

$I = 0$ for $V_{LED} < V_{TH}$

$I = \infty$ for $V_{LED} > V_{TH}$

Design Procedure:

1. $$I_{LED(min)} = \frac{V_{S(min)} - (V_{LED(max)} + V_{OUT(max)})}{R(max)}$$

We need endpoints of the load line.

a. @V_{out} = 0 $\Rightarrow$ $$I_{LED(min)} = \frac{V_{S(min)} - V_{LED(max)}}{R(max)}$$

b. @$V_{out} + V_{LED(max)} = V_S \Rightarrow I = 0$
($V_{LED(max)}$ = 2V)

2. Plot a and b

Assuming an I_{min} = 7 mA, $V_{S(min)}$ = 4.5V
from 1 $R_{(max)}$ = 357Ω

Draw the load line with slope −1/357 crossing
$V_{out} = V_S - V_{LED(max)}$ = 4.5 − 2 = 2.5V.
(*Figure 16*).

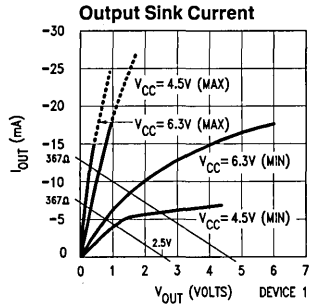
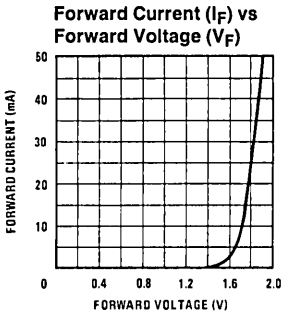


FIGURE 16. COP420



TL/DD/8440-17

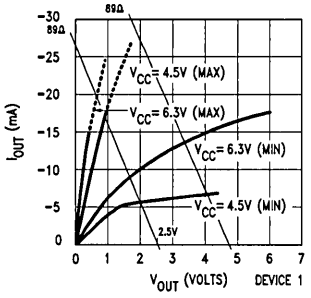
FIGURE 17. LED I/V Characteristic

The intersection of this load line and V_{CC} = 4.5V (min) curve, we find an actual value of $I_{(min)}$ = 4.25 mA.

To determine I_{max} (at R = 357Ω) we draw a parallel load line intersecting V_{out} = 6.3 − 2.0 = 4.3V and find that @ V_{CC} = 6.3V, $I_{(max)}$ = 13 mA.

3. From above calculations we observe that our $I_{(min)}$ (actual) is way off. Let's try to rotate our first load line around V_{out} = 2.5V to increase I_{min} and then check I_{max} and R. (*Figure 18*).

Let's go for an I_{min} (actual) = 6 mA. This will give us R = 89Ω and the max. plot goes off the graph to = 36 mA.



TL/DD/8440-18

FIGURE 18. COP420

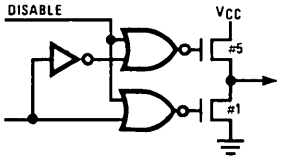
Comments:

1. The design must be a compromise between the two extremes (battery life should also be considered).

2. The lower the LED threshold the better. (The load line moves further up the device curve.)

F. TRI-STATE PUSH-PULL OUTPUT

This option is specifically available to meet the specifications of National's MICROBUS, outputting data over the data bus to a host CPU. It has two enhancement-mode devices to ground and V_{CC}.



TL/DD/8440-19

FIGURE 19. TRI-STATE Push Pull (L output)

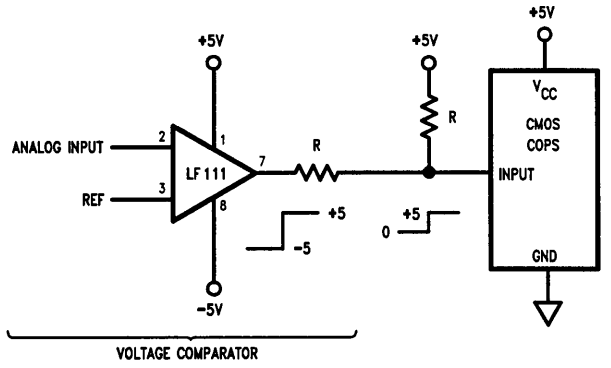
The TRI-STATE logic can disable both enhancement-mode devices to free the MICROBUS data lines for input operation.

CAUTION! Never try to pull against the TRI-STATE Output (too much source current) with the drivers enabled and Q register previously loaded with "1". The choices we have are mentioned earlier. Either TRI-STATE L-port or use Standard L output option.

II. INPUTS

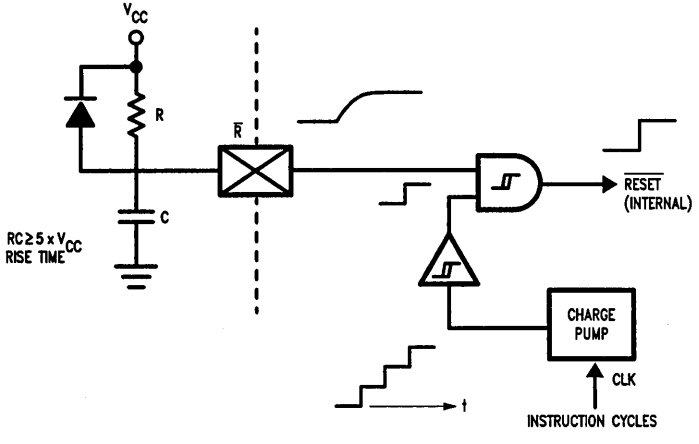
COPS inputs may be programmed either with a depletion load device to V_{CC} or floating (Hi-Z input). All inputs are TTL/CMOS compatible. Hi-Z inputs should not be left floating; they should be connected to the output of a "high" and "low" driving device if active or to V_{CC} and ground if unused. Especially when using CMOS COPS (very high impedance inputs), the open inputs can float to any voltage. This will cause incorrect logic function and more power dissipation. Also, the CMOS inputs are more susceptible to static charge which causes gate oxide rupture and destroys the device. Unlike inputs, the outputs should be left open to allow the output switch without drawing any DC current. Another precaution is powering up the device. Never apply power to inputs or TRI-STATE outputs before both V_{CC} and ground are connected. This will forward bias input protection diodes, causing excessive diode currents. It will also power the device.

Special care must be practiced when interfacing a CMOS-COPS input to an analog IC, powered by different supply voltages. Avoid overvoltage conditions resulting



TL/DD/8440-20

FIGURE 20



TL/DD/8440-21

FIGURE 21

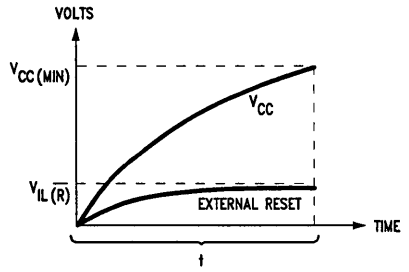
from such situations. As an example, consider the interface of a CMOS-COPS with the LF111 voltage comparator:

When the low level "−5V" appears on the comparator's output, the COPS input is pulled low below "logic low" of "0V". This will cause damage if the comparator sinks enough current. The use of a current-limiting resistor in series with the input is helpful. A better solution is to use a voltage divider as shown in *Figure 20*. Any time a low level appears on the comparator's output, a total voltage drop of 10V will appear across both resistors each dropping 5V, causing the input to sit at 0V. Whenever the output goes high, the resistors will not drop any voltage (no current through the resistors) and a logic high of 5V will appear on the input. To reduce power dissipation introduced by resistors, the resistor value must be high (>100k), because the CMOS inputs have very high input impedance.

$\overline{\text{RESET}}$ INPUT

All COPS Microcontroller have internal reset circuitry. Internally there is an AND gate with one input coming from the $\overline{\text{RESET}}$ input, and the second input connected to a charge pump circuitry. In the Charge pump circuit, a tiny capacitor is being charged upon execution of each internal instruction cycle. When the voltage across this internal capacitor reaches a high logic level, the second input of the AND gate is released.

The Reset logic will initialize (clear) the device upon power-up if the power supply rise time is less than 1 ms and greater than 1 μs. With a slowly rising power supply, the part may start running before V_{CC} is within the guaranteed range. In this case, the user must provide an external RC network and diode shown in *Figure 21* above. The external RC network is there to hold the $\overline{\text{RESET}}$ pin below V_{IL} until V_{CC} reaches at least $V_{CC(min)}$. The desired response is shown in *Figure 22*.



TL/DD/8440-22

FIGURE 22

t = 500-600 instruction cycles (8 msec)
for COPxxxL

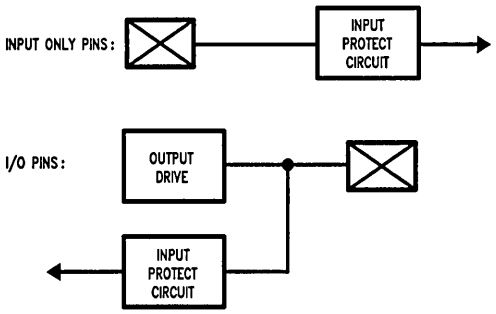
t = 900–1000 instruction cycles (4 msec)
for COPxxxC

The diode is included in the reset circuitry to cause a "forced Reset" when the power supply goes away and recovers quickly. In such a situation the diode helps discharge the capacitor quickly. Otherwise, if the power failure occurs for a short time, the capacitor will not be fully discharged and the chip will continue operation with incorrect data.

Note that on the CMOS COPS, the internal charge pump circuitry can be disabled when using a very slow clock (<32 kHz) [option 23 = 1]. This is necessary, because one can run from DC to 4 μs instruction cycle time (fully static). In such a situation external RC network discussed earlier must be used.

INPUT PROTECTION DEVICES

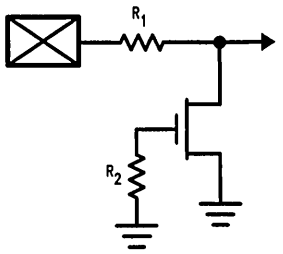
All inputs and I/O pins have input protection circuitry. This circuitry is there regardless of any option selected. It is the first circuitry encountered at the pin.



TL/DD/8440-23

FIGURE 23

For NMOS and XMOS devices, the circuits are of the form:

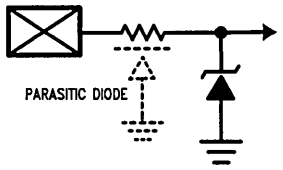


TL/DD/8440-24

FIGURE 24

This is a standard circuit defined for the process. R_1 is on the order of 200Ω. R_2 is around 300Ω (note that the R values are not precise).

This circuit is functionally equivalent to:



TL/DD/8440-25

FIGURE 25

The zener breakdown is around 10–15V; the gate breakdown is 50V.

CONCLUSION

All COPS Microcontrollers have a number of I/O options necessary to implement dedicated control functions in a wide variety of applications. The flexibility to select different options allows the user to tailor within limits, the I/O characteristics of the Microcontroller to the system. Thus, the user can optimize COPS for the system, thereby achieving maximum capability and minimum cost. This application note deals with the basic functionality of COPS I/O characteristics and does not address electrical differences among the various COPS devices.

MICROWIRE™ Serial Interface

National Semiconductor Corp.
Application Note 452
Abdul Aleaf

INTRODUCTION

MICROWIRE is a simple three-wire serial communications interface. Built into COPS™, this standardized protocol handles serial communications between controller and peripheral devices. In this application note are some clarifications of MICROWIRE logical operation and of hardware and software considerations.

LOGICAL OPERATION

The MICROWIRE interface is essentially the serial I/O port on COPS microcontrollers, the SIO register in the shift register mode. SI is the shift register input, the serial input line to the microcontroller. SO is the shift register output, the serial output line to the peripherals. SK is the serial clock; data is clocked into or out of peripheral devices with this clock.

It is important to examine the logical diagram of the SIO and SK circuitry to fully understand the operation of this I/O port *(Figure 1)*.

The output at SK is a function of SYNC, EN0, CARRY, and the XAS instruction. If CARRY had been set and propagated to the SKL latch by the execution of an XAS instruction, SYNC is enabled to SK and can only be overridden by EN0 *(Figure 2)*. Trouble could arise if the user changes the state of EN0 without paying close attention to the state of the latch in the SK circuit.

If the latch is set to a logical high and the SIO register enabled as a binary counter, SK is driven high. From this state, if the SIO register is enabled as a serial shift register, SK will output the SYNC pulse immediately, without any intervening XAS instruction.

The SK clock (SYNC pulse) can be terminated by issuing an XAS instruction with CARRY = 0 *(Figure 3)*.

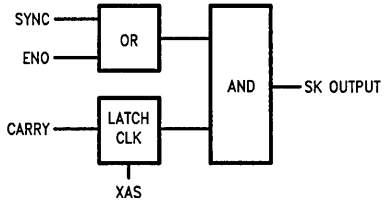


FIGURE 1. Logical Diagram of SK Circuit

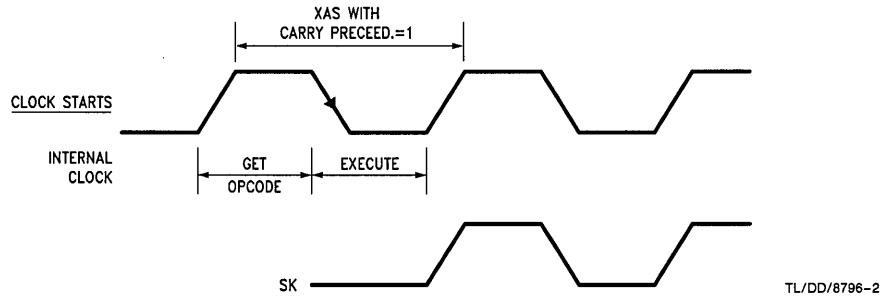


FIGURE 2. SK Clock Starts

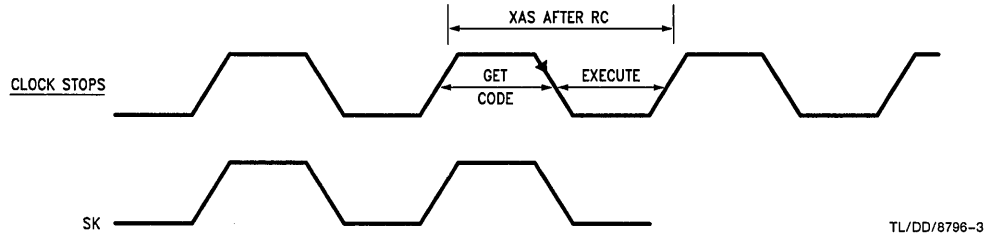


FIGURE 3. SK Clock Stops

The SIO register can be compared to four master-slave flip-flops shown in *Figure 4*. The masters are clocked by the rising edges of the internal clock. The slaves are clocked by the falling edges of the internal clock. Upon execution of an XAS, the outputs of the masters are exchanged with the contents of the accumulator (read and overdrive) in such a way that the new data are present at the inputs of the four slaves when the falling edge of the internal clock occurs. The content of the accumulator is, therefore, latched respectively in the four slave flip-flops and bit 3 appears directly on SO.

This means that:

a) SO will be shifted out upon the falling edges of SK and will be stable during rising edges of SK.

b) SI will be shifted in upon the rising edge of SK, and will be stable when executing, i.e., an XAS instruction.

The shifting function is automatically performed on each of the four instruction cycles that follow an XAS instruction *(Figure 5)*.

When the SIO register is in the shift register mode (EN0 = 0), it left shifts its contents once each instruction cycle. The data present on the SI input is shifted into the least significant bit (bit 0) of the serial shift register. SO will output the most significant bit of the SIO register (bit 3) if EN3 = 1. Otherwise, SO is held low. The SK is a logic controlled clock which issues a pulse each instruction cycle. To ensure that the serial data stream is continuous, an XAS instruction must be executed every fourth time. Serial I/O timing is related to instruction cycle timing in the following way:

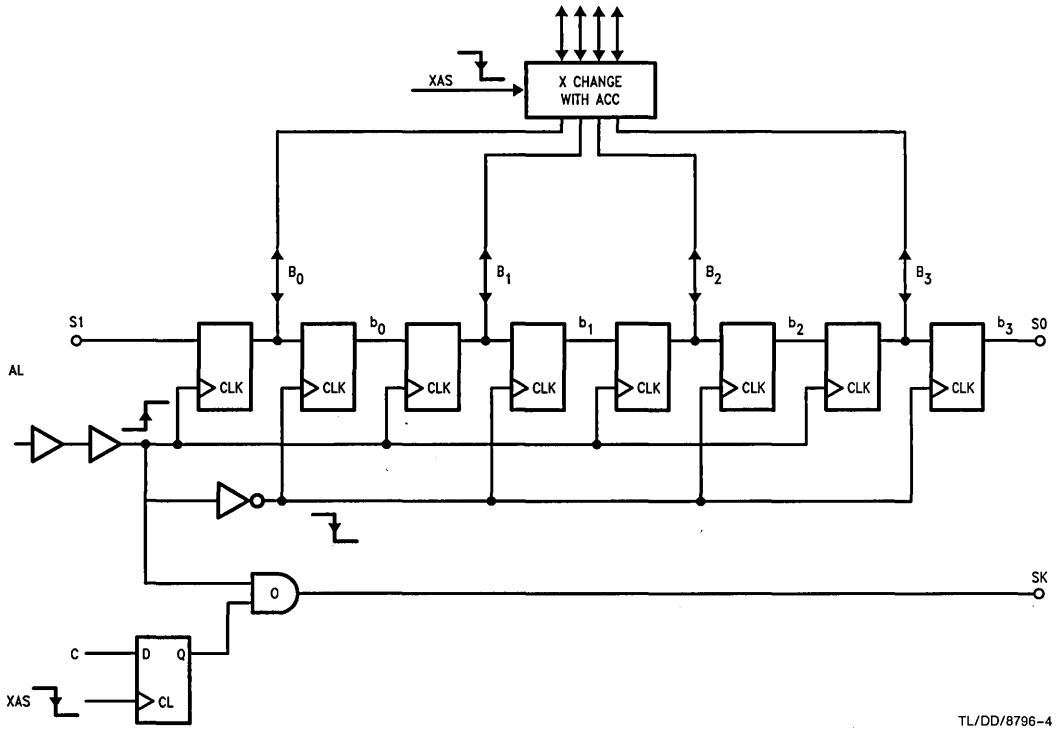


FIGURE 4

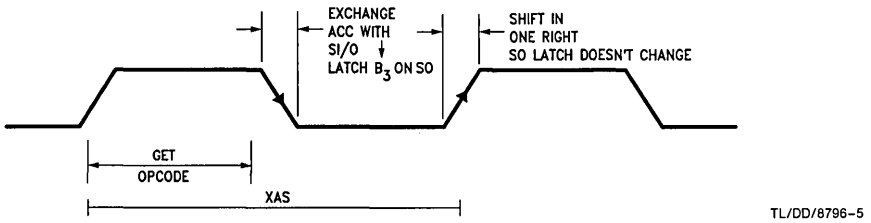
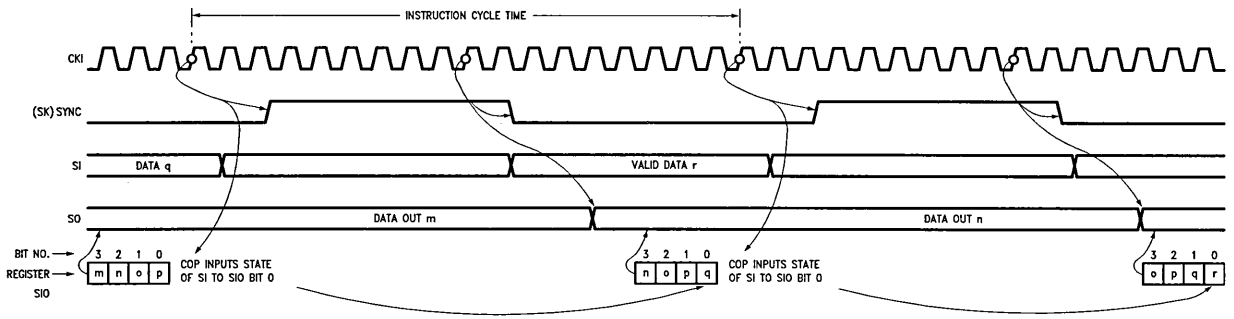
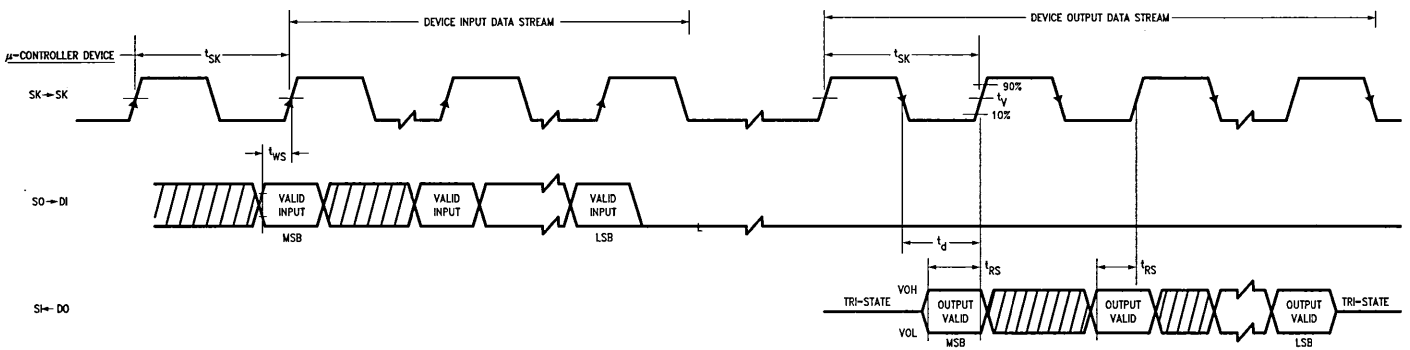


FIGURE 5. XAS Sequence



TL/DD/8796–6

FIGURE 6. Serial I/O Timing



TL/DD/8796–7

FIGURE 7. MICROWIRE Serial Data Exchange Timing

To write to device: $t_{ns} > t_{setup}$

To read from device: $t_d \ll t_{SK} - t_r$; $t_{RS} > t_{SK}/4$

Where: t_{ws} is MICROWIRE write data-in (DI) setup time,

t_{setup} is device data sheet min data setup time to latch in valid data,

t_{SK} is system clock (SK) cycle time (Recommended 50% duty cycle),

t_r is rise time (10% to 70% bout) of system clock (SK),

t_d is device actual delay time before data-out (DO) valid and

t_{RS} is minimum data setup time for controller to shift-in valid data

The first clock rising edge of the instruction cycle triggers the low-to-high transition of SYNC output via SK. At this time, the processor reads the state of SI into SIO bit 0, shifting the current bits 0–2 left. Halfway through the cycle (shown in *Figure 6* as the eight clock rising edge), SK is reset low and the new SIO bit 3 is outputted via SO.

INTERFACING CONSIDERATIONS

To ensure data exchange, two aspects of interfacing have to be considered: 1) serial data exchange timing; 2) fan-out/fan-in requirements. Theoretically, infinite devices can access the same interface and be uniquely enabled sequentially in time. In practice, however, the actual number of devices that can access the same serial interface depends on the following: system data transfer rate, system supply requirement capacitive loading on SK and SO outputs, the fan-in requirements of the logic families or discrete devices to be interfaced.

HARDWARE INTERFACE

Provided an output can switch between a HIGH level and a LOW level, it must do so in a predetermined amount of time for the data transfer to occur. Since the transfer is strictly synchronous, the timing is related to the system clock (SK) *(Figure 7)*. For example, if a COPS controller outputs a value at the falling edge of the clock and is latched in by the peripheral device at the rising edge, then the following relationship has to be satisfied:

$$t_{DELAY} + t_{SETUP} \leq t_{CK}$$

where t_{CK} is the time from data output starts to switch to data being latched into the peripheral chip, t_{SETUP} is the setup time for the peripheral device where the data has to be at a valid level, and t_{DELAY} is the time for the output to read the valid level. t_{CK} is related to the system clock provided by the SK pin of the COPS controller and can be increased by increasing the COPS instruction cycle time.

The maximum t_{SETUP} is specified in the peripheral chip data sheets. The maximum t_{DELAY} allowed may then be derived from the above relationship.

Most of the delay time before the output becomes valid comes from charging the capacitive load connected to the output. Each integrated circuit pin has a maximum load of 7 pF. Other sources come from connecting wires and connection from PC boards. The total capacitive load may then be estimated. The propagation delay values given in data sheets assume particular capacitive loads (e.g. V_{CC} = 5V, V_{OH} = 0.4V , loading = 50 pF, etc.).

If the calculated load is less than the given load, those values should be used. Otherwise, a conservative estimate is to assume that the delay time is proportional to the capacitive load.

If the capacitive load is too large to satisfy the delay time criterion, then three choices are available. An external buffer may be used to drive the large load. The COPS instruction cycle may be slowed down. An external pullup resistor may be added to speed up the LOW level to HIGH level transition. The resistor will also increase the output LOW level and increase the HIGH level to LOW level transition time, but the increased time is negligible as long as the output LOW level changes by less than 0.3V. For a 100 pF load, the standard COPS controller may use a 4.7k external resistor, with the output LOW level increased by less than 0.2V. For the same load, the low power COPS controller may use a 22k resistor, with the SO and SK LOW levels increased by less than 0.1V.

Besides the timing requirements, system supply and fan-out/fan-in requirements also have to be considered when interfacing with MICROWIRE. For the following discussion, we assume single supply push-pull outputs for system clock (SK) and serial output (SO), high-impedance input for serial input (SI).

To drive multi-devices on the same MICROWIRE, the output drivers of the controller need to source and sink the total maximum leakage current of all the inputs connected to it and keep the signal level within the valid logic "1" or logic "0". However, in general, different logic families have different valid "1" and "0" input voltage levels. Thus, if devices of different types are connected to the same serial interface, the output driver of the controller must satisfy all the input requirements of each device. Similarly, devices with TRI-STATE® outputs, when connected to the SI input, must satisfy the minimum valid input level of the controller and the maximum TRI-STATE leakage current of all outputs.

So, for devices that have incompatible input levels or source/sink requirements, external pull-up resistors or buffers are necessary to provide level-shifting or driving.

SOFTWARE INTERFACE

The existing MICROWIRE protocol is very flexible, basically divided into two groups:

1) 1AAA.....ADDD.....D

   where leading 1 is the start bit and leading zeroes are ignored.

   AAA.....A is device variable instruction/address word.

   DDD.....D is variable data stream between controller and device.

2) No start bit, just bit stream, i.e., bbb.....b

where b is a variable bit stream. Thus, device has to decode various fields within the bit stream by counting exact bit position.

SERIAL I/O ROUTINES

Routines for handling serial I/O are provided below. The routines are written for 16-bit transmissions, but are trivially expandable up to 64-bit transmissions by merely changing the initial LBI instruction. The routines arbitrarily select register 0 as the I/O register. It is assumed that the external device requires a logic low chip select. It is further assumed that chip select is high and SK and SO are low on entry to the routines. The routines exit with chip select high, SK and SO low. GO is arbitrarily chosen as the chip select for the external device.

SERIAL DATA OUTPUT

This routine outputs the data under the conditions specified above. The transmitted data is preserved in the microcontroller.

```
OUT2:  LBI   0,12   ; point to start of
                      data word
       SC
       OGI   14     ; select the external
                      device
```

TABLE I. MICROWIRE Standard Family

Features		Part Number							
		DS3906	MM545X	COP470	COP472	COP430 (ADC83X)	COP498/499	COP452L	COP494 (NMC9306)
GENERAL									
Chip Function		AM/PM PLL	LED Display Driver	VF Display Driver	LCD Display Driver	A/D	RAM & Timer	Frequency Generator	E²PROM
Process		ECL	NMOS	PMOS	CMOS	CMOS	CMOS	NMOS	NMOS
V_{CC} Range		4.75V–5.25V	4.5V–11V	−9.5V to −4.5V	3.0V–5.5V	4.5V–0.3V	2.4V–5.5V	4.5V–6.3V	4.5V–5.5V
Pinout		20	40	20	20	8/14/20	14/8	14	14
HARDWARE INTERFACE									
Min V_{IH}/Max V_{IL}		2.1V/0.7V	2.2V/0.8V	−1.5V/−4.0V	$0.7V_{CC}$/0.8V	2.0V/0.8V	$0.8V_{CC}$/$0.4V_{CC}$	2.0V/0.8V	2.0V/0.8V
SK Clock Range		0–625 kHz	0–500 kHz	0–250 kHz	4–250 kHz	10–200 kHz	4–250 kHz	25–250 kHz	0–250 kHz
Write Data DI	Setup Min	0.3 μs	0.3 μs	1.0 μs	1 μs	0.2 μs	0.4 μs	800 ns	0.4 μs
	Hold Min	0.8 μs	(3)	50 ns	100 ns (Note 1)	0.2 μs	0.4 μs	1.0 μs	0.4 μs
Read Data Prop Delay		(Note 4)	(Note 3)	(Note 3)	(Note 3)	(Note 3)	2 μs (Note 2)	1 μs (Note 2)	2.0 μs
Chip Enable	Setup	0.3 μs	0.4 μs	1.0 μs Min	1 μs (Note 1)	0.2 μs	0.2 μs (Note 1)	(Note 3)	0.2 μs
	HOLD	0.8 μs	(Note 3)	1.0 μs Min	1 μs (Note 2)	0.2 μs	0 (Note 2)	(Note 3)	0
Max Frequency Range	AM	8 MHz	(Note 3)	(Note 3)	(Note 3)	(Note 3)	(Note 3)	(Note 3)	(Note 3)
	FM	120 MHz	(Note 3)	(Note 3)	(Note 3)	(Note 3)	(Note 3)	(Note 3)	(Note 3)
Max Osc Freq.		(Note 3)	(Note 3)	250 kHz	(Note 3)	(Note 3)	2.1 MHz (−21) 32 kHz (−15)	256–2100 kHz (−4) 64–525 kHz (−2)	(Note 3)
SOFT									
Serial I/O Protocol		11D1...D20	1D1...D35	8 Bits At a Time	b1...b40	1xxx	1yyxxD6...D0 Start Bit	1yxxxx	1AA...DD
Instruction/ Address Word		None	None	None	None	(Note 4)	(Note 4)	(Note 4)	(Note 4)

Note 1: Reference to SK rising edge.

Note 2: Reference to SK falling edge.

Note 3: Not defined.

Note 4: See data sheet for different modes of operation.

```
        LEI  8      ; enable shift
                      register mode
        JP   SEND2
SEND1:  XAS
SEND2:  LD          ; data output loop
        XIS
        JP   SEND1
        XAS         ; send last data
        RC
        CLRA
        NOP
        XAS         ; turn SK clock off
        OGI  15     ; deselect the device
        LEI  0      ; turn SO low
        RET
```

The code for reading serial data is almost the same as the serial output code. This should be expected because of the nature of the SIO register and the XAS instruction.

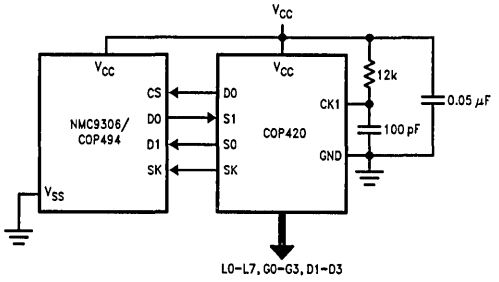
MICROWIRE STANDARD FAMILY

A whole family of off-the-shelf devices exists that is directly compatible with MICROWIRE serial data exchange standard. This allows direct interface with the COPS family of microcontrollers.

Table I provides a summary of the existing devices and their functions and specifications.

TYPICAL APPLICATION

Figure 8 shows pin connection involved in interfacing an NMC9306/COP494 E²PROM with the COP420 microcontroller.

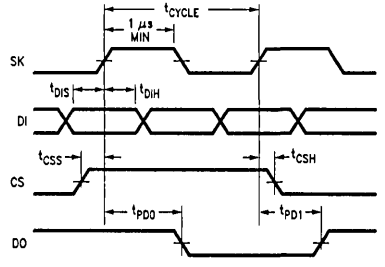


TL/DD/8796-8

FIGURE 8. NMC9306/COP494-COP420 Interface

The following points have to be considered:

1. For COP494 the SK clock frequency should be in the 0 kHz–250 kHz range. This is easily achieved with COP420 running at 4 µs–10 µs instruction cycle time (SK period is the COP420 instruction cycle time). Since the minimum SK clock high time is greater than 1 µs, the duty cycle is not a critical factor as long as the frequency does not exceed the 250 kHz max.
2. CS low period following an E/W instruction must not exceed 30 ms maximum. It should be set at typical or minimum spec of 10 ms. This is easily done in software using the SKT timer on COP420.



TL/DD/8796-9

FIGURE 9. NMC9306/COP494 Timing

3. As shown in WRITE timing diagram, the start bit on DI must be set by a "0" to "1" transition following a CS enable ("0" to "1") when executing any instruction. One CS enable transition can only execute one instruction.
4. In the read mode, following an instruction and data train, the DI can be a "don't care," while the data is being outputted, i.e., for the next 17 bits or clocks. The same is true for other instructions after the instruction and data has been fed in.
5. The data-out train starts with a dummy bit 0 and is terminated by chip deselect. Any extra SK cycle after 16 bits is not essential. If CS is held on after all 16 of the data bits have been outputted, the DO will output the state of DI until another CS LO to HI transition starts a new instruction cycle.
6. After a read cycle, the CS must be brought low for one SK clock cycle before another instruction cycle starts.

INSTRUCTION SET

Commands	Opcode	Comments
READ	10000A3A2A1A0	Read Register 0–15
WRITE	11000A3A2A1A0	Write Register 0–15
ERASE	10100A3A2A1A0	Erase Register 0–15
EWEN	111000 0 0 1	Write/Erase Enable
ENDS	111000 0 1 0	Write/Erase Disable
***WRAL	111000 1 0 0	Write All Registers
ERAL	111000 1 0 1	Erase All Registers

All commands, data in, and data out are shifted in/out on rising edge of SK clock.

Write/erase is then done by pulsing CS low for 10 ms.

All instructions are initiated by a LO-HI transition on CS followed by a LO-HI transition on DI.

READ— After read command is shifted in DI becomes don't care and data can be read out on data out, starting with dummy bit zero.

WRITE— Write command shifted in followed by data in (16 bits) then CS pulsed low for 10 ms minimum.

ERASE

ERASE ALL—Command shifted in followed by CS low.

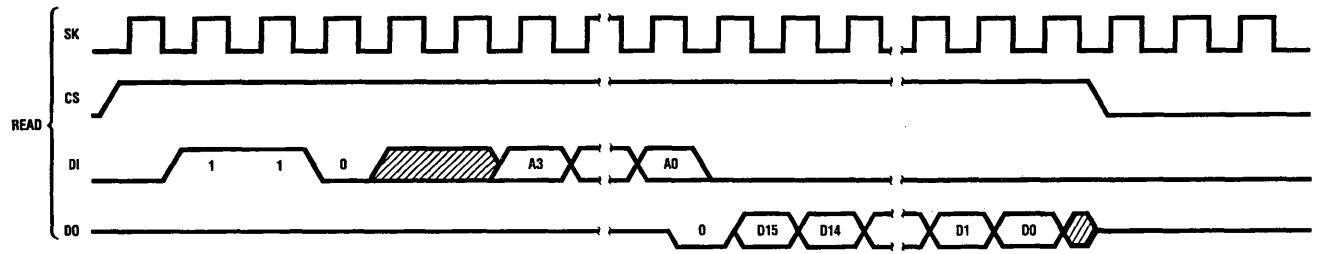
WRITE ALL—Pulsing CS low for 10 ms.

WRITE

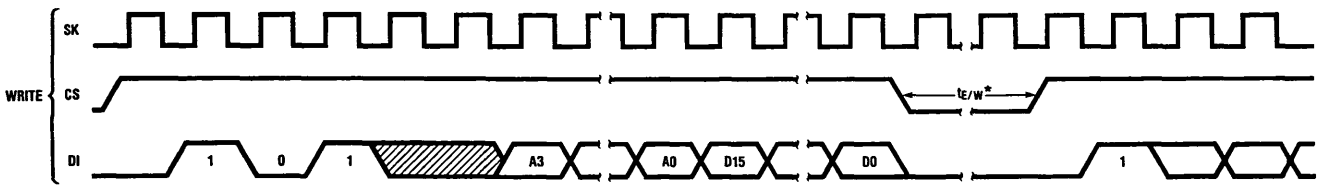
ENABLE/DISABLE—Command shifted in.

***(This instruction is not speced on Data sheet.)

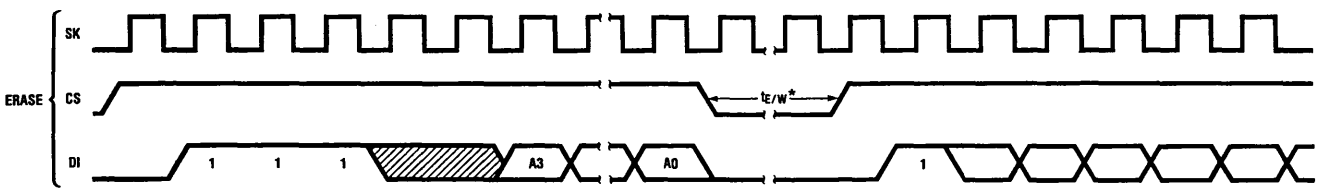
Instruction Timing



TL/DD/8796–10

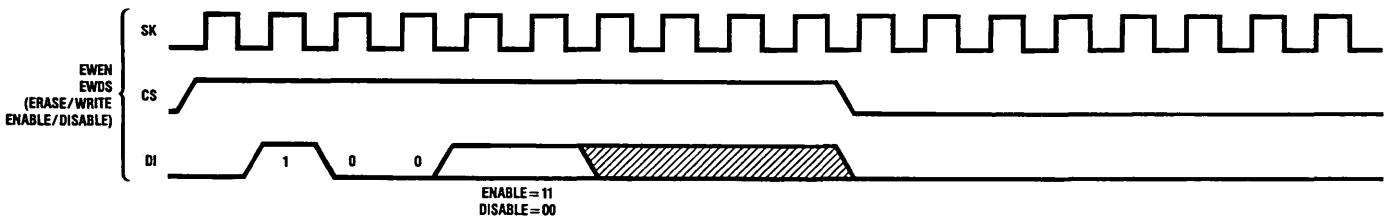


TL/DD/8796–11

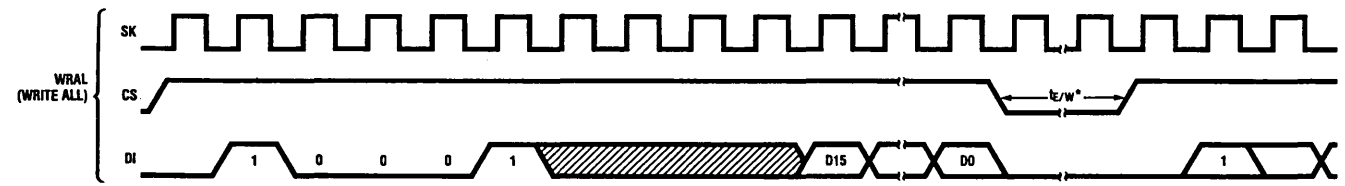


TL/DD/8796–12

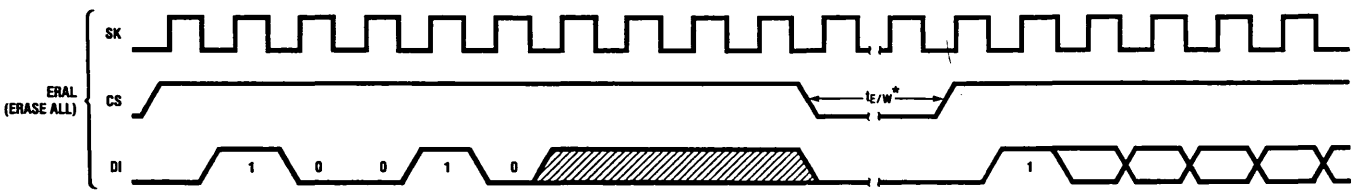
*$t_{E/W}$ measured to rising edge of SK or CS, whichever occurs last.



TL/DD/8796–13



TL/DD/8796–14



TL/DD/8796–15

*$t_{E/W}$ measured to rising edge of SK or CS, whichever occurs last.

I/O ROUTINE TO EVALUATE COP494

```
  1                         .TITLE      E494,  "I/O ROUTINE TO EVALUATE COP494"
  2     01A4                .CHIP       420
  3     0000                .PAGE       0
  4
  5                 ;THIS IS I/O ROUTINE TO EVALUATE COP494
  6                 ;
  7                 ;
  8
  9                 ;RAM VARIABLES DECLARATIONS:
 10     000E               COMMAND = 0, 14          ;494 8BITS INST/ADDR WORD
 11     001C               RWDATA  = 1, 12          ;494 16BITS R/W DATA BUFFER
 12
 13 000 00          PON:    CLRA                    ;POWER-ON INIT
 14 001 32                  RC                      ;RESET SK CLOCK
 15 002 4F                  XAS
 16 003 3F          CLRAM:  LBI     3, 0            ;CLEAR RAM FROM 7, 0 TO 0, 15
 17 004 00          CLR:    CLRA                    ;
 18 005 04                  XIS                     ;
 19 006 C4                  JP      CLR             ;CONTI CLEAR REG
 20 007 12                  XABR                    ;(A) TO BR
 21 008 5F                  AISC    15              ;REG 0 CLEARED?
 22 009 600F        DONE:   JMP     C494DR          ;Y, DONE CLEAR RAM, CALL 494 D
 23 00B 12                  XABR                    ;N, DEC BR
 24 00C C4                  JP      CLR             ;CONTI CLEAR REG TILL DONE
 25 00D 44                  NOP
 26 00E 44                  NOP
 27
 28                 ;***    START 494 DRIVER SAMPLE CALLING SEQUENCE   ***
 29
 30                 C494DR:                         ;INIT CALLING SEQUENCE
 31 00F 3350                OGI     0               ;GO=L TO DESELECT 494
 32 011 3368                LEI     8               ;ENABLE SIO AS S.R.
 33                 ERASE:
 34 013 0D                  LBI     COMMAND         ;PRELOAD 494 ERASE REG A3-A0
 35 014 7C                  STII    0C              ;PRELOAD 494 ERASE INST
 36 015 70                  STII    0               ;SELECT REG A3-A0
 37 016 690E                JSR     WI4P4           ;SEND IT
 38                 WEEN:
 39 01B 0D                  LBI     COMMAND         ;LOAD 494 WHEN REG A3-A0
 40 019 73                  STII    3               ;PRELOAD 494 WREN INST
 41 01A 70                  STII    0               ;SELECT REG A3-A0
 42 01B 690E                JSR     WI494           ;SEND IT
 43                 WRITE:
 44 01D 0D                  LBI     COMMAND         ;PRELOAD WR REG A3-A0
 45 01E 74                  STII    4               ;PRELOAD 494 WRITE INST
 46 01F 70                  STII    0               ;SELECT REG A3-A0
 47 020 1B                  LBI     RWDATA          ;PRELOAD 494 SAMPLE WRITE DATA
 48 021 75                  STII    5
 49 022 7A                  STII    ON
 50 023 75                  STII    5
```

I/O ROUTINE TO EVALUATE COP494 (Continued)

```
 51 024 7A             STII    0A
 52 025 6900           JSR     WD494           ;SEND THEM TO 494
 53           READ:
 54 027 0D             LBI     COMMAND         ;PRELOAD READ REG A3-A0
 55 028 78             STII    8               ;PRELOAD 494 READ INST
 56 029 70             STII    0               ;SELECT REG A3-A0
 57 02A 6908           JSR     RD494           ;READ 494 DATA BACK VIA SI
 58 02C 44             NOP
 59 02D 44             NOP
 60
 61     0080           .PAGE   2               ;SUBROUTINE PAGE
 62 080 32    SETUP:   RC                      ;RESET SK BEFORE SELECT 494
 63 081 4F             XAS
 64 082 3351           0GI     1               ;GO=1 TO SELECT 494
 65 084 00             CLRA                    ;ENSURE SO=L BEFORE GEN START B
 66 085 22             SC                      ;
 67 086 4F             XAS                     ;TURN ON SK CLOCK
 68 087 00             CLRA                    ;GENERATE 494 START BIT
 69 088 51             AISC    1               ;
 70 089 22             SC                      ;
 71 08A 4F             XAS                     ;SEND IT AS MSB VIA SO
 72 08B 0D             LBI     COMMAND         ;FETCH 1ST INST/ADDR WORD
 73 08C 05             LD
 74 08D 44             NOP                     ;
 75 08E 4F             XAS                     ;SEND IT (MSB OF INST FIRST)
 76 08F 0E             LBI     COMMAND+1       ;FETCH 2ND INST/ADDR NIBBLE
 77 090 05             LD
 78 091 44             NOP
 79 092 4F             XAS                     ;SEND IT
 80 093 1B             LBI     RWDATA          ;POINT TO READ/WRITE DATA BUFFER
 81 094 48             RET                     ;RET OF SETUP
 82
 83 095 00    TWEDLY:  CLRA                    ;VPP WIDTH, TWE>20MS @ 4Us/INST
 84 096 5B    TWECONT: AISC    11              ;5 SKT LOOPS?
 85 097 99             JP      . + 2           ;N,CONTI
 86 098 48    TWEDONE: RET                     ;Y,DONE
 87 099 41             SKT                     ;
 88 09A 99             JP      . -1            ;
 89 09B 96             JP      TWECONT         ;CONTI TWE TIME
 90
 91 09C 48    RET;     RET                     ;2 CYCLES DELAY
 92
 93     0100           .PAGE   4               ;
 94
 95           ;***     START 494 I/O DRIVER SUBROUTINE    ***
 96
 97 100 80    WD494:   JSRP    SETUP           ;ENTRY TO WRITE 494 REG A3-A0
 98 101 05    RWLOOP:  LD                      ;R/W 494 16 DATA BITS
 99 102 4F             XAS                     ;
100 103 04             XIS                     ;
```

SOFTWARE DEBUG OF SERIAL REGISTER FUNCTIONS

In order to understand the method of software debug when dealing with the SIO register, one must first become familiar with the method in which the COPS MOLE™ (Development System) BREAKPOINT and TRACE operations are carried out. Once these operations are explained, the difficulties which could arise when interrogating the status of the SIO register should become apparent.

SERIAL OUT DURING BREAKPOINT

When the MOLE BREAKPOINTs, the COPS user program execution is stopped and execution of a monitor-type program, within the COP device, is started. At no time does the COP part "idle." The monitor program loads the development system with the information contained in the COP registers.

Note also that single-step is simply a BREAKPOINT on every instruction.

If the COP chip is BREAKPOINTed while a serial function is in progress, the contents of the SIO register will be destroyed.

By the time the monitor program dumps the SIO register to the MOLE, the contents of the SIO register will have been written over by clocking in SI. To inspect the SIO register using BREAKPOINT, an XAS must be executed prior to BREAKPOINT; therefore, the SIO register will be saved in the accumulator.

An even more severe consequence is that the monitor program executes an XAS instruction to get the contents of the SIO register to the MOLE. Therefore, the SK latch is dependent on the state of the CARRY prior to the BREAKPOINT. In order to guarantee the integrity of the SIO register, one must carefully choose the position of the BREAKPOINT address.

As can be seen, it is impossible to single-step or BREAKPOINT through a serial operation in the SIO register.

SERIAL OUT DURING TRACE

In the TRACE mode, the user's program execution is never stopped. This mode is a real-time description of the program counter and the external event lines; therefore, the four external event lines can be used as logic analyzers to monitor the state of any input or output on the COPS device. The external event lines must be tied to the I/O which is to be monitored.

The state of these I/O (external event lines) is displayed along with the TRACE information. The safest way to monitor the real-time state of SO is to use the TRACE function in conjunction with the external event lines.

CONCLUSIONS

National's super-sensible MICROWIRE serial data exchange standard allows interfacing to any number of specialized peripherals using an absolute minimum number of valuable I/O pins; this leaves more I/O lines available for system interfacing and may permit the COPS controller to be packaged in a smaller package.

COPS™ Based Automobile Instrument Cluster

National Semiconductor Corp.
Application Note 453
Venkata T. Gobburu

ABSTRACT

Dedicated microprocessor systems find increasing applications in automobile instrumentation. Fuel injection systems, digital radio tuners and similar applications employing the microcontroller have become common place. This paper describes a cost effective microcontroller implementation of an automobile instrument cluster by the COPS group of National Semiconductor, Santa Clara. The instrument cluster provides a vacuum fluorescent display of the vehicle speed, engine RPM, odometers, battery voltage, engine oil pressure and the fuel level. A modular design involving a single microcontroller in conjunction with peripherals to aid in data acquisition from the transducers allows the quantities to be computed with high accuracies and displayed on a real time basis. The single microcontroller environment places severe restrictions on the availability of RAM and ROM. Coupled with the requirement of real time operation the application poses a non trivial challenge. A nonvolatile RAM accumulates the mileage covered. Hamming code techniques ensure the integrity of the data contained in the nonvolatile memory. Inclusion of diagnostics allows a rapid and thorough check against improper operation of the microcontroller, peripherals and the nonvolatile memory. This paper describes the implementation with a COP444L containing 128 nybbles of RAM and 2K bytes of ROM. A display updation rate of 16 Hz can be comfortably realized.

Over the microcomputer usage has diversified dramatically in its scope and breadth. Dedicated microprocessor systems find increasing application in automobile instrumentation and control. From its inception the automobile has acquired considerable sophistication. Increasing demands have been made of the car. Fuel efficiency, higher acceleration rates, simplicity of control and improved ride quality rank high in the demands made of the car. In response the automobile engine has evolved into a complex machine. Crude methods to control or monitor its performance no longer suffice. Microprocessor based fuel injection techniques and ignition control are becoming quite ubiquitous.

The automobile instrument cluster monitors the engine and regularly updates a status display for the operator's benefit. Pertinent information includes the vehicle speed, the engine crankshaft rotational speed, oil pressure in the engine cylinders, condition of the battery and the mileage accumulated. The instrument cluster provides a visual feedback link to the operator allowing corrective action to be initiated as the need arises.

THE AUTOMOBILE INSTRUMENT CLUSTER

The heart of the Automobile Instrument Cluster (AIC) lies in obtaining raw data from various transducers and manipulating it to a form suitable for feedback to the human operator. The feedback, normally visual, conveys the vehicle speed, the engine rpm, the engine temperature, oil pressure, the battery voltage and the odometer values. The AIC can be viewed as a collection of either inherently independent or weakly linked subtasks. Each subtask can be further partitioned into three blocks viz. of raw data collection, processing and displaying it. The component subtasks, in spite of their high degree of independence, can be grouped on the basis of signal available from the transducers. Grouping the subtasks modularizes the design. Partitioning the design in this manner highlights two groups, the first requires a frequency to be measured and the second a voltage level. The two major groupings are briefly examined.

Transducers for the vehicle speed monitor the driveshaft rotation. Computing the engine rpm involves measuring the crankshaft revolution rate. The two independent problems can be seen to basically consist of measuring revolution rates. Transducers based on Hall effect phenomena have been used with commendable success. Alternately the fact that mounting magnets around the driveshaft circumference generates a known number of pulses per shaft rotation can be used effectively. A normally open cam operated reed switch with closure to ground creates a simple revolution transducer. In all the cases the transducer generates a frequency proportional to the quantity under consideration. Obviously some signal conditioning is required before using the frequency with digital components. The describing function can be simply stated as

$$V = k \times f \quad (1)$$

where

V is the quantity under measurement, the vehicle speed or the engine rotational speed
k is a proportionality constant
f is the transducer freqeuncy output

The proportionality constant, k, can be suitably modified to include changes back and forth between British and metric units.

The problem of measuring the transducer output frequency can be restated to be one of measuring the time period. In case of digital frequencies the equation (1) can be rewritten as

$$V = k/(Ton + Toff) \quad (2)$$

where

Ton is the ON time and
Toff is the OFF time

while the remaining symbols retain their definition from the earlier equation.

The remaining quantities such as the engine temperature, oil pressure, battery voltage and available fuel prove to be slow changing ones. The lower dynamics allow them to be transduced as voltage level signals. Equation (3) states the underlying relation and closely resembles the equations stated above.

$$P = k \times v \quad (3)$$

where

v is the voltage output of the transducer
P is the quantity under measurement
k is the proportionality constant

Evaluating the accumulating mileage depends indirectly upon the vehicle speed subtask. Integrating the signal from the vehicle speed transducer over time allows the mileage to be accumulated. The associated problems of storing the odometer information and ensuring its integrity require error correcting techniques. They are covered in a later section of the paper.

SYSTEM DESCRIPTION

The COPS Group of National Semiconductor, Santa Clara, offers a wide array of microcontrollers and peripherals to suit this application. Judicious selection of peripherals to aid the microcontroller can reinforce the partitioning suggested earlier to considerably simplify the implementation. *Figure 1* presents a functional block diagram of the AIC.

A COP444L four bit microcontroller provides the necessary computing and decision making capability. Equipped with 128 nybbles of RAM space organized in a matrix fashion and 2K ROM space for storage of the control program, the COP444L operating at an instruction cycle rate of 16 microseconds sequentially obtains information from the peripherals and formats the manipulated results to be manageable by the display drivers. Transducers for the vehicle speed and the engine speed provide proportional frequency signals. Two COP452 peripherals, placed in a Waveform Measure Mode, track the ON time and OFF time of the conditioned transducer outputs. Voltage level signals available from the transducers for the engine temperature, oil pressure, battery condition and the fuel tank can be monitored by a COP438, an eight channel A/D converter. An electronically erasable non volatile RAM, the COP494, allows the odometer information to be stored safely under power down conditions.

A combination of LEDs, vacuum fluorescent displays and high intensity lamps comprise the optical elements of the AIC Standard eight segment alphanumeric and bargraph format displays have been used. A 32 segment LED bargraph, controlled by a MM5450 static display driver, displays the engine rpm. Eight segment alphanumeric vacuum fluorescent displays are used for the vehicle speed and the odometer values. Sixteen segment vacuum fluorescent bargraph displays are used for the engine temperature and available fuel quantity. The battery voltage and oil pressure utilize eight segment vacuum fluorescent bargraph displays. Any potentially dangerous situations detected by the COP444L are underlined by high intensity lamps. Five COP470 display drivers multiplex the various displays under the microcontroller's orchestration.

Single pole single throw switches allow the user to select between the British or the metric units, the trip or the accumulated odometer and reset the trip odometer.

SYSTEM DIAGNOSTICS

Diagnostics aid in isolating faulty components within a system. The algorithmic nature of the diagnostic procedure allows it to be implemented via a microprocessor. A great deal of attention has been focused on diagnostics as considerable cost savings can accrue from a microprocessor based scheme minimizing human involvement. Programming the AIC, in addition to its normal functions, with self test capabilities increases its potential for high volume applications. Normally diagnostics imply using independent means to evaluate the system's performance. Attempting to incorporate self test capabilities necessitates adopting an "inside out" strategy. A basic kernel is first evaluated as functioning correctly. Over iterations the kernel expands by establishing correct operation of other modules.

The AIC implementation described in this paper has an extensive repertoire of diagnostics to check the microcontroller and ensure correct operation of the peripherals. The probability of the microcontroller ROM failing proves to be negligibly small compared to a fault developing in the hardware interconnections. Also the idea of encoding in ROM the algorithm to check ROM data proves suspect. Control program stored in the ROM forms the kernel assumed to be functioning correctly. Writing and reading back an alternating pattern of ones and zeros in the microcontroller RAM checks for leakage of data into adjacent locations. Applying a known voltage, derived locally, to one of the four unused channels on the A/D converter allows it to be tested. The architecture of the COP452 peripherals consists of two independent register-counter pairs. The counters count down from the initial value. To test the COP452 both the register counter pairs have to be checked. By placing the two in a Duty Cycle Mode, the counters can be loaded with initial values from the registers and set to count down. The contents of the counters after a predetermined delay can detect incorrect operation of the device. A fault at the level of a register-counter pair can thus be isolated.

The COP494 stores the odometer information. It becomes vital to maintain the integrity of the information stored in the nonvolatile memory. Continuous use of particular locations in the COP494 can result in failures, typically bit dropouts. It is imperative to be capable of recovering from such errors. Requiring a single COP494 unit to last at least the expected lifetime of the vehicle influences the design of the storage scheme. The AIC implementation described in this paper depends upon Hamming encoding techniques to provide single bit error recovery. Subsequent to recovering from a single bit error all data transactions are carried out from a new location. A flashing display sequence alerts the operator of the occurrence of a non-recoverable error. Suspending all normal functions during such conditions can be used to force the vehicle to be taken to an authorized dealer. Breaking up the odometer data into sections allows updating of particular sections as opposed to restoring the whole every time. Such a strategy maximizes the lifetime of the nonvolatile memory.

SOFTWARE DESCRIPTION

The functional objectives of the AIC and the hardware required to realize them have been detailed in earlier sections of the paper. A summary of the software features completes the description and aids in developing a global understanding of the AIC. The AIC software, written in COP microcontroller assembly language, reflects the modular nature of the problem. The finite amount of memory of ROM space available on the COP444L coupled with real time operation requirements makes programming the AIC a non-trivial problem. Each subtask grouping has been organized as a distinct block of code. The microcontroller sequentially processes each subtask. A brief examination of the salient features follows.

It must be borne in mind that the COP452 peripheral captures an instantaneous picture of the frequency. The strength of the magnets, mounted circumferentially on the driveshaft to transduce revolution rate, cannot be precisely controlled. As a result the transducer, although generating a fixed number of pulses per revolution of the driveshaft, produces a pulse train showing both pulse period and duty cycle variations. Directly using the pulse period from the

COP452 leads to erroneous values of the vehicle speed. The computed vehicle speed, under steady vehicle speed conditions, shows excursions on either side of the nominal value. The first AIC implementation studied the application of an essentially single pole filter with different damping constants to exclude the oscillations. Although a sufficiently damped filter can effectively reduce the oscillations the scheme was discarded in lieu of the resulting degradation in response time. The solution lies in basing the vehicle speed computation on pulse period measurements averaged over consecutive pulses. Since the number of pulses per revolution is known, eight in this case, averaging the pulse period over this number minimizes the steady state error and responds fast. The nature of the solution affects the software organization. It falls upon the microcontroller to sample the conditioned output of the transducer and obtain pulse periods for eight consecutive pulses. To achieve this the software adopts a foreground-background organization. Monitoring the transducer output to catch the consecutive pulses forms the background job. The normal functions of the AIC form the foreground job. Additionally a minimal sampling rate has to be maintained to ensure that even at highest attainable vehicle speeds the microcontroller measures consecutive pulses.

The AIC electronically stores the odometer information in the non-volatile memory. Loss of odometer integrity can be disastrous. Consequently the ability to recover from errors in the non-volatile memory becomes very important. The AIC depends on single bit error correcting Hamming coding methods to avoid loss of information. The algorithm processes the odometer nybble fashion and simplifies the related problems of encoding the data prior to storing it and decoding the composite for data retrieval to trivial table lookups. LQID, a powerful member of the microcontroller instruction set, allows an eight bit value to be looked up based on the key value in the addresed RAM location. To minimize ROM space both the encoding and the decoding sections of the algorithm share the same error table and code for table lookups.

The remaining sections of the AIC software, also exhibit a block structure, do not prove to be as subtle. The straight forward code includes routines such as multiplications and divisions to help in the computations and routines allowing the microcontroller to communicate serially over the MICROWIRE™ with the peripherals.

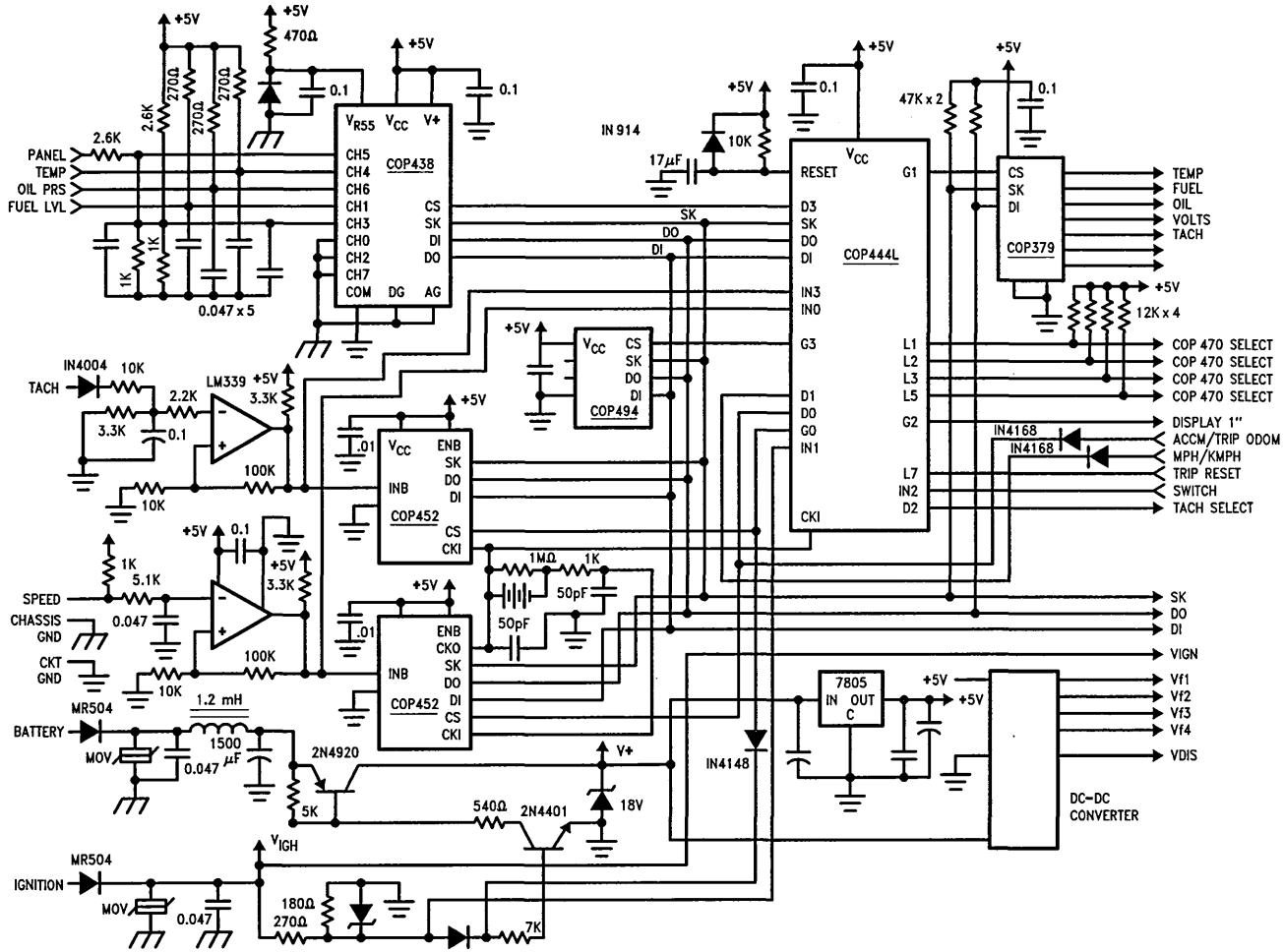
RESULTS AND CONCLUSIONS

The AIC implemented via the COP444L approximately uses 2K of ROM space. The COP444L, running at an instruction cycle time of 16 microseconds, sequences through all the functions in 228 milliseconds. The resulting display updation rate of approximately 4 Hz can be trivially increased to 16 Hz by replacing the COP444L with the equivalently packaged COP440. Table I presents in tabular form the accuracies and speeds at which the different measurements are done. It also shows the proportional speed increases obtainable.

The minimal number of peripherals used combined with the inclusion of diagnostics and error correction emphasize its low cost capabilities. The results serve to validate the feasibility of a cost effective microcontroller based Automobile Instrument Cluster.

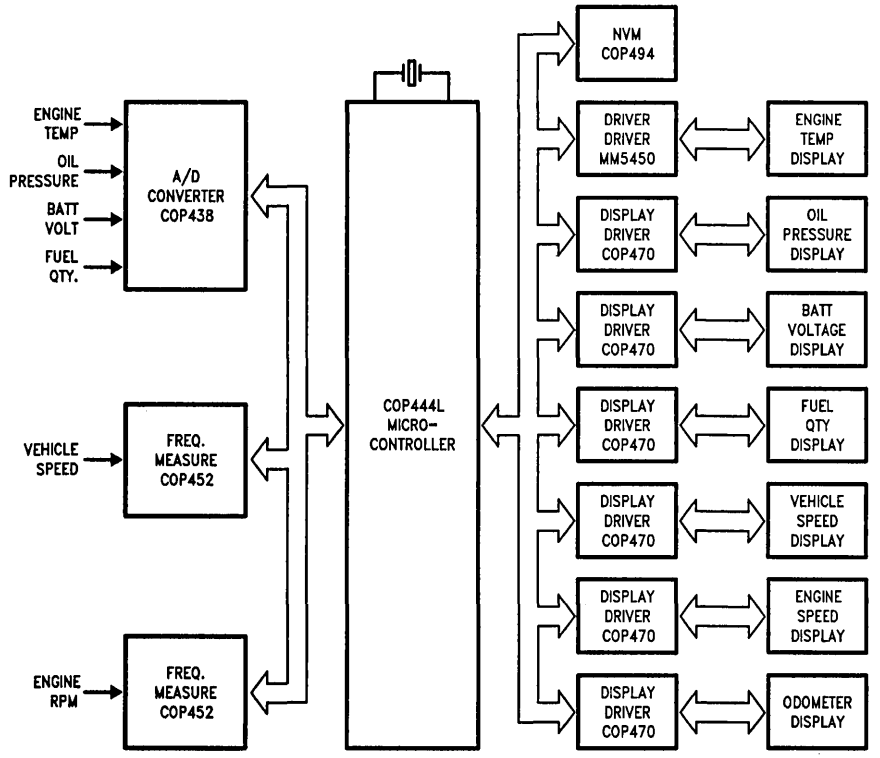
TABLE I. Comparison of Speed and Resolution of Measurements Taken with the COP444L and the COP440

	Measurements with a COP444L		Measurements with a COP440	
	Time Taken μsecs	Resolution Bits	Time Taken μsecs	Resolution Bits
1. Engine rpm	768	17	192	17
2. Vehicle Speed	768	17	192	17
3. Engine Temperature	256	8	64	8
4. Oil Pressure	256	8	64	8
5. Battery Voltage	256	8	64	8
6. Fuel Quantity	256	8	64	8



TL/DD/8798–1

FIGURE 1. Functional Block Diagrams of the AIC



TL/DD/8798–2

FIGURE 2

Automotive Multiplex Wiring

National Semiconductor Corp.
Application Note 454
Abdul H. Aleaf

INTRODUCTION

The evolutionary development of vehicle electronic systems has rapidly increased the number of individual wires in the vehicle. The conventional wiring harness will not provide solutions to the problems such as reducing size and weight in addition to meeting cost and reliability objectives. Several approaches have been taken to provide long term solutions. None has succeeded. Miniaturization of cables and wires is one example of a temporary solution.

Multiplexing on the other hand has been regarded as a technique which allows considerable savings to be made in the size and cost of the harness. It can also enhance reliability by reducing the number of electrical connections.

In a multiplex system the control functions will be distributed around the vehicle and complex interconnections between diagnostic terminals, sensors, instruments and switches will not add to the harness complexity. With all its advantages it has not been implemented on a production car yet. The reason has been economical feasibility and lack of suitable semiconductor components for power switching. But, with the rapid technology advances in power FETs and introduction of low cost microcomputers, multiplex wiring can be regarded as a logical successor to conventional wiring systems. Extended development efforts are necessary to introduce a reliable system at reasonable cost.

The Microcontroller Applications Group at National Semiconductor has taken a step towards this goal. A low end multiplex wiring system focusing on asynchronous serial communication in a multi node network has been developed. This paper describes the development of this system on an abstract model which forms the basis for analysis of communication protocol and various node functions.

SYSTEM CONFIGURATION

Figure 1 presents a general view of the system. The system is a centralized single master multiple slave-node scheme. All units are connected together by a balanced twisted pair. The expandable interconnection of different subsystems is achieved with 9600 Baud communication over a standard UART bus. The bus handles the interface between a master controller and the intelligent nodes.

The approach to have a centralized control system offers several advantages as compared against a non-centralized system. It prevents the problem of bus monopolization by a faulty node and is potentially cheaper due to the need for only one complex node (master). The master-slave architecture also prevents bus contention problems.

The master is COP1430. The COP1430 is a 4-bit microcontroller with on chip UART that handles asynchronous communication with other processors at speeds up to 9600 Baud.

The use of 4-bit 49¢ microcontrollers (COP413L) at the nodes not only provides intelligence which reduces the required bus bandwidth, it also reduces the incremental cost associated with automotive multiplexing. All standard nodes are identical. One standard program is used. This uniformity contributes to the system flexibility and expandability. External standard nodes may be added to the system to control additional functions. Node types and addresses are selected via external wire jumpers or switches. The slave nodes consist of four remote units to handle functions such as headlamps, tail lamps, etc. These nodes are the front right, front left, rear right and rear left nodes. Incorporated into the system are also a keyboard node, a EIC node and a display node.

The keyboard node may call for a control action at any time. This node is being continuously monitored by the master controller which receives status and processes the command or information.

Overall system intelligence and flexibility is increased by dedicating a node to NS455 the Terminal Management Processor. This node takes the responsibility to display information on a 4" flat CRT display.

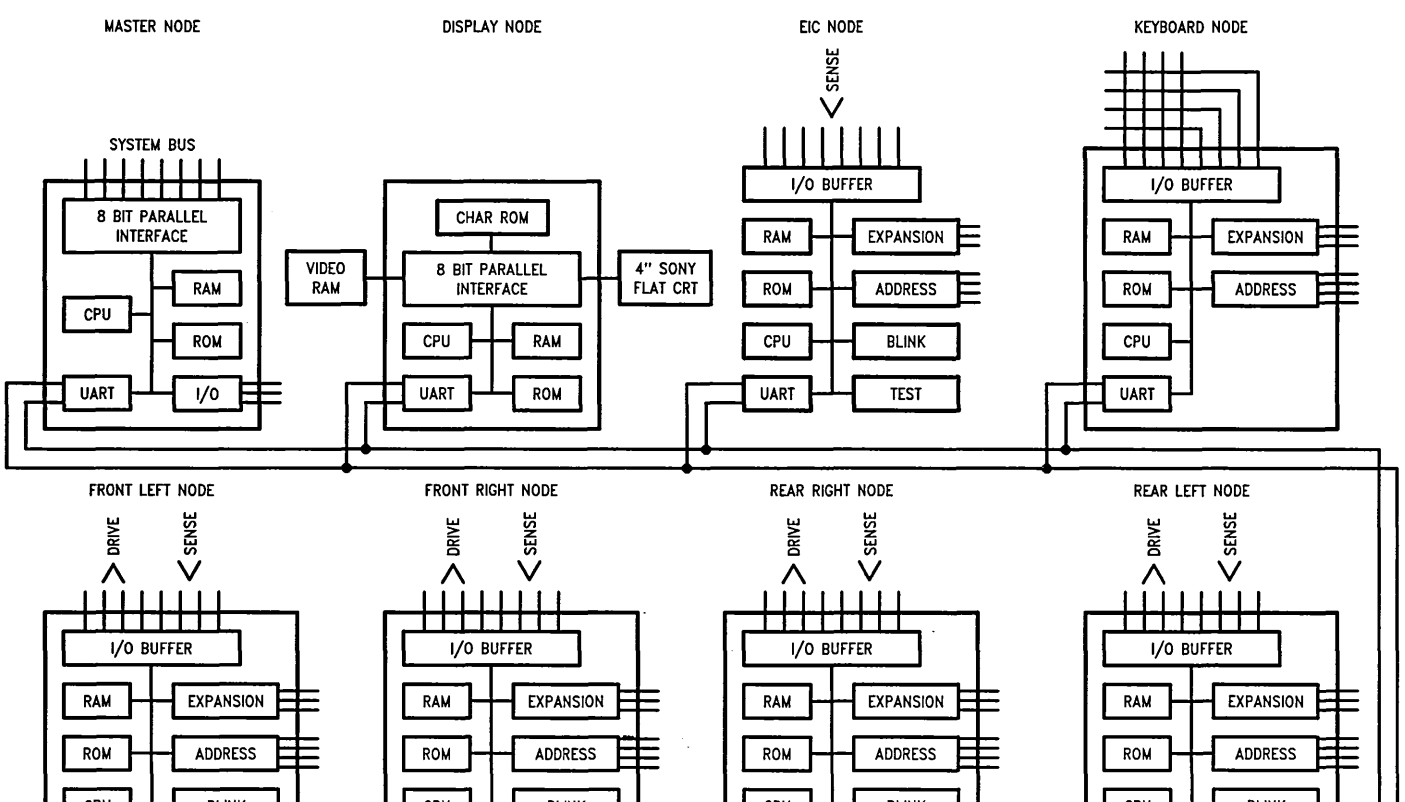
An Electronic Instrument Cluster (EIC) system is a completely independent system. It typically performs all functions associated with the automobile dashboard such as vehicle speed, odometers to accumulate mileage, gauges to display engine temperature, fuel level and so on. It also indicates error conditions such as high engine temperatures, low fuel level etc. The multiplex wiring system uses a standard slave node as a bridge between the two independent systems. The slave node monitors error conditions from the EIC system and passes them to the master node upon request. It becomes relatively simple to allow the master to access all activity in the EIC system via additional commands to the slave node serving the EIC system:

THE COMMUNICATION PROTOCOL

The master unit addresses the remote units sequentially and receives a status reply from each individual node. Data communication is via the standard UART format. It has a start bit, eight data bits, an even parity bit and one stop bit.

Information to be transmitted from the master to a slave node is organized as a frame. Each frame contains the address of destination and command or data. The information in a frame is transmitted as byte format. Address/data differentiation is done by means of a flag. The byte is an address byte if the MSB is set ("1"), otherwise it is a data byte.

Two different types of addressing schemes have been incorporated into the communication protocol; node addressing and class addressing. A class of nodes is formed by grouping together slave nodes with common functions. Commands may be executed either by specific individual nodes or by slave classes. All nodes of the same class execute the command simultaneously. The system implementation at National involved four classes with seven slave nodes per class. So, the total number of nodes possible in this system is 28.



TL/DD/8799–1

FIGURE 1. Block Diagram

The partitioning between the class address and node address reduces the density of bus traffic significantly by eliminating repetative command transmission to individual node class. Lower bus traffic implies that lower transmission bit rate can be used, allowing additional noise immunity. Another advantage of the class addressing is the provision of synchronization for control signals such as HAZARD, LEFT/RIGHT turns.

Error correction is incorporated into the communication protocol. The UART error flags such as PARITY and FRAMING ERRORS protect the system at the physical layer. At the system level, the nodes simply avoid sending an acknowledgement to the master when an error is detected. The master times out and sends the command again.

THE MASTER NODE

The master controller is the heart of the system. Its responsibility is to generate the controlling commands and synchronize the system. It transmits to the remote units and listens to them to get the vehicle status and acts accordingly. Circuit complexity is reduced by implementing extensive software programming in the master controller. This means that the burden is essentially on the master and must be engineered to very high standards of reliability. The device used in the implementation as the master is the COP1430. It is a cost effective 4-bit single chip microcontroller. It features on chip UART which handles asynchronous communications at speeds up to 9600 Baud.

THE SLAVE NODES

The standard slave nodes are based upon the COP413L. The COP413L is a low cost 4-bit microcontroller which may be customized in production. A system such as multiplex wiring requires power consumption to be absolutely minimal. Another basic requirement is that the system should be cost effective. These two facts directed us to use the COP413L at the standard slave node. The COP413L is a low cost (49¢!) low power microcontroller from NSC drawing less than 7 mA at 4.5V to 5.5V. The device contains an 8-bit bidirectional I/O port and a serial expansion port. The CMOS version of COP413L will also be available.

THE DISPLAY NODE

This node can serve as a condition monitoring unit for the vehicle. A considerable quantity of diagnostic information collected from transducers, switches, sensors and various loads are fed to this unit to be displayed on a CRT display. The node is based on a Terminal Management Processor the NS455. The NS455 is a CRT controller on chip. The messages are updated over the serial I/O line by the master controller. The communication format is:

a) The node receives the address.

b) If address matches the local node address, send the copy command

c) Receive new address and execute.

OUTPUT STAGES

The power FETs used for local switching throughout the system are IRF541[4]. These N-channel FETs provide much better drive circuit specification as compared to bipolar output stages. They also feature all of the well established advantages of MOSFET such as voltage control, very fast switching, and very low on state resistance. Another advantage is the lower cost as compared to comparibly rated p-channel devices.

TRANSMISSION MEDIUM

A balanced twisted pair is used for bus medium which provides high noise immunity. The transceiver selected for the bus is DS3695 *(Figure 2)*. This device is a high speed differential TRI-STATE® Bus/line transceiver designed to meet EIA standard for multipoint bus transmission. Bus contention or fault situations that cause excessive power dissipation within the device are handled by a standard thermal shutdown circuit, which forces the driver outputs into the high impedance state.

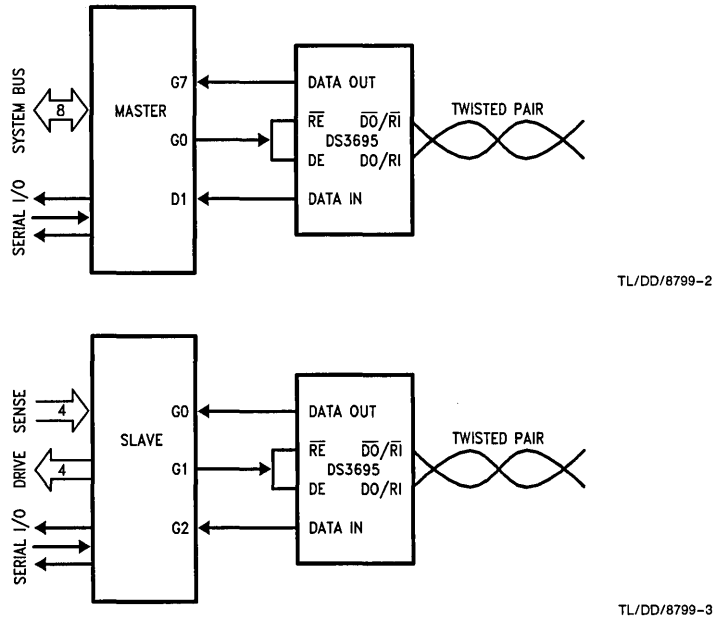


FIGURE 2. Bus Interface

CONCLUSIONS

Multiplex wiring system potentially seems to be a good replacement for conventional wiring system. Reduced complexity, increased flexibility and diagnostic capability could be achieved by incorporating microcontroller devices at nodes within the wiring system. The 4-bit microcontrollers selected are available in a price range, as low as 49¢, that will allow multiplex wiring to compare favorably on a cost-performance basis with the conventional harness.

REFERENCES

1. Michael W. Lowndes and Paul E. V. Phillips, "The Motorcar Multiplex Systems", IEE Conference on Automotive Electronics, 229, England, Nov. 1983.
2. R. F. Robins/W. J. Brittain/M. R. Lunt, "A Car Multiplex Wiring System with Self Coding Control Modules", IEE Conference on Automotive Electronics, 229, Ford Motor Company, UK, Nov. 1983.
3. Booth, J. A., 1983 "Vehicle Interconnection Systems for the Future", IEE Conference on Automotive Electronics, London, Nov. 1983.
4. International Rectifier, HEXFET Databook, 1985.

The COPS™ Programming Manual

National Semiconductor Corp.
Application Note 455

Table of Contents

1.0 Introduction to COPS™ Microcontrollers

1.1 SCOPE AND PURPOSE OF THIS MANUAL

How is an efficient COPS program written? The answer to this question begins with dividing the broad category of microcomputers into two areas: microcontrollers and microprocessors. This distinction is made because these are really two different types or classes of devices. Microcontrollers generally have a dual-bus architecture rather than the memory-mapped von Neumann architecture common in most microprocessors. For control applications, microcontrollers are generally more memory efficient than microprocessors. The microcontroller instruction set is quite different in nature than the microprocessor instruction set. Microcontrollers are invariably single-chip devices and microprocessors are, generally, multi-chip devices. Microcontrollers dominate the microcomputer marketplace in terms of volume. To be sure, the division between microcontroller and microprocessor is sometimes blurred but the distinction is real nonetheless.

COPS devices are microcontrollers. It is the intent of this manual to provide the user/programmer of COPS microcontrollers the requisite information to write an efficient COPS program—to take full advantage of the characteristics of the devices. To achieve that end, this manual is written from the programmer's perspective. The various characteristics of COPS microcontrollers are described in the context of the effect of those characteristics on the programming of the devices. The COPS architecture is discussed; the instruction set is described in detail; general techniques of COPS programming are explained; and standard programs are provided. The standard programs are commonly used as vehicles to illustrate various programming techniques. The user or reader would be well advised to carefully read the explanations associated with routines showing multiple implementations. The intent of providing multiple implementations is not to show how many different ways a routine can be written but rather to show techniques, "tricks", tradeoffs, considerations, etc. Therefore, a great deal of useful information is included in those explanations.

This manual does not attempt to explain the detailed physical or electrical characteristics of COPS microcontrollers. To the extent any such information is provided here, it is to explain some software effect or characteristic. Therefore, the physical details may be simplified to clarify the software explanation.

1.2 THE COPS MICROCONTROLLER FAMILY

1.2.1 General Decscription

COPS devices are general purpose, single-chip microcontroller. These microcontrollers are complete microcomputers containing all system timing, internal logic, ROM, RAM, and I/O necessary to implement dedicated control functions in a wide variety of applications. The COP400 family presently consists of a large number of devices enabling the user to select the device best suited to his application. The software is upward compatible—programs written on one device may be transferred to the next larger device (in terms of memory capacity) with little or no change. The package pin configurations have also been selected so that movement up or down (using memory size as the variable parameter) within the family can be accomplished easily. All COPS microcontrollers, regardless of memory size or number of pins, have the same basic architectural structure. In addition to the large number and wide range of devices, all COPS microcontrollers have a number of I/O options, specified at the same time as the program, which allow the user to tailor, within limits, the I/O characteristics of the microcontroller to the system. Thus, the user can optimize the microcontroller for the system, thereby achieving maximum capability and minimum cost.

This manual deals with the basic functionality of COPS microcontrollers. It does not address electrical differences among the various devices. Thus, this manual does not distinguish between the COP400 and the COP300 series. These two series differ only in electrical characteristics and not in function. This manual further does not distinguish the high-speed devices from the low-power devices or from the CMOS devices except to the extent that some of the devices may have features that affect programming.

1.2.2 COPS ROMless Microcontrollers

Several COPS microcontrollers are designed to use external program memory. Basically, these devices have been created by removing the ROM from their single-chip counterparts. These devices are primarily intended to be used in program development and debug, device emulation, and low-volume production. Table I provides a list of COP400 ROMless devices currently available or in design. The devices are designed so that each COPS microcontroller has at least one ROMless device that can be used for accurate emulation. Since these devices are functionally equivalent to the single-chip microcontrollers, this manual does not generally distinguish the ROMless device from its single-chip counterpart.

1.2.3 COPS Single-Chip Microcontrollers

Table II provides a list of COPS single-chip microcontrollers currently available or in design. It is readily apparent that the list is quite extensive. Many of the variations are simply different packagings of the same device, e.g., the COP441 is the COP440 in a 28-lead package; the COP442 is the COP440 in a 24-lead package; the COP440 is a 40-pin device. Another important characteristic is the commonality of the pinouts of the single-chip devices; all 40-pin devices have the same pinout; all 24-pin devices have the same pinout; the COP411L and COP411C have the same pinout; the COP422 and COP422L have the same pinout. See Connection Diagrams *Figure 1* through *Figure 4*.

TABLE I. COPS ROMless Microcontrollers—General Software Overview

COP	401L	402	402M	404C	404L	404	
External ROM x 8	Up to 512	Up to 1024		Up to 2048			
RAM x 4	32	64		128		160	
Inputs	0	4		4		4	
Bidirectional TRI-STATE® I/O	8	8		8		16	
Bidirectional I/O	4	4		4		8	
Outputs	4	4		4		4	
Serial I/O and External Event Counter	Yes	Yes		Yes		Yes	
Internal Time Base Counter	No	Yes		Yes		Yes	
Time Base Counter Programmable	No	No		Yes	No	Yes	
Interrupt	No	Yes	No	Yes		Yes — 4 Sources	
Stack Levels	2	3		3		4	4 per CPU
Microbus™ Option	No	No	Yes	Yes	No	Yes	
Instruction Cycle (μs) min–max	15–40	4–10		4–DC	15–40	4–10	
Package Size (Pins)	40	40		48	40	48	
Availability	Now	Now		Now	Now	Now	

*These devices are NOT available as of this writing. The information on these devices is preliminary and subject to change. Advance information has been provided for completeness and as an aid to the user. Announcements will be made by National Semiconductor at the appropriate times regarding the availability and ultimate charcteristics of these devices.

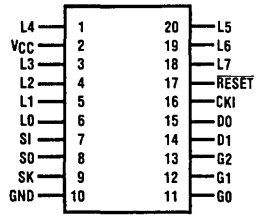
TABLE II. COPS Microcontrollers—General Software Overview

COP	410L	410C	411L	411C	413L	413C	420	420L	424C	421	421L	425C	422	422L	426C	444L	444C	445L	445C	440	441	442
ROM x 8	512						1024									2048						
RAM x 4	32						64									128				160		
Inputs	0						4			0						4		0		4		0
Bidirectional TRI-STATE I/O	8						8									8				16		8
Bidirectional I/O	4		3		4		4			4			2			4				8		4
Outputs	4		2		No		4			4			2			4				4		
Serial I/O and External Event Counter	Yes						Yes									Yes				Yes		
Internal Time Base Counter	No						Yes									Yes				Yes		
Time Base Counter Programmable	No						No		Yes	No		Yes	No		Yes	No	Yes	No	Yes	Yes		
Interrupt	No						Yes			No						Yes		No		Yes 4 Sources		Yes 2 Sources
Stack Levels	2						3									3				4		
MICROBUS Option	No						Yes	No	Yes	No						No	Yes	No		Yes		No
Instruction Cycle (μs) min–max	16–40	4–DC	16–40	4–DC	16–40	4–DC	4–10	16–40	4–DC	4–10	16–40	4–DC	4–10	16–40	4–DC	16–40	4–DC	16–40	4–DC	4–10		
Package SRE (Pins)	24		20				28			24			20			28		24		40	28	24
Availability	Now						Now			Now			Now			Now		Now		Now		

*These devices are not available as of this writing, the information on these devices is preliminary and subject to change. Advance information has been provided for completeness and as an aid to the user. Announcements will be made by National Semiconductor at the appropriate time regarding the availability and ultimate characteristics of these devices.

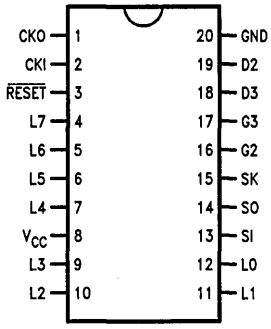
Connection Diagrams

COP411L/411C



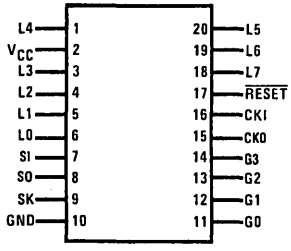
TL/DD/8800-1

COP422/422L/426C



TL/DD/8800-2

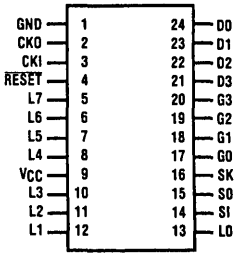
COP413L/413C



TL/DD/8800-3

FIGURE 1

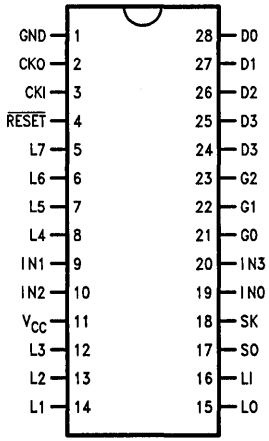
COP410L/410C/421/421L/425C
COP445L/445C/442



TL/DD/8800-4

FIGURE 2

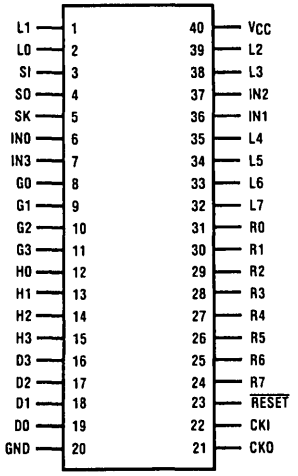
COP420/420L/424C/444L
COP444C/441



TL/DD/8800-5

FIGURE 3

COP440



TL/DD/8800-6

FIGURE 4

1.2.4 Conclusion

COPS microcontrollers comprise a broad, general purpose, powerful, and flexible family of devices. The hardware and software compatibility of the devices allow the user to move easily within the family as the need arises or the application dictates. Many ROMless devices are available to aid in emulation and development. The applications of COPS devices are unlimited. COPS microcontrollers have been used in automotive (trip computer, seat position controller, electronic instrument cluster, ignition systems, diagnostic systems), appliance (ovens, microwave ovens, vacuum cleaners, sewing machines, washers, dryers, food processors), home electronic (electronically tuned radios, cassette recorders, video cassette recorders, stereo systems), security system, timekeeping, energy management, industrial/commercial (utility meters, keyboard encoders, cash registers, dictation equipment, coin changers, vending machines, jukeboxes), telephone (repertory dialers, simple phone dialers, call timers), exercise equipment (exercise bicycle, jogging machine), miscellaneous home (garage door openers, lawn sprinklers, Christmas ornaments, cable television), toy, game, and many other applications.

2.0 Architecture of COPS Microcontrollers

2.1 INTRODUCTION

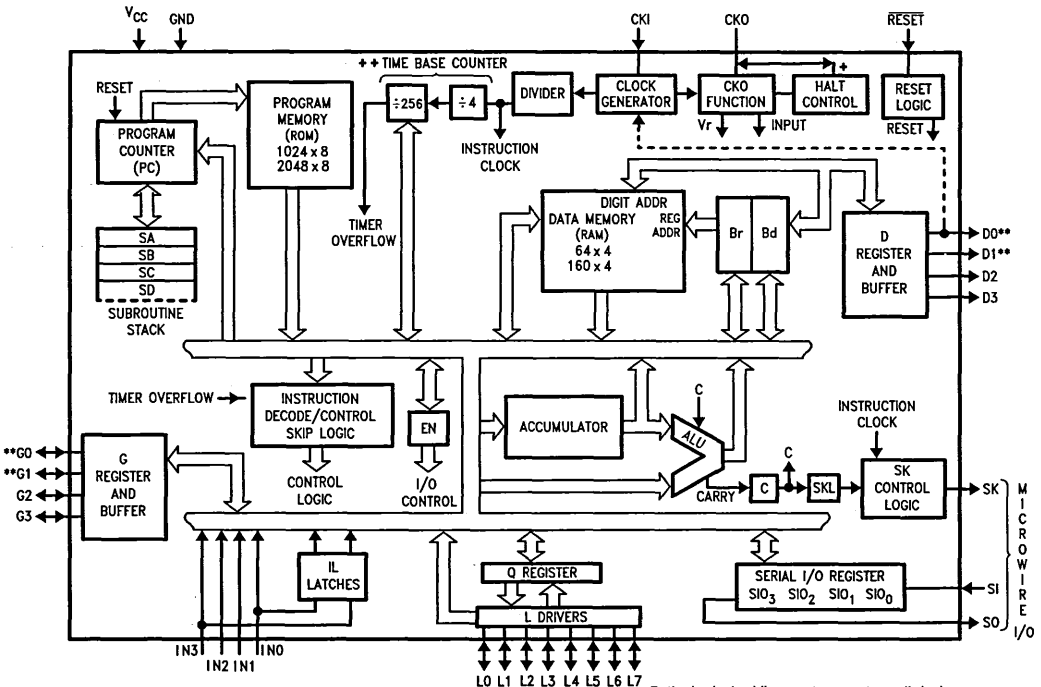
This section deals with the architecture of COPS microcontrollers. *Figure 2-1* is the generic block diagram for COPS microcontrollers. The diagram is accurate as is for the COP420/421/422, COP420L/421L/422L, COP424C/425C/426C, COP444L/445L, COP444C/445C, and COP484/485 devices. The addition or deletion of certain elements creates the other microcontrollers in the COPS family. *Figure 2-2,* the block diagram of the COP410L/411L/413L/413C and COP410C/411C, *Figure 2-3,* the block diagram of the COP440/441/442 illustrate this fact. It is clear, even from a cursory examination, that all COPS microcontrollers possess the fundamental architecture that is indicated in *Figure 2-1.* Therefore, *Figure 2-1* is the focal point for the discussion of the COPS architecture. The additions or deletions that lead to the other block diagrams are discussed where appropriate.

2.2 COPS MEMORY STRUCTURE

2.2.1 Program Memory—ROM

The program memory in COPS microcontrollers is a read-only (ROM) organized as a number of eight-bit words. COPS microcontrollers with ROM capacities of 512, 1024, and 2048 words are presently available. Devices with ROM capacity of 3072 and 4096 words are currently in design. The ROM words are addressed sequentially by a binary program counter (in ROMless devices, the program counter is brought out to pins to address external memory). The program counter starts at zero and, if there are no jumps or subroutines or table lookups, will increment to the maximum value possible for the device and rolls over to zero and begins again.

Internally, COPS microcontrollers have a semi-transparent page, block, and chapter structure to the ROM. A page is composed of 64 contiguous ROM words. The lower six bits of the program counter are zeroes at the first address of a page and ones at the last address of a page. A block, which is significant only in the table lookup and indirect jump operations, is composed of four contiguous pages (256 contiguous ROM words). The lower eight bits of the program counter are zeroes at the first address of a block and ones at the last address of a block. The first address of a block is also the first address of a page and the last address of a block is also the last address of a page. The chapter division is relevant only in COPS devices with more than 2048 ROM words or ROMless devices capable of addressing more than 2048 ROM words. The lower 11 bits of the program are zeroes at the first address of a chapter and ones at the last address of a chapter. The first address of a chapter is also the first address of a block and the last address of a chapter is also the last address of a block. Table III lists the hexadecimal address and the corresponding page/chapter/block divisions.



TL/DD/8800–7

Paths in dashed lines not present on all devices.

+ SD register available only on 36- and 48-lead devices.

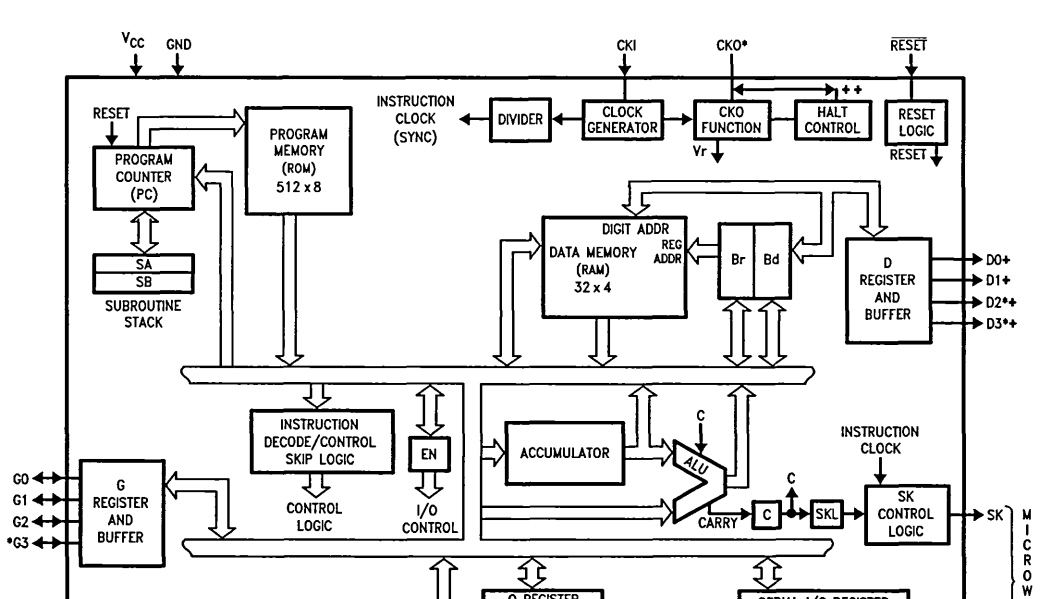
*These pins not available on 24- and 28-lead devices.

**These pins not available on 20-lead devices.

+ Only CMOS COPS except 411C.

+ + Not available on ½k devices.

FIGURE 2-1. Basic Block Diagram for COPS Microcontrollers



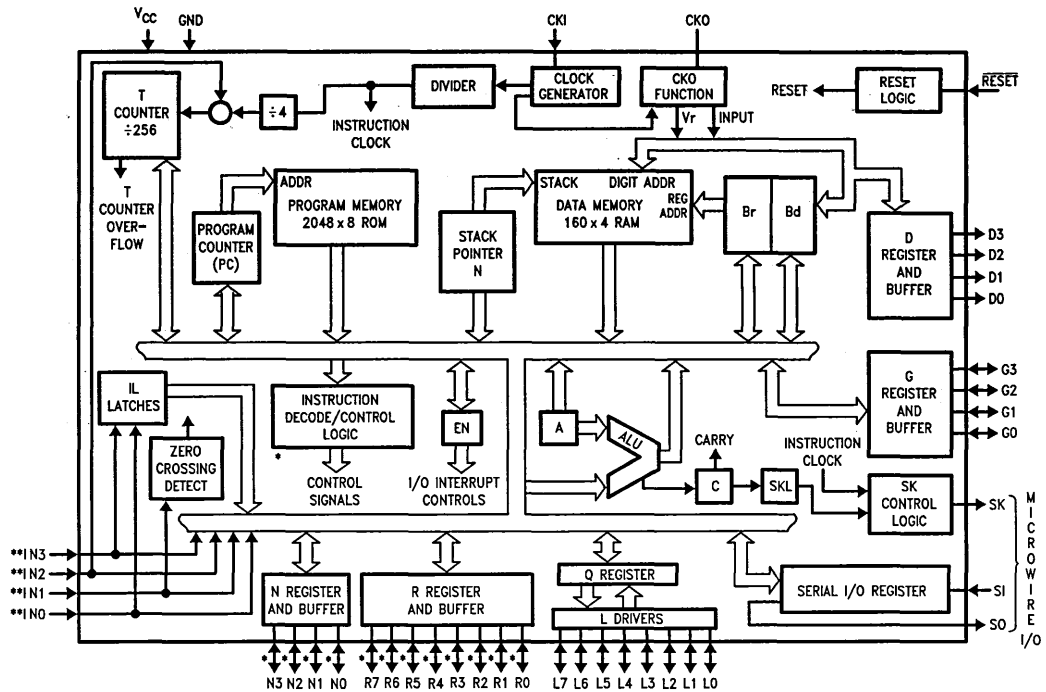
TL/DD/8800-8

*These pins not available on COP410L and 411C.

+ These pins not available on the COP413L/COP413C.

++ Only CMOS COPS except 411C.

FIGURE 2-2. COP410L/411L/413L and COP410C/411C Block Diagram



TL/DD/8800–9

*These pins not available on 28- or 24-lead devices.

**These pins not available on 24-lead devices.

FIGURE 2-3. COP440/411/442 Microcontrollers Block Diagram

TABLE III. Address-Page-Block-Chapter Mapping

Hex Address	Page	Block	Chapter
000-03F 040-07F 080-0BF 0C0-0FF	0 1 2 3	0	
100-13F 140-17F 180-1BF 1C0-1FF	4 5 6 7	1	
200-23F 240-27F 280-2BF 2C0-2FF	8 9 10 11	2	
300-33F 340-37F 380-3BF 3C0-3FF	12 13 14 15	3	0
400-43F : 4C0-4FF	16 : 19	4	
500-53F : 5C0-5FF	20 : 23	5	
600-63F : 6C0-6FF	24 : 27	6	
700-73F : 7C0-7FF	28 : 31	7	
800-83F : 8C0-8FF	32 :	8	
900-93F : 9C0-9FF	36 : 39	9	
A00-ACF : AC0-AFF	40 : 43	10	
B00-B3F : BC0-BFF	44 : 47	11	
C00-C3F : CC0-CFF	48 : 51	12	1
D00-D3F : DC0-DFF	52 : 55	13	
E00-E3F : EC0-EFF	56 : 59	14	
F00-F3F : FC0-FFF	60 : 63	15	
1000-103F : .	64 : .		

This internal structure is semi-transparent. Only some jumps, some subroutine calls, and table lookups are affected by this structure. As indicated earlier, the block divisions come into play only in the table lookups and indirect jumps. The page and chapter divisions affect some direct jumps and subroutine calls. Section 4 explains the effects of these divisions on the pertinent instructions. Complete operational programs can be written without consideration of this internal structure. Such a program, however, will use more code, and therefore require larger ROM capacity, than a program written with this structure in mind. Section 4 will address this in greater detail. This page/block/chapter structure has no effect on the program counter. **The binary program counter will freely increment through page, block, or chapter boundaries.**

2.2.2 Data Memory—RAM

The data memory (RAM) in COPS microcontrollers is organized as a matrix. Each row in the matrix is called a register; each column in the matrix is called a digit. A digit is 4 bits wide. As shall be seen, this particular structure contributes to the general efficiency of COPS microcontroller. All RAM addressing is based on this register-digit (or row-column) organization. The RAM address register identifies a specific digit in the RAM matrix. COPS devices with RAM sizes of 32 digits (4 registers by 8 digits, 128 bits), 64 digits (4 registers by 16 digits, 256 bits), 128 digits (8 registers by 16 digits), and 160 digits (10 registers by 16 digits) are presently available. A device with RAM sizes 256 digits (16 registers by 16 digits) is in design. A ROMless device with 512 digits (32 registers by 16 digits) of RAM is also in design.

The RAM in COPS microcontrollers is not in the program memory space. The RAM is not addressed by the program counter but has its own address register, the B register. The B register can be loaded directly or through the accumulator. Since the RAM has its own address register, most COPS instructions which access RAM do not contain an address field. This tends to promote ROM code efficiency. The B register is divided into two distinct parts: Br—the row or register address and Bd—the column or digit address. Bd is 4 bits wide in all COPS microcontrollers. Br is between 2 and 5 bits wide depending on the particular device. Bd, in addition to being the digit address, is the source for the D output register. On software command, the contents of Bd can be transferred to the D port where the information is latched.

The data memory digit addressed by the B register is normally accessed through the accumulator. The contents of the RAM digit may be directed, under software command, to one of several output ports as well as used in the normal program flow. Two instructions, LDD and XAD, carry a RAM address with them. These instructions operate (load or exchange) on the specified RAM digit without modifying the B register.

2.2.3 Subroutine Stack

COPS microcontrollers have a subroutine stack of two, three, or four save registers. On all COPS microcontrollers with two or three save registers in the subroutine stack, a physical transfer of register contents within the stack occurs on all operations affecting the stack, primarily calls and returns. On these devices, the stack is physically and logically separate from data RAM. The user does not have access to the stack and, therefore, may not read or write the stack in these devices.

On COPS devices with four or more stack levels, the stack is located in data RAM. Four stack levels use up one data register. The user has access to the stack since the data RAM contains the stack. However, in no case does the stack expand beyond its assigned area into the rest of the data memory. These devices contain a stack pointer which is incremented or decremented on operations affecting the stack. Overflowing the stack merely causes the stack pointer to wrap around from its maximum value back to zero. On the COP440 series, **only** the user also has access to the stack pointer and may read or write the pointer. In all of these devices which permit stack access, the programmer has increased versatility. However, caution is recommended. Increased power brings with it increased risk, and the programmer should exercise care that the stack is not accidentally accessed in these devices.

2.3 THE ARITHMETIC LOGIC UNIT

The arithmetic logic unit (ALU) in COPS microcontrollers is a 4-bit parallel binary adder. It performs all the arithmetic and logic functions in the microcontrollers. The destination for all such operations is the 4-bit accumulator, and one input to the ALU is always the accumulator. The other input is either an immediate operand as specified by an instruction or, more commonly, the data RAM digit addressed by the B register. The one-bit C register sometimes is a third input to the ALU. The ALU outputs a carry bit which, depending on the instruction being executed, can be loaded into the C register. See the instruction set description and Section 4.4 for more details on carry and the C register.

2.4 INPUT/OUTPUT

2.4.1 Inputs

Only one input port, the IN port, is available on COPS microcontrollers. This port is available only on devices with 28 or more leads. On software command, the four IN lines are read, as a group, into the accumulator. In addition to the direct inputs, IN0 and IN3 have latches associated with them. These latches capture a high to low transition on the particular line. The status of the latches is read into the accumulator on software command. Thus, the programmer can read the present status of the IN lines directly or can read the status of the latches associated with IN0 and IN3.

The IN1 input can, under software control, serve as an interrupt input. The enabling or disabling of interrupts is a software decision. As such, in a given program, interrupts may be always enabled, never enabled, or sometimes enabled. On the COP440/441 devices only, IN1 may be mask programmed to be a zero crossing input. As such, interrupts may be generated at each zero crossing. Note that the zero crossing option is a mask, i.e., hardware option and not a software option.

On the new COP424C and the COP444C, IN2 may be mask programmed to be an input to the time base counter. Again, this is a hardware option and is not software alterable. On the COP440/441 devices, IN2 may also be selected as an input to the time base counter. On these devices, however, the choice is controlled in software by the programmer.

2.4.2 Bidirectional TRI-STATE I/O

All COPS microcontrollers have at least one eight-bit bidirectional I/O port. This is the L port. In output operations, the L lines output the contents of the eight-bit Q register. The input path is from the pins to the accumulator and RAM. Note that the L lines are drives only: they do not retain any data. Output data for the L port is stored in the Q register. The L drivers can be placed in the high impedance, or TRI-STATE mode for ease in interface to a system bus.

The COP440 has an additional eight-bit bidirectional I/O port, the R port. The R port contains latches and drivers. Data to be output is latched into the R register. The input path is from the pins to the accumulator and RAM. Input data at the R pins is not, and cannot be, latched into the R register by any external signal. This must be done indirectly by the program. The R drivers, like the L drivers, can be put into a high impedance, or TRI-STATE, condition for simple bus interface.

Both the L port and the R port can be inputs. There is no input state per se. If used as inputs, either port may be put into a high impedance, or TRI-STATE, condition. In this case, the external signal must drive the line both high and low and guarantee the valid "O" and "1" logic levels. Alternatively, for both ports, the Q register or the R register can serve as a pullup for the L and R lines respectively. The programmer may write "1's" to the input positions and enable the drivers. In this case, the external signal need only pull the line down to a valid low level.

2.4.3 Bidirectional I/O

The G port is a four-bit bidirectional I/O port. The G outputs are latches and drivers. Therefore, data can be saved in the G port. The input path is from the pins to the accumulator. In addition to reading the port, the G lines can be directly tested, either individually or as a four-bit group, in software. Note, the latches on G are for output only; input signals are not latched into the G port.

The COP440 has an additional bidirectional four-bit port, the H port. The H port is essentially a duplicate of the G port except that H cannot be directly tested.

There is no restriction on H or G as to which lines may be inputs or outputs. All G lines may be inputs; all G lines may be outputs; any G line, or group of G lines, may be outputs with the remaining G lines inputs. The same is true of the H lines.

2.4.4 Outputs

The D port is an output-only port. The outputs are latched. On software command, the contents of Bd, the digit address portion of the RAM address register, are copied to the D port. These outputs will remain in that state until the next write to D. The D port is loaded only from Bd.

2.4.5 The SIO Register

The SIO register is a dual-purpose four-bit register. Depending on the status of the EN register, whose contents are user alterable, this register may be a four-bit binary down coutner or a four-bit serial shift register. When SIO is a down counter, SI is the counter input, the counter decrements on the high to low transition, provided that the input remains low for two instruction cycles of the signal at the SI output. SO and SK are logic level outputs which can be directly controlled by the program. When SIO is a shift register, SI is the input to the 4-bit shift register and SO is the shift register output. SK is a serial clock running at the instruction cycle

rate. By means of the EN register, and while SIO remains enabled as a shift register, SO can be disabled, i.e., forced to zero. Similarly SK can also be forced to zero in this mode. Note that when SIO is enabled as a shift register and SO enabled as a shift register output, whatever is at SI will appear at SO four instruction cycles later unless the program alters the contents of SIO. When enabled as a shift register, SIO is always shifting at the instruction cycle rate regardless of the status of SO or SK.

MICROWIRE™ I/O

The MICROWIRE concept provides a simple, easy to use serial interface between COPS microcontrollers and various peripheral devices. The MICROWIRE interface is, essentially, the serial I/O port on COPS microcontrollers, the SIO register in the shift register mode. SI is the shift register input, the serial input line to the microcontroller. SO is the shift register output, the serial output line to the peripherals. SK is the serial clock, data is clocked into or out of peripheral devices with this clock. MICROWIRE is available on all COPS microcontrollers.

MICROWIRE PERIPHERALS

For MICROWIRE interface, a peripheral device requires some or all of the following:

DI Data Input. This is the serial input to the peripheral. This is connected to SO on the microcontroller. All MICROWIRE peripherals **must** have this pin.

SK Serial Clock. This is the serial clock connected to SK of the microcontroller. All MICROWIRE peripherals **must** have this pin.

CS Chip Select. This merely selects a particular device. It may be connected to any convenient microcontroller output. Chip Select is required in any multiple peripheral systems. In a single peripheral system, whether or not Chip Select must be connected to a microcontroller output depends on the peripheral itself and its design.

DO Data Output. This is the serial output from the peripheral. It is connected to SI of the microcontroller. DO is required only on peripherals that communicate back to the microcontroller.

2.4.6 Microbus

Microbus is a universal eight-bit parallel system bus. Certain COPS microcontrollers have a mask option permitting them to be used as Microbus-compatible peripheral devices. As far as the COPS device is concerned, the Microbus is composed of the following elements:

- An eight-bit bidirectional data bus
- Data Strobes - a read strobe and write strobe
- Chip Select - to identify the device
- Interrupt/Acknowledge - return line to main CPU

In COPS microcontrollers, the data bus is the Q register-L drivers combination. If the device is selected and a write strobe occurs, data is transferred from the bus-L directly into the Q register. Similarly, if the device is selected and a read strobe occurs, data is copied from the Q register onto the bus-L. Input IN_1 becomes $\overline{RD}$, the read strobe. Input IN_2 becomes $\overline{CS}$, the chip select. Input IN_3 becomes $\overline{WR}$, the write strobe. Note that these three inputs are all active low. A logical "0" on $\overline{CS}$ (IN_2) selects the COPS device and enables operation of $\overline{RD}$ and $\overline{WR}$. A logical "0" on $\overline{RD}$ (IN_1) or $\overline{WR}$ (IN_3) when $\overline{CS}$ is also a logical "0" will cause the data read or write as described above. I/O pin G0 serves as an interrupt/acknowledge or ready pin back to the main CPU. G0 is normally high-ready. It is set high by the user program. The occurrence of a write strobe while the device is selected automatically sets G0 to the low or busy state. The user program sets G0 high again.

The Microbus option on COPS microcontrollers is completely compatible with the Microbus standard. The timing and timing relationships are those defined by that standard.

The Microbus option is a mask option, i.e., a hardware option. The functions of IN_1, IN_2, IN_3, G0, and L drivers and the Q register are physically altered by this option. The Microbus option is available on the following COPS microcontrollers only: COP420, COP424C, COP444C, COP440 and COP441.

2.5 THE ENABLE REGISTER

The ENABLE (EN) register is an internal four- or eight-bit loaded under program control. The state of the individual bits of this register selects or deselects certain features in the microcontroller.

2.5.1 EN_0 through EN_3

These four bits of the EN register are present on all COPS microcontrollers. Their function is as follows:

EN_0, the least significant bit of the enable register, controls the status of the SIO register. With EN_0 set, a logical "1", the SIO register is a four-bit asynchronous binary down counter decrementing its value by one upon each low going pulse at the SI input. The pulse must be low at least two instruction cycles. With EN_0 equal to "1", SO and SK are logic signals. SK outputs the value of SKL. SO outputs the value of EN_3. With EN_0 reset (low), the SIO register is a four-bit serial shift register that shifts left, from SI toward SO, one bit each instruction cycle time. Data is shifted into the least significant bit of SIO from SI. SO can be enabled to output the most significant bit of SIO. When EN_0 reset, SK becomes a logic controlled clock whose period is the instruction cycle time.

EN_1 controls the interrupt. With EN_1 set, the interrupt is enabled. If a signal meeting the timing requirements appears at the interrupt input when EN_1 is set, the interrupt will be recognized. With EN_1 reset, the interrupt is disabled, the signal at the interrupt input is ignored. Obviously, the status of EN_1 is significant only in those COPS microcontrollers having interrupt capability.

EN_2 controls the L drivers. With EN_2 set, the L drivers output the data in the Q register to the L I/O port. With EN_2 reset, the L drivers are disabled thereby placing the L I/O port into a high impedance, or TRI-STATE, condition. EN_2 has no effect on the L drivers in devices that have the Microbus option implemented.

On the COP440 and COP441, devices which have the Microbus option selected, EN_2 serves a different function. In this case, EN_2 set will disable any writing, by the program, into G0 which is the ready signal back to the main CPU.

EN_3, in conjunction with EN_0, controls the SO output. As stated above, if EN_0 is set, SO outputs the value of EN_3. If EN_0 is reset and EN_3 is set, SO is the output of the SIO serial shift register. If EN_0 is reset and EN_3 is reset SO is set to logical "0". SIO remains a shift register shifting data in from SI; SO is merely held low by internal logic. Table IV provides a summary of the SIO modes associated with EN_0 and EN_3.

TABLE IV. Effects of EN_3, EN_0, on SIO, SI, SO, and SK

EN_3	EN_0	SKL	SIO	SI	SO	SK
0	0	0	Shift Register	Input to Shift Register	0	0
0	0	1	Shift Register	Input to Shift Register	0	Clock
1	0	0	Shift Register	Input to Shift Register	Serial Out	0
1	0	1	Shift Register	Input to Shift Register	Serial Out	Clock
0	1	0	Binary Down Counter	Input to Binary Counter	0	0
0	1	1	Binary Down Counter	Input to Binary Counter	0	1
1	1	0	Binary Down Counter	Input to Binary Counter	1	0
1	1	1	Binary Down Counter	Input to Binary Counter	1	1

Note: SKL not affected by EN_3 or EN_0, but SKL does affect SK status.

2.5.2 EN_4 through EN_7

These "extra" four bits of the enable register are present only in the following devices: COP440, COP441 and COP442. Obviously, therefore, the information in this section applies to those devices only.

EN_4—In conjunction with EN_5, EN_4 selects the interrupt source. See Table V.

EN_5—In conjunction with EN_4, EN_5 selects the interrupt source. See Table V.

EN_6—With EN_6 set (high), IN_2 becomes the input to the internal eight-bit T counter. With EN_6 reset (low), the input to the eight-bit T counter is the output of a divide by 4 prescaler from the instruction cycle frequency, thus providing a ten-bit time base counter.

On the COP442 IN_2 is not available as an input. Therefore, on these devices, EN_6 functions as a T counter disable: EN_6 set disables further counting, and EN_6 reset produces the ten-bit time base counter.

EN_7 controls the R I/O port. With EN_7 set, the contents of the R register are output to the R I/O port. With EN_7 reset, the R I/O port is placed into a TRI-STATE, or high impedance, condition. The contents of the R register are not affected.

TABLE V. Interrupt Source Selection

EN_5	EN_4	Interrupt Source
0	0	IN_1 - Low Going Pulse
0	1	CKO Input (If CKO Input Option Mask Programmed)
1	0	Zero Crossing on IN_1 (or IN_1 Level Transition)
1	1	T Counter Overflows

2.6 INTERNAL TIMER

All COPS microcontrollers except the COP410L, COP411L, COP413L, COP413C, COP410C, and COP411C have an internal time base counter. This counter is in the form of a ten-bit counter with the input being the instruction cycle frequency. Thus, this counter divides the instruction cycle frequency by 1024 or overflows once every 1024 instruction cycle times. A timer latch is set every time the counter overflows. This latch may be tested and reset (a single instruction) by the user's program.

2.6.1 Access to the Timer

All COPS microcontrollers that have the time base counter have the ability to test and reset the timer latch. Some devices, however, also have the ability to read and write the upper eight bits (the T counter) of the timer. The devices with this capability are as follows: The COP424C, COP425C, COP426C, COP444C/445C, COP440/441/442, and their associated ROMless devices. The timer overflow latch is still present and is still set when the counter overflows. These devices allow the user to modify, under program control, the overflow rate of the time base counter.

2.6.2 External Event Counter

On some devices, the COP424C, COP444C, COP440/441, the upper eight bits or the T counter of the time base counter may be disconnected from the instruction cycle clock and connected to input IN_2. In this mode, the T counter counts external pulses. The timer overflow latch is set whenever the T counter overflows. The latch is tested in the normal manner. This characteristic is a mask option on the COP424C and COP444C devices. Thus, on these devices, the T counter may be connected to form the ten-bit time base counter or the T counter may be connected to IN_2 to count external events. On the COP440/441 devices, this characteristic is a software option. The user's program controls the connection of the T counter via EN_6. There is no restriction, in these devices, on changing the T counter connection during program execution. The user is free to alternate between a time base counter and an external event counter if doing so is useful in his or her application.

2.7 OSCILLATOR AND BASIC TIMING

2.7.1 Clock Generator and Divider

The clock generator on COPS microcontrollers is extremely versatile and, by means of mask options, will work with a variety of oscillators: crystal, external, simple RC, or more involved RC, RLC, or LC networks. Furthermore, the clock generator will usually operate over a fairly large range in order to give the user maximum flexibility in selecting the oscillator frequency. Several divider (prescaler) options are available, as mask options, to insure that the COPS microcontroller is operating within its valid range with the oscillator frequency being used. See the various device data sheets for precise details regarding the oscillator frequency, clock generator, and divider.

2.7.2 The Instruction Cycle

The instruction cycle frequency is the frequency after the divider or prescaler. The period of this frequency, or the instruction cycle time, is the basic timing reference in COPS

microcontrollers. Minimum pulse widths, for counter inputs, interrupt, etc., are expressed in terms of instruction cycle times. The highest degree of resolution with which a COPS microcontroller can read input pulses or generate output pulses is the instruction cycle time.

The instruction cycle time or frequency can be measured by the user. The period of the SK output when the microcontroller is reset (RESET low) or when SK is enabled as a clock output is the instruction cycle time.

2.8 INITIALIZATION

On power up, providing the timing parameters in the data sheets are met, the following registers are cleared on all COPS microcontrollers: A, B, C, D, EN, G, and the program counter PC. The SK latch, SKL, is set on all devices. In addition the T counter is cleared on the COP440 series, the COP424C series, and the COP444C series devices. In the COP440 the IL register and the Q register are also cleared. (Note that these two registers are not cleared on other devices.) The R, H, and N registers are also cleared on reset in the COP440.

Reset, or initialization, occurs on power up and whenever a logical "0" at least three instruction cycles wide appears at the $\overline{\text{RESET}}$ input. On the COP440 and the COP424C/COP444C series devices, the T counter is cleared within these three instruction cycle times. On other COPS microcontrollers, the logical "0" at the $\overline{\text{RESET}}$ input must be ten cycles wide to clear the time base counter. In this situation, the timer overflow latch is set.

The reset condition of COPS microcontrollers is as follows:

- The program counter, PC, is set to 0.
- The accumulator, A, is 0.
- The RAM address register, B, is set to 0,0.
- The carry, C, is set to 0.
- The D register and D output port are set to 0.
- The enable register is set to 0 and SKL set to 1.

1. SIO is a shift register.
2. SI is shift register input.
3. SO is 0.
4. SK is clock output.
5. Interrupts are disabled.
6. The L port is put into a high impedance, or TRI-STATE condition.

- The G output port is set to 0.
- On the COP440/COP2440 series, the following is also true:

1. The Q register is set to 0.
2. The H register and I/O port are set to 0.
3. The R register is set to 0.
4. The R I/O port is put into a high impedance, or TRI-STATE condition.
5. The interrupt source is IN_I, low-going pulse.
6. The T counter is cleared and connected to form the time base counter.
7. The IL register is set to 0.

3.0 The COPS Instruction Set

3.1 BASIC CHARACTERISTICS

The instruction set of COPS microcontrollers is designed to take maximum advantage of the COPS dual-bus architecture. The COPS instruction set, merged with the COPS architecture, provides the user with the power, versatility, and efficiency to achieve the maximum function and capability in minimum memory.

Since COPS microcontrollers are not memory-mapped devices, most instructions do not have the burden of carrying some form of address field. Therefore, most instructions are one byte in length. This, in turn, increases program efficiency. ROM space is devoted to performing a function rather than pointing to the locations of various items.

It is quite common for a COPS instruction to contain a multiplicity of function. This obviously creates program efficiency by performing in a single instruction a number of functions that would otherwise require several instructions.

The test instructions, like most COPS instructions, do not contain an address. Therefore, a successful test causes the text instruction to be skipped. It is quite common for one or both of the instructions following the test to be jumps. More importantly, however, this skipping characteristic allows the programmer to do a number of "unusual" things. Tests without following jumps are common. B or A or other parameters can be altered in line without jumping by judicious use of the test instructions. Examples of this, and further details, are provided in Section 4. Furthermore, the skip feature has been "built into" a number of arithmetic functions thereby eliminating the need to make separate tests.

3.2 DETAILED INSTRUCTION DESCRIPTION

For purposes of discussion and explanation, the COPS instructions are loosely grouped into the following six categories:

1. Arithmetic/Logic Instructions
2. Transfer of Control Instructions
3. Memory Reference Instructions
4. Register Reference Instructions
5. Test Instructions
6. Input/Output Instructions

This section provides a detailed description of all COPS instructions. This description includes the following information:

- The instruction mnemonic.
- A written description of the instruction.
- The data or program flow associated with the instruction.
- The instruction opcode in hex and binary.
- The instruction execution time—expressed in instruction cycles.
- Skip conditions associated with the instruction.
- Any restrictions on the instruction or its use; any "special effects" of the instruction.
- The COPS microcontrollers which have or do not have the instruction.

For ease and simplicity of description, the COPS microcontrollers are divided into the following four groups:

Group 1 devices: COP401L, COP410L, COP411L, COP413L, COP413C, COP410C, COP411C

Group 2 devices: COP402, COP402M, COP404L, COP404C, COP420, COP421, COP422, COP420L, COP421L, COP422L, COP424C, COP425C, COP444L, COP445L, COP444C, COP445C

Group 3 devices: COP404, COP440, COP441, COP442

The following list defines the symbols used in the descriptions of the instructions.

A	4-bit accumulator
B	RAM address register
Br	Upper bits of B, register address
Bd	Lower 4 bits of B, digit address
C	1-bit carry register
D	4-bit data output port
EN	Enable register
G	4-bit register to latch data for G I/O port
H	4-bit register to latch data for H I/O port
IL	Two 1-bit latches associated with IN3 and IN0 inputs
IN	4-bit input port
IN_1 Z	Zero crossing input
L	8-bit TRI-STATE I/O port
M	4-bit contents of RAM addressed by B
N	Subroutine stack pointer
PC	ROM address register, program counter
Q	8-bit register to latch data for L I/O port
R	8-bit register to latch data for R I/O port
SIO	4-bit shift register and counter
SK	Logic controlled clock output
T	8-bit binary counter register
RAM(B)	4-bit contents of RAM addressed by B
RAM_N	Contents of RAM location addressed by stack pointer N
ROM(t)	Contents of ROM location addressed by t
$PC_{a:b}$	Bits a through b of program counter PC

3.2.1 Arithmetic/Logic Instructions

ASC

Binary add, with carry, the accumulator with the memory location specified by the B register. The result is placed in the accumulator. If result > 15_{10}, generate a skip.

A ← A+C+RAM(B) C: Set or reset according to carry from bit three

Hex Code	7	6	5	4	3	2	1	0
30	0	0	1	1	0	0	0	0

Execution Time: 1 Instruction Cycle
Skip Conditions: If 1 → C, skip
Restrictions: None
Availability: All COPS microcontrollers

ADD

Binary add the accumulator with the memory location specified by the B register. Result is placed in the accumulator.

A ← A+RAM(B) C: Not used or affected

Hex Code	7	6	5	4	3	2	1	0
31	0	0	1	1	0	0	0	1

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

ADT

Binary add 10 to the accumulator. Instruction used for decimal adjust.

A ← A+10_{10} C: Not used or affected

Hex Code	7	6	5	4	3	2	1	0
4A	0	1	0	0	1	0	1	0

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: None
Availability: Not available on: Group 1 devices

AISC y

Binary add the immediate value y to the accumulator and place the result in the accumulator. Generate a skip if there is a carry out of bit 3.

A ← A+y C: Not used or affected

Hex Code	7	6	5	4	3	2	1	0
5y	0	1	0	1	y_3	y_2	y_1	y_0

Execution Time: 1 Instruction Cycle
Skip Conditions: If carry from bit 3, skip
Restrictions: $y \neq 0$, $0 < y \leq FH$
Availability: All COPS microcontrollers

CASC

Binary add, with carry, of the one's complement of the accumulator with the data in the memory location specified by the B register. Generate a skip if result >15_{10}. This is the basic subtract instruction.

A ← $\overline{A}$+RAM(B)+C C: Set or reset according to carry from bit three

Hex Code	7	6	5	4	3	2	1	0
10	0	0	0	1	0	0	0	0

Execution Time: 1 Instruction Cycle
Skip Conditions: If 1 → C, skip
Restrictions: None
Availability: Not available on: Group 1 devices

CLRA

Clear the accumulator.

A ← 0 C: Not affected

Hex Code	7	6	5	4	3	2	1	0
00	0	0	0	0	0	0	0	0

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

COMP

Replace the value in A with its one's complement.

A ← $\overline{A}$ C: Not affected

Hex Code	7	6	5	4	3	2	1	0
40	0	1	0	0	0	0	0	0

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

NOP

No operation. C: Not affected

Hex Code	7	6	5	4	3	2	1	0
44	0	1	0	0	0	1	0	0

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

OR

Logical OR of accumulator with contents of memory location specified by the B register. Result in accumulator.

A ← A v RAM(B) C: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
1A	0	0	0	1	1	0	1	0

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: None
Availability: Group 3 devices

RC

Reset/clear the one-bit carry register.

C ← 0 A: Not affected

Hex Code	7	6	5	4	3	2	1	0
32	0	0	1	1	0	0	1	0

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

SC

Set the one-bit carry register.

C ← 1 A: Not affected

Hex Code	7	6	5	4	3	2	1	0
22	0	0	1	0	0	0	1	0

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

XOR

Exclusive OR, bit by bit, of accumulator with contents of memory location specified by the B register. Result placed in accumulator.

A ← A ⊕ RAM(B) C: Not affected

Hex Code	7	6	5	4	3	2	1	0
02	0	0	0	0	0	0	1	0

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

3.2.2 Transfer of Control Instructions

JID

Jump Indirect. This involves a two-step modification of the program counter. First, load the lower eight bits of the program counter with the contents of the accumulator (upper four bits) and the memory location specified by the B register. The data addressed by this modified program counter is then loaded into the lower eight bits of the program counter. Execution continues at this second address.

(1) PC ← PC+1 C: Not affected
(2) $PC_{7:0}$ ← A,RAM(B) A: Not affected
(3) PC ← PC+1
(4) $PC_{7:0}$ ← ROM($PC_{10:8}$,A,RAM(B))

Hex Code	7	6	5	4	3	2	1	0
FF	1	1	1	1	1	1	1	1

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: JID at last word of block looks to next block for vector addresses (step one above). Vector address at last word of block points into next block (Step 3, 4 above).
Availability: All COPS microcontrollers

JMP a

Jump Direct. Load the program counter (lower 11 bits) with the address specified in the instruction. Continue program execution at this address.

$PC_{10:0}$ ← a C: Not affected
A: Not affected

Hex Code	7	6	5	4	3	2	1	0
6–	0	1	1	0	0	a_{10}	a_9	a_8
–	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: $a_{10}=0$, $a_9=0$ Group 1 devices
$a_{10}=0$ in 1k devices
JMP in last two words of chapter jumps to next chapter
Availability: All COPS microcontrollers

JP a

Jump within Page.

(1) PC ← PC+1 C: Not affected
(2) $PC_{6:0}$ ← a–pages 2, 3 only A: Not affected
or
(2) $PC_{5:0}$ ← a–all other pages

Hex Code	7	6	5	4	3	2	1	0
–	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0

(above for pages 2, 3 only)

1	1	a_5	a_4	a_3	a_2	a_1	a_0

(all other pages)

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: May not JP to last word of a page. JP in last word of a page jumps to next page (Step 1 above).
Availability: All COPS microcontrollers

JSRP a

Jump to subroutine within Page 2.

(1) PC ← PC+1
(2) SB ← SA ← PC Group 1 devices
or
(2) SC ← SB ← SA ← PC Group 2 devices
or
(2) RAM_N ← PC Group 3 and Group 4 devices
N ← N+1
(3) $PC_{5:0}$ ← a address within Page 2
$PC_{8:6}$ ← 010 load Page 2
$PC_{all\ other}$ ← 0

Hex Code	7	6	5	4	3	2	1	0
–	1	0	a_5	a_4	a_3	a_2	a_1	a_0

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: May not be used within Pages 2 and 3
May not JSRP to last word of Page 2
Availability: All COPS microcontrollers

JSR a

Jump to subroutine direct. Load lower 11 bits of the program counter with the address a. Push the subroutine stack. Continue execution at the address specified by the instruction.

(1) PC ← PC+2
(2) SB ← SA ← PC Group 1 devices
or
(2) SC ← SB ← SA ← PC Group 2 devices
or
(2) RAM_N ← PC; N ← N+1 Group 3 and Group 4 devices
(3) $PC_{10:0}$ ← a

Hex Code	7	6	5	4	3	2	1	0
6–	0	1	1	0	1	a_{10}	a_9	a_8
–	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: $a_{10}=0$, $a_9=0$ in Group 1 devices
$a_{10}=0$ in 1k devices
JSR in last two words of chapter calls subroutine in next chapter.
Availability: All COPS microcontrollers

RET

Return from subroutine and return control to the main program at the instruction following the JSR, or JSRP, or JSRL.

PC ← SA ← SB Group 1 devices
or
PC ← SA ← SB ← SC Group 2 devices
or
N ← N-1 Group 3 and Group 4 devices
PC ← RAM_N

Hex Code	7	6	5	4	3	2	1	0
48	0	1	0	0	1	0	0	0

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

RETSK

Return from subroutine. Return control to the main program and always skip the instruction following the JSR, JSRP, or JSRL.

PC ← SA ← SB Group 1 devices
or
PC ← SA ← SB ← SC Group 2 devices
or
N ← N-1, PC ← RAM_N Group 3 and Group 4 devices

Hex Code	7	6	5	4	3	2	1	0
49	0	1	0	0	1	0	0	1

Execution Time: 1 Instruction Cycle
Skip Conditions: Always skip on return
Restrictions: None
Availability: All COPS microcontrollers

HALT

Stop all internal operation of the device. Retain all internal status. Resume operation as result of external stimulus.

A, B, C, PC, G, L, Q, EN, RAM, T: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
38	0	0	1	1	1	0	0	0

Execution Time: 2 Instruction Cycles
Skip Conditions: None

Restrictions: Requires Hardware external restart

Availability: COP410C, COP411C, COP413C, COP424C, COP425C, COP426C, COP444C, COP445C, and COP404C

Note: This instruction places the eight microcontrollers mentioned above in their minimum power dissipation state.

IT

Stop all internal operation, except the timer, of the device. Resume operation at the instruction following IT when the timer overflows.

PC ← PC A, B, C, G, L, Q, EN, PC, RAM: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
39	0	0	1	1	1	0	0	1

Execution Time: 2 Instruction Cycles

Skip Conditions: None

Restrictions: None

Availability: COP424C, COP425C, COP426C, COP444C, COP445C, COP404C

3.2.3 Memory Reference Instructions

CAME

Copy the eight-bit contents of A and the memory location addressed by the B register to the eight-bit enable register (Note: the enable register is eight bits long in COP440 and COP2440 series only). This is the inverse of the CEMA instruction in function and with respect to the four bits of the enable register with which A and RAM(B) communicate.

$EN_{7:4} \leftarrow A_{3:0}$ A: Not affected

$EN_{3:0} \leftarrow RAM(B)_{3:0}$ C: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
1F	0	0	0	1	1	1	1	1

Execution Time: 2 Instruction Cycles

Skip Conditions: None

Restrictions: None

Availability: Group 3 devices

CAMQ

Copy the eight-bit contents of the accumulator and the memory location addressed by the B register to the eight-bit Q register. This is the inverse of the CQMA instruction in function and with respect to the four bits of Q with which A and RAM(B) communicate.

$Q_{7:4} \leftarrow A_{3:0}$ A: Not affected

$Q_{3:0} \leftarrow RAM(B)_{3:0}$ C: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
3C	0	0	1	1	1	1	0	0

Execution Time: 2 Instruction Cycles

Skip Conditions: None

Restrictions: None

Availability: All COPS microcontrollers

CAMT

Copy the eight-bit contents of the accumulator and the memory location addressed by the B register to the eight-bit timer register (T). This is the inverse of the CTMA instruction in function and with respect to the four bits of T with which A and M communicated.

$T_{7:4} \leftarrow A_{3:0}$ A: Not affected

$T_{3:0} \leftarrow RAM(B)_{3:0}$ B: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
3F	0	0	1	1	1	1	1	1

Execution Time: 2 Instruction Cycles

Skip Conditions: None

Restrictions: None

Availability: Group 3 devices, COP424C, COP425C, COP426C, COP444C, COP445C, COP404C

CEMA

Copy the contents of the eight-bit enable register (COP440 series only) to the memory location addressed by the B register and to the accumulator. This is the inverse of the CAME instruction in function and with respect to the four bits of the enable register with which A and RAM(B) communicate.

$A_{3:0} \leftarrow EN_{3:0}$ C: Not affected

$RAM(B)_{3:0} \leftarrow EN_{7:4}$

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
0F	0	0	0	0	1	1	1	1

Execution Time: 2 Instruction Cycles

Skip Conditions: None

Restrictions: None

Availability: Group 3 devices

CQMA

Copy the contents of the eight-bit Q register to the memory location addressed by the B register and to the accumulator. This is the inverse of the CAMQ instruction in function and with respect to the four bits of the Q register with which A and RAM(B) communicate.

$A_{3:0} \leftarrow Q_{3:0}$ C: Not affected

$RAM(B)_{3:0} \leftarrow Q_{7:4}$

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
2C	0	0	1	0	1	1	0	0

Execution Time: 2 Instruction Cycles

Skip Conditions: None

Restrictions: None

Availability: Not available on COP410L, COP411L, COP401L, COP410C, COP411C

CTMA

Copy the eight-bit contents of the time register to the memory location addressed by the B register and to the accumulator. This is the inverse of the CAMT instruction in function and with respect to the four bits of T with which A and RAM(B) communicate.

$A_{3:0} \leftarrow T_{3:0}$ C: Not affected
$RAM(B)_{3:0} \leftarrow T_{7:4}$

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
2F	0	0	1	0	1	1	1	1

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: None
Availability: Group 3 devices, COP424C, COP425C, COP426C, COP444C, COP445C, COP404C

LD n

Load the accumulator with the contents of the memory location addressed by the B register. Also, exclusive-OR the upper part of the B register (Br) with the n value.

A ← RAM(B) C: Not affected
Br ← Br ⊕ n

Hex Code	7	6	5	4	3	2	1	0
n5	0	0	n_1	n_0	0	1	0	1

n = 0, 1, 2, or 3

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: n = 0, 1, 2, 3 only
Availability: All COPS microcontrollers

LDD r,d

Load the accumulator with the contents of the memory addressed by the operand field, r,d. The B register is not used or altered.

A ← RAM(r,d) B: Not affected
C: Not affected

Hex Code	7	6	5	4	3	2	1	0
23	0	0	1	0	0	0	1	1
rd	0	r_2	r_1	r_0	d_3	d_2	d_1	d_0

r = 0:7; d = 0:15

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: r = 0, 1, 2, 3, 4, 5, 6, or 7 **only**
Availability: Not available in Group 1 devices

LID

Load the accumulator and the memory location addressed by the R register ith the eight-bit ROM word addressed by the upper bits of the program counter, A and RAM(B).

PC ← PC+2 C: Not affected
$RAM(B) \leftarrow ROM(PC_{10:8},A,RAM(B))_{7:4}$
$A \leftarrow ROM(PC_{10:8},A,RAM(B))_{3:0}$

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
19	0	0	0	1	1	0	0	1

Execution Time: 3 Instruction Cycles
Skip Conditions: None
Restrictions: LID in last word of block will access next block (#1 above)
Availability: Group 3 devices

LQID

Load the Q register with the eight-bit ROM word addressed by the upper bits of the program counter, the accumulator and the memory location addressed by the B register.

PC ← PC+1 A: Not affected
$Q_{7:4} \leftarrow ROM(PC_{10:8},A,RAM(B))_{7:4}$ C: Not affected
$Q_{3:30} \leftarrow ROM(PC_{10:8},A,RA,(B))_{3:0}$

Hex Code	7	6	5	4	3	2	1	0
RF	1	0	1	1	1	1	1	1

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: LQID in last word of a block accesses next block (#1 above). One level of subroutine stack is used by this instruction in Group 1 and Group 2 devices.
Availability: All COPS microcontrollers

RMB 0, RMB 1, RMB3

Reset the bit specified in the instruction in the memory location addressed by the B register.

RAM(B)n ← 0 C: Not affected
n = n,1,2,3 A: Not affected

		Hex Code	7	6	5	4	3	2	1	0
RMB	0	4C	0	1	0	0	1	1	0	0
RMB	1	45	0	1	0	0	0	1	0	1
RMB	2	42	0	1	0	0	0	0	1	0
RMB	3	43	0	1	0	0	0	0	1	1

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

SMB 0, SMB 1, SMB2, SMB 3

Set the bit specified in the instruction in the memory location addressed by the B register.

RAM(B)n ← 1 C: Not affected
n = 0,1,2,3 A: Not affected

		Hex Code	7	6	5	4	3	2	1	0
SMB	0	4D	0	1	0	0	1	1	0	1
SMB	1	47	0	1	0	0	0	1	1	1
SMB	2	46	0	1	0	0	0	1	1	0
SMB	3	4B	0	1	0	0	1	0	1	1

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

STII y

Store the immediate value y into the memory location addressed by the B register. Then increment the lower four

bits of the B register (Bd). The upper portion of the B register (Br) is not affected.

RAM(B) ← y A: Not affected
Bd ← Bd + 1 C: Not affected

Hex Code	7	6	5	4	3	2	1	0
7y	0	1	1	1	y_3	y_2	y_1	y_0

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

X n

Exchange the contents of the accumulator with the contents of the memory location addressed by the B register. Then replace Br with the exclusive OR of Br and n. Bd is not affected.

A ⟷ RAM(B) C: Not affected
Br ← Br ⊕ n

Hex Code	7	6	5	4	3	2	1	0
n6	0	0	n_1	n_0	0	1	1	0

n = 0, 1, 2, 3

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: n = 0,1,2, or 3 **only**
Availability: All COPS microcontrollers

XAD r,d

Exchange the contents of the accumulator with the contents of the memory location addressed by r,d. The B register is not affected.

A ⟷ RAM(r,d) B: Not affected
C: Not affected

	Hex Code	7	6	5	4	3	2	1	0
Group 1 devices	23	0	0	1	0	0	0	1	1
	BF	1	0	1	1	1	1	1	1
All Others	23	0	0	1	0	0	0	1	1
	r,d	1	r_2	r_1	r_0	d_3	d_2	d_1	d_0

r = 0:7; d = 0:15

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: On Group 1 devices; r=3, d = 15 **only**
All other COPS microcontrollers:
r = 0,1,2,3,4,5,6, or 7 **only**
Availability: All COPS microcontrollers

XDS n

Exchange the contents of the accumulator with the contents of the memory location addressed by the B register. Replace Br with the exclusive OR of Br and n. Decrement Bd by 1. Generate a skip if Bd decrements from 0 to 15.

A ⟷ RAM(B) C: Not affected
Br ← Br ⊕ n
Bd ← Bd − 1

Hex Code	7	6	5	4	3	2	1	0
n7	0	0	n_1	n_0	0	1	1	1

n = 0,1,2, OR 3

Execution Time: 1 Instruction Cycle
Skip Conditions: Generate a skip if Bd-1 = 15
Restrictions: n = 0,1,2, or 3 **only**
Availability: All COPS microcontrollers

XIS n

Exchange the contents of the accumulator with the contents of the memory location addressed by the B register. Replace Br with the exclusive OR of Br and n. Increment Bd by one. Generate a skip if Bd increments from 15 to 0.

A ⟷ RAM(B) C: Not affected
Br ← Br ⊕ n
Bd ← Bd + 1

Hex Code	7	6	5	4	3	2	1	0
n4	0	0	n_1	n_0	0	1	0	0

n = 0,1,2,3

Execution Time: 1 Instruction Cycle
Skip Conditions: Generate a skip if Bd+1 = 0
Restrictions: n = 0,1,2, or 3 **only**
Availability: All COPS microcontrollers

3.2.4 Register Reference Instructions

CAB

Copy the contents of the accumulator to the lower four bits of the B register.

Bd ← A A: Not affected
C: Not affected
Br: Not affected

Hex Code	7	6	5	4	3	2	1	0
50	0	1	0	1	0	0	0	0

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

CBA

Copy the lower four bits of the B register to the accumulator.

A ← Bd C: Not affected
B: Not affected

Hex Code	7	6	5	4	3	2	1	0
4E	0	1	0	0	1	1	1	0

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

LBI r,d

Load the B register immediate with the values r (to the upper portion of the B register). Skip all subsequent LBI instructions until an instruction that is not an LBI is encountered.

Br ← r A: Not affected
Bd ← d C: Not affected

Hex Code	7	6	5	4	3	2	1	0
r(d-1)	0	0	R_1	R_0	(d - 1)			

r = 0:3; d = 0, 9:15

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
rd	1	r2	r1	r0	d3	d2	d1	d0

r = 0:7; d = 0:15

33	0	0	1	1	0	0	1	1
7–	0	1	1	1	0	0	0	r_4
rd	r3	r2	r1	r0	d3	d2	d1	d0

r = 0:31; d = 0:15

Execution Time: 1 Instruction Cycle (One-byte form)
2 Instruction Cycles (Two-byte form)
3 Instruction Cycles (Three-byte form)
Skip Conditions: Skip until not an LBI
Restrictions: One-byte form:
r = 0,1,2,3 **only**
d = 0,9,10,11,12,13,14,15 **only**
Two-byte form: r = 0,1,2,3,4,5,6,7 **only**
Three-byte form: None
Availability: One-byte form: All COPS microcontrollers
Two-byte form: Not available on Group 1 devices
Three-byte form: Available on Group 4 devices **only**

LEI y

Load the enable regiter (lower four bits on COP440 and COP2440 series) with the immediate value y.

$EN_{3:0}$ ← y A: Not affected
C: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
6y	0	1	1	0	y_3	y_2	y_1	y_0

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

XABR

Exchange the contents of the accumulator with the contents of the upper part of the B register (Br). If Br is less than four bits wide, zeroes are placed in the corresponding bits of the accumulator.

Br ⟷ A, A_3 ← 0,A_2 ← 0 Devices with 64 or 32 RAM digits

Br ⟷ A, A_3 ← 0 COP404L, COP404C, COP444L, COP445L, COP444C, COP445C

Br ⟷ A Group 3 and Group 4 devices

Hex Code	7	6	5	4	3	2	1	0
12	0	0	0	1	0	0	1	0

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: None
Availability: Not available on Group 1 devices

XAN

Exchange the contents of the accumulator with the contents of the two-bit subroutine stack pointer. The lower two bits of A go into the stack pointer and the same two bits of A are loaded with the pointer value. The upper two bits of A are cleared.

$A_{1:0}$ ⟷ N C: Not affected
A_2 ← 0, A_3 ← 0

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
0B	0	0	0	0	1	0	1	1

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: None
Availability: Group 3 devices

3.2.5 Test Instructions

SKC

If the one-bit carry register (C) is equal to "1", skip the next program instruction.

A: Not affected
C: Not affected

Hex Code	7	6	5	4	3	2	1	0
20	0	0	1	0	0	0	0	0

Execution Time: 1 Instruction Cycle
Skip Conditions: Skip if C = 1
Restrictions: None
Availability: All COPS microcontrollers

SKE

If the contents of the accumulator are equal to the contents of the memory location addressed by the B register, skip the next program instruction.

A: Not affected
C: Not affected

Hex Code	7	6	5	4	3	2	1	0
21	0	0	1	0	0	0	0	1

Execution Time: 1 Instruction Cycle
Skip Conditions: Skip if A = RAM(B)
Restrictions: None
Availability: All COPS microcontrollers

SKGZ

If all four G lines are low ("0"), skip the next program instruction.

A: Not affected
C: Not affected
G: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
21	0	0	1	0	0	0	0	1

Execution Time: 2 Instruction Cycles
Skip Conditions: Skip if $G_{3:0} = 0$
Restrictions: None
Availability: All COPS microcontrollers

SKGBZ n, n = 0,1,2,3

If G(n) is zero, skip the next program instruction.

A,C,G: Not affected

		Hex Code	7	6	5	4	3	2	1	0
SKGBZ	0	33	0	0	1	1	0	0	1	1
		01	0	0	0	0	0	0	0	1
SKGBZ	1	33	0	0	1	1	0	0	1	1
		11	0	0	0	1	0	0	0	1
SKGBZ	2	33	0	0	1	1	0	0	1	1
		03	0	0	0	0	0	0	1	1
SKGBZ	2	33	0	0	1	1	0	0	1	1
		13	0	0	0	1	0	0	1	1

Execution Time: 2 Instruction Cycles
Skip Conditions: Skip if specified G bit is zero
Restrictions: None
Availability: All COPS microcontrollers

SKMBZ n n = 0,1,2,3

If the specified bit in the memory location addressed by the B register is "0", skip the next program instruction.

A,C,RAM(B): Not affected

		Hex Code	7	6	5	4	3	2	1	0
SKMBZ	0	01	0	0	0	0	0	0	0	1
SKMBZ	1	11	0	0	0	1	0	0	0	1
SKMBZ	2	03	0	0	0	0	0	0	1	1
SKMBZ	3	13	0	0	0	1	0	0	1	1

Execution Time: 1 Instruction Cycle
Skip Conditions: Skip if RAM(B)n = 0
Restrictions: None
Availability: All COPS microcontrollers

SKSZ

If the four-bit serial input/output register is "0", skip the next program instruction.

A,C: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
1C	0	0	0	1	1	1	0	0

Execution Time: 2 Instruction Cycles
Skip Conditions: Skip if SIO = 0
Restrictions: None
Availability: Group 3 devices

SKT

If T counter carry (overflow) has occurred since the last test (last SKT), skip the next program instruction. Reset the SKT latch. (Timer carry/overflow sets SKT latch. SKT instruction tests and resets this latch).

SKTL ← 0 A,C,T: Not affected

Hex Code	7	6	5	4	3	2	1	0
41	0	1	0	0	0	0	0	1

Execution Time: 1 Instruction Cycle
Skip Conditions: Skip if SKTL = 1
Restrictions: None
Availability: Not available on Group 1 devices

3.2.6 Input/Output Instructions

CAMR

Copy the contents of the accumulator and the memory location addressed by the B register to the eight-bit R port. This is the inverse of the INR instruction in function and with respect to the four bits of R which are accessed by A and RAM(B).

$R_{7:4}$ ← A A: Not affected
$R_{3:0}$ ← RAM(B) C: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
3D	0	0	1	1	1	1	0	1

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: None
Availability: Group 3 devices

Note: On COP441 and COP442,R as I/O port is not present, but R as eight-bit internal register is available.

ING

Copy the status of the G I/O port into the accumulator.

A ← G C: Not affected
G: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
2A	0	0	1	0	1	0	1	0

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

INH

Copy the status of the H I/O port to the accumulator.

A ← H C: Not affected
H: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
2B	0	0	1	0	1	0	1	1

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: None
Availability: Group 3 devices

Note: On COP441 and COP442, H as I/O port is not present, but H as four-bit internal register is available.

ININ

Copy the status of the four IN lines to the accumulator.

A ← IN C: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
28	0	0	1	0	1	0	0	0

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: None
Availability: COP420, COP420L, COP444L, COP440, COP441, COP424C, COP444C

INIL

Copy the status of the IL latches and CKO input and zero cross input (COP440, 441) to the accumulator. Reset the IL latches.

1a) A 3:0 ← IL 3, CK0, IN 1Z, IL 0 COP440, COP441

or

1b) A 3:0 ← IL 3, CK0, "0", IL 0 COP420, COP420L, COP444L, COP424C, COP444C

or

1c) A 3:0 ← "0", CKO, "0", "0" COP442, COP2442, COP421, COP422, COP421L, COP422L, COP445C, COP425C, COP426C

2) IL_3 ← 0, IL_0 ← 0

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
29	0	0	1	0	1	0	0	1

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: If CKO is not selected as general input, "1" is loaded into A_2. IL latches are reset at power on in Group 3 and Group 4 devices only. On other devices, the latches are undefined until first INIL
Availability: Not available on Group 1 devices

INL

Copy the status of the eight-bit L port to the memory location addressed by the B register and the accumulator.

RAM(B) ← $L_{7:4}$ C: Not affected
A ← $L_{3:0}$

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
2E	0	0	1	0	1	1	1	0

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

INR

Copy the status of the eight-bit R port to the memory location addressed by the B register and the accumulator. This is the inverse of the CAMR instruction and with respect to the four bits of R which are accessed by A and RAM(B).

RAM(B) ← $R_{7:4}$ C: Not affected
A ← $R_{3:0}$

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
2D	0	0	1	0	1	1	0	1

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: None
Availability: Group 3 devices

Note: On COP441 and COP442, R as an I/O port is not present but R as eight-bit internal register is available.

OBD

Copy the contents of the lower four bits of the B register (Bd) to the D output port.

D ← Bd A: Not affected
B: Not affected
C: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
3E	0	0	1	1	1	1	1	0

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

OGI y

Output the immediate value y to the four-bit G port.

G ← y A: Not affected
C: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
5y	0	1	0	1	y_3	y_2	y_1	y_0

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: None
Availability: Not available on Group 1 devices

OMG

Copy the contents of the memory location addressed by the B register to the four-bit G port.

G ← RAM(B) A: Not affected
C: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
3A	0	0	1	1	1	0	1	0

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

OMH

Copy the contents of the memory location addressed by the B register to the four bit H port.

H ← RAM(B) A: Not affected
C: Not affected

Hex Code	7	6	5	4	3	2	1	0
33	0	0	1	1	0	0	1	1
3B	0	0	1	1	1	0	1	1

Execution Time: 2 Instruction Cycles
Skip Conditions: None
Restrictions: None
Availability: Group 3 devices

XAS

Exchange the contents of the accumulator with the contents of the SIO register. Copy the contents of the one-bit C register to the SK latch. This is the basic MICROWIRE interface instruction and is the primary control over the serial port.

A ⟷ SIO C: Not affected
SKL ← C

Hex Code	7	6	5	4	3	2	1	0
4F	0	1	0	0	1	1	1	1

Execution Time: 1 Instruction Cycle
Skip Conditions: None
Restrictions: None
Availability: All COPS microcontrollers

3.3 NOTES ON ADDRESSING MODES

COPS microcontrollers do not have addressing modes in the sense of most popular microprocessors. To be sure, every instruction can be said to have some form of addressing mode associated with it. For example, a jump can be direct (JMP), indirect (JID), or "modified relative" (JP); and adds can be immediate (AISC) or inherent/implied (ASC,ADD). A classification of this kind can be made, but it is awkward and forced; it is an attempt to impose the structure of one type of microcomputer on another type of microcomputer. Because of the difference in kind between these microcomputers, a comparison on the basis of number of addressing modes between COPS and some other microcomputer is not valid. One may be able to find six or seven kinds of addressing modes in the COPS instruction set, but such an effort is more an exercise of the imagination than a meaningful evaluation of the instruction set. Comparisons should be made on what the instruction set really requires, in terms of the relevant parameters, memory usage, and speed, to perform a given function.

4.0 Programming COPS Microcontrollers

4.1 INTRODUCTION

This section deals with all aspects of programming COPS devices. The concepts, structures, rules, suggestions, and tricks for COPS programming are discussed. The detailed effects of various instructions are also discussed.

4.2 BOUNDARY CONDITIONS

Although the program counter in COPS microcontrollers will increment linearly throughout the address space, three types of boundaries exist in the program space that the user should remember.

- Page boundaries
- Block boundaries
- Chapter boundaries

Even though these boundaries exist, their impact on the actual programming is minimal. This is true because these boundaries are important in only a few instructions and even there the primary effect, in most cases, is to allow the user to use a more code efficient instruction.

4.2.1 Page Boundaries

A page is composed of 64 contiguous ROM words. Page 0 is the group of ROM words located at hex addresses 000 through 03F; Page 1 is the group of ROM words located at hex addresses 040 through 07F; etc. (See Table III.) The page boundary saves code by allowing the use of the single-byte jump (JP) and the single-byte subroutine call (JSRP).

Furthermore, Pages 2 and 3 are the special subroutine pages. Page 2 is the destination page for subroutines called by JSRP instruction.

THE JP INSTRUCTION

The JP instruction is the single-byte jump. It loads the lower six bits of the program counter only; therefore, it causes a jump within a page only. There is an exception to this, however. A JP instruction located at the last word of a page (hex addresses 03F, 07F, 0BF, 0FF, etc.) will cause a jump into the next page. In all COPS microcontrollers, the program counter is incremented before the execution of the instruction. Thus, the program counter will increment from hex address 13F, the last word of a page, to hex address 140, the first word of the next page; then the JP will load the lower six bits of the PC. The effect is to cause a jump from one page to the next page with the single-byte JP.

The JP instruction cannot be used to jump to the last word of a page. The reason for this is evident from an examination of the instruction OP codes. The two most significant bits of the JP instruction are 11. The lower bits of the address of the last word of a page are all ones. Thus, the OP code of a JP to the last word of a page would be hex FF. This, however, is the opcode for the JID instruction. Therefore, JP cannot be used to jump to the last word of a page because the opcode that would otherwise implement that jump has been used to create the JID instruction.

The JP instruction has an expanded range within the subroutine pages—Pages 2 and 3. In these two pages only, the JP instruction loads the lower 7 bits of the program counter. Thus, a JP within Pages 2 and 3 may jump anywhere, except the last word of Page 2 or last word of Page 3, within Pages 2 and 3.

The JSRP Instruction

The JSRP instruction is the single-byte subroutine call. Page 2 is the destination page for the subroutine jump. The instruction indicates the address within Page 2 where the subroutine begins. The two restrictions on the use of JSRP are as follows:

1. JSRP to the last word of Page 2 is not allowed
2. JSRP may not be used within Page 2 or 3

The reason for both restrictions is evident from the opcodes. The most significant two bits of JSRP are 10. The lower six bits are the address within Page 2. Thus, JSRP to the last word of Page 2 would have the opcode hex BF. This opcode, however, has been used to implement the LQID

instruction. Thus, a JSRP to hex address 0BF, the last word of Page 2, is not allowed. JSRP may not be used within Pages 2 and 3 simply because the opcodes have been used to expand the range of the JP instruction as explained in Section 4.2. The sacrifice of the JSRP to expand JP in the subroutine pages helps to create more entry points in Page 2 which tends to increase program efficiency.

4.2.2 Block Boundaries

A block is composed of four contiguous pages or 256 contiguous ROM words. Block 0 consists of Pages 0 through 3; Block 1 consists of Pages 4 through 7; etc. (See Table III.) The block boundary is significant only with respect to the indirect instructions: JID, LQID, and LID. These instructions operate within a block and do not normally cross block boundaries.

LQID and LID

These are the table look-up instructions. LQID looks up data identified by A and RAM(B) and puts the value in Q. LID does the same but returns the value to A and RAM(B). Hence, the look up is based on an eight-bit value. The lower eight bits of the program counter are temporarily replaced by the contents of A and RAM(B). The remaining bits of the PC are not affected by the instruction. Thus, these instructions work within a block.

Just as with the JP instruction, a special situation exists if the LQID is at the last word of a block of LID is at the last two words of a block. In this situation, the look up is performed in the next block. The reason is, as explained before, the program counter is incremented before the instruction is executed. Thus, the program counter will be in the next block before the look-up operation is performed.

The JID Instruction

The JID instruction looks up an address on the basis of A and RAM(B), then loads the lower eight bits of the program counter with that address. Again, since eight-bit values are being used, block boundaries are respected.

Since the program counter is incremented prior to instruction execution, a JID at the last word of a block will look up its address in the next block and execute the jump in that block. An additional related special case exists with the JID instruction. If the look-up address for the JID is at the last word of a block (i.e., A = 15_{10} and RAM(B) = 15_{10}), then the jump will be in the next block. A final combination case exists: If JID is at the last word of a block and A = 15_{10} and B = 15_{10}, then the jump will be in the second block from the present block (see Table VI).

4.2.3 Chapter Boundaries

The Chapter is the largest memory division in COPS microcontrollers. A Chapter is composed of eight contiguous blocks (32 contiguous pages, 2048 contiguous ROM words). Obviously, the Chapter boundary has no relevance, in fact does not exist, if the microcontroller has fewer than 2048 words of program memory. Only the two-byte JMP and two-byte JSR are affected by the Chapter boundary. These instructions will jump anywhere within a Chapter or call a subroutine anywhere within a Chapter and will not normally cross a Chapter boundary. The exception is basically the same as seen before: a JMP at the last two words of a Chapter will jump to the next Chapter; a JSR at the last two words of a Chapter will call a subroutine in the next Chapter. The reason is the same: the program counter is incremented before the instruction is executed.

TABLE VI. Effects of Block Boundaries on JID Destinations

JID Location	A	RAM(B)	Destination
Block N, anywhere except last word	≠15	≠15	Block N
Block N, anywhere except last word	15	≠15	Block N
Block N, anywhere except last word	≠15	15	Block N
Block N, anywhere except last word	15	15	Block N + 1
Block N, last word	≠15	≠15	Block N + 1
Block N, last word	15	≠15	Block N + 1
Block N, last word	≠15	15	Block N + 1
Block N, last word	15	15	Block N + 2

4.3 SKIP CONDITIONS

In COPS microcontrollers, program address information is contained only in the jump and subroutine call instructions. Thus, decision instructions, or tests, do not contain a branch address. There is no single instruction equivalent of "If condition X is true (false) branch to address A." Instead, in COPS devices, if the test condition is met a skip is generated. This skip prohibits the execution of the following instruction, i.e., "skipping" that instruction. The number of program bytes in the instruction has no bearing on the skip operation. Thus, following a test instruction with jumps or subroutine calls produces the desired branching. However, the skip feature allows much greater flexibility than merely branching. In many cases, the skip feature eliminates the need for branching since almost any register or variable parameters in COPS microcontrollers can be modified, in line, on the basis of a skip (see Section 4.7).

4.3.1 Effect of Skips on Timing Loops

Software timing loops are commonly part of a microcontroller program. In such a case, it is usually necessary that various paths through the loop take the same amount of time. The skip feature actually helps to achieve this goal rather than, as might be expected, conflicting with it. The reason is in the operation of the skip. If an instruction is to be skipped, the internal logic forces a NOP equal in length to the number of program bytes in the skipped instruction in place of that instruction. Then the NOP is executed. Thus, whether or not an instruction is skipped has no effect on the time to execute a given sequence of instructions. Note: this "hardware NOP" is temporary; it exists for the duration of the skipped instructions only and in no way alters the ROM contents. It therefore becomes a simple matter to compute execution time through a given sequence. Merely count the number of bytes, not instructions, in the path without regard to tests or skips and multiply by the instruction cycle time.

The Indirect Instructions—An Exception

The indirect instructions JID, LQID, and LID constitute an exception to this general rule. These are the only COPS instructions that require more instruction cycle times than the number of bytes in the instruction to execute. They require one more instruction cycle time than the number of bytes to execute: JID and LQID are one-byte instructions and require two instruction cycles to execute; LID is a two-byte instruction that requires three instruction cycles to exe-

cute. The result is that these instructions use one more instruction cycle when executed than when skipped because the hardware forced NOP is related to the number of bytes in the instruction rather than the execution time of the instruction. This distinction is significant only for these three instructions.

4.3.2 Instructions That Generate a Skip

As would be expected, all test instructions can generate a skip. If the test condition is met, a skip is generated. However, certain other instructions can also generate skips. The following arithmetic instructions generate a skip if the result of a four-bit binary addition is greater than 15_{10}: ASC, CASC, and AISC. The advantage here is that the common test after such instructions (testing carry or overflow) is built directly into the instruction thereby eliminating the need for a separate test instruction.

The LBI (load B register immediate) can also generate a skip. This instruction forces a skip until an instruction is reached that is not an LBI. This permits multiple entry points to a common routine without affecting the code. The code savings of this feature are more subtle, but this allows the user a degree of flexibility not found in other devices. Section 4.7 will explain this feature in more detail.

The XIS and XDS instructions can also generate skips. These generate a skip when one increments or decrements "off the end" of a register (Bd incrementing from 15 to 0 or decrementing from 0 to 15). This becomes very useful in loop operations as the need for testing for completion of the loop is often eliminated—another test is eliminated. Section 4.7 will illustrate the use of these instructions.

The final instruction that generates a skip is RETSK. When executed, this instruction always forces a skip of the instruction located at the return address. This instruction becomes very valuable in implementing complex tests in a subroutine, or in reversing the direction of a frequently used test by means of a special subroutine. It is, of course, useful whenever the user wishes to force a skip of a subroutine return address.

4.4 CARRY

The ALU in COPS microcontrollers is a four-bit parallel binary adder. The user does not have access to the bit-to-bit carry within the ALU, but does have varying degrees of access to the carry as a result of a four bit operation. Within this category the user should be aware of several distinctions: the carry register, the carry out of the ALU, and simple arithmetic overflow. These are not always the same thing and the difference can be important. The carry register, C, may be set or reset directly by the program. Those instructions that do an "add with carry", ASC and CASC, use the C register in the addition. These same two instructions are the **only** instructions that load the carry out of the ALU, the carry as a result of the four-bit addition, into the C register. The SKC instruction test the status of the C register, not the carry from the ALU.

The carry from the ALU is the controlling factor in those arithmetic instructions that can generate a skip: ASC, CASC, AISC. If the carry from the ALU is a one as a result of any of these instructions, a skip is generated. The C register is not used for this form of skip generation. In fact, the AISC instruction neither uses nor affects the C register.

The ADD and ADT instructions cause an add to be performed. This add may well cause an arithmetic overflow. This overflow, however, is not quite the same as the carry from the ALU since no skip condition occurs. Furthermore, the C register is neither used nor affected.

This can be viewed as a hierarchy of overflows:

1. Simple arithmetic overflow; no skip; C neither used nor affected. ADD, ADT
2. Carry from the ALU (= arithmetic overflow which generates a skip); C neither used nor affected. AISC
3. Carry from the ALU that loads C (= arithmetic overflow which generates skip, C loaded with status of carry from ALU); C both used and affected. ASC, CASC

4.5 INPUT/OUTPUT

All input/output operations are handled by unique instructions. The instructions may be executed at any point in the program.

4.5.1 Unidirectional Ports

Two unidirectional ports are found in COPS microcontrollers: the IN input port and the D output port. The IN port is read by the ININ instruction. The IL latches, associated with the IN port, are read by the INIL instruction. Pin CKO may be configured as an input, via a mask option, on some devices. The INIL instruction also reads the state of the CKO input in those devices that have that option. See the descriptions of the ININ and INIL instructions for further details.

The D output is loaded from the lower four bits of the B register (Bd) by means of the OBD instruction. There is no path from the accumulator or RAM to the D port.

4.5.2 Bidirectional Ports

Non TRI-STATE Ports

There are two bidirectional, non TRI-STATE ports available: The G port, available, at least partially, on all COPS microcontrollers; and the H port, available on the COP440. The output function is simple; merely write the data to the port with the appropriate instruction: OGI, OMG, or OMH. Data is read via the ING or INH instruction. In addition, the G lines may be directly tested individually or as a four bit group. When using any of the G or H lines as inputs the user must write a "1" to the lines used as inputs. This is a requirement imposed by hardware rather than software considerations. The external circuitry will pull the line to logic "0".

On 20-pin COPS devices, only two of the four G lines are brought out. The other two lines, however, are available for internal use as flags or storage. The same is true of the H port on the COP441/COP442.

Any G or H line, or any combination, may be used as inputs while the others are used as outputs. There is no conflict and the user has complete flexibility.

TRI-STATE Ports

Two eight-bit bidirectional TRI-STATE ports are available: The Q register-L drivers available on all COPS microcontrollers and the R port available on the COP440. The L port is written by loading Q with CAMQ or LQID and enabling L, via LEI or CAME. The application will determine if L should be enabled before or after loading Q or enabled all the time. The decision is not significant in terms of software. Remember, the L outputs are dirvers **only**. They are not latched. When enabled, L outputs the contents of Q. L **must** be enabled in order to output data. The R port is a latched output port. The user writes to the R register by means of the CAMR instruction. The R drivers are enabled by means of the CAME instruction. In terms of software alone, it is not significant when the R drivers are enabled, but the drivers must be enabled to output the contents of the R register.

There are two ways to use these lines as inputs. The first method requires that the drivers be disabled. In this case, the lines are truly floating and in an undefined state. The external circuitry must provide good logic levels, both high and low, to the input pins. The inputs are then read by the INL or INR instructions. The second method is very similar to the technique used for G and H. The drivers are enabled. A "1" must be written to the Q or R register in the positions of the input lines. The external circuitry will then be required only to pull the line down to a logic "0". The line will pull itself up to a logic "1". The INL and INR instructions are used as before to read the lines.

Any L or R line, or any combination, may be used as inputs while the others are used as outputs. However, the L drivers are enabled or disabled as a group. The same is true of the R drivers. The L drivers are enabled or disabled by means of the LEI or CAME instructions. The R drivers are controlled by means of the CAME instruction only. On most devices, the Q register can be read without affecting L. The R register can be read only through the R lines. The data on the L lines does not affect the contents of the Q register except on devices with the MICROBUS option selected. The data on the R lines does not affect the contents of the R register.

The R lines are available only on the COP440. The R register, however, is available and can be used in the COP441/COP442.

4.5.3 The Serial I/O Port—MICROWIRE

As explained in Section 2.4.5, the serial I/O port may be configured as a serial shift register or a four-bit binary down counter. In the shift register mode, the serial port is the MICROWIRE interface (see Section 2.4.5). The operating mode of the serial port is controlled by the Enable register (see Section 2.5 and Table IV).

In the binary counter mode, SO and SK are logic controlled outputs. The state of SO is directly controlled by the LEI instruction. SK outputs the status of SKL, the SK latch. In the shift register mode, SO is either "0" or serial out, and SK is either "0" or a clock output as indicated in Table IV. Regardless of mode, SKL is loaded with the status of the C register whenever an XAS instruction is executed. Thus, SK is controlled by setting or resetting C and then executing an XAS. The XAS instruction, however, is also the means of reading the SIO register. Therefore, every time the user reads SIO, C is copied to SKL. Therefore, the user should insure the status of C before executing an XAS instruction if the status of SK is important. Also note that if SIO is in counter mode and SKL is "1" (SK = 1), and SIO changed to shift register mode, SK will become a clock immediately. The converse is also true: If SIO is shift register and SKL = 1, and SIO is changed to a counter, SK will go to a high state immediately.

Regardless of mode, SI can be used as a general purpose input. In the shift register mode, data will shift in at the SI pin. The user can read the status of SI with the XAS instruction. In the counter mode, SIO will, in effect, capture a low-going pulse. The user can preload the counter by setting the accumulator to some value, typically 0 or 15, and loading that value into SIO with an XAS instruction. The user would then periodically read SIO to see if the value had been decremented. If it had, the pulse had occurred.

With the SIO register in the shift register mode, continuous data streams can be sent or received. In this mode, data is normally in multiples of four bits. To preserve proper timing, an XAS must appear every fourth instruction cycle. As will be seen, this is simple to implement. The reason for this requirement should be obvious. SIO is a four-bit shift register which shifts at the instruction cycle rate. Thus data must be read, or new data loaded, every fourth instruction cycle.

4.6 INTERRUPT

The interrupt input on COPS microcontrollers is IN_1. In the COP440 series and COP2440 series, the CKO input may also be an interrupt input. Thus, except for the COP442, interrupt is not available on any device that does not have the IN_1 input.

4.6.1 Conditions for Interrupt Recognition

An interrupt will be recognized or acknowledged if and only if the following conditions are met:

1. Interrupt has been enabled by setting bit EN_1 of the enable register.
2. A low-going pulse ("1" to "0") at least two instruction cycles wide occurs at the IN_1 (or CKO in COP440 series) input. The high to low transition must occur while EN_1 is set.
3. A currently executing instruction is completed.
4. All successive transfers of control instructions and successive LBI instructions are completed (e.g., if the main program is executing a jump or subroutine call which transfers control to another jump or subroutine call, the interrupt will not be acknowledged until the second jump or subroutine call has been executed).

4.6.2 Effects of Interrupt Acknowledge

When an interrupt has been acknowledged as explained in Section 4.6.1, the following occurs:

1. The next sequential program counter address (PC + 1) is pushed onto the program stack.
2. On COP440 series device, an interrupt status bit is stored with the address in the subroutine stack.
3. On all other COPS microcontrollers, the interrupt status bit, which remembers the status of the skip logic, is saved separately. This bit is not carried with the address in these devices.
4. The program counter is set to address 0FF hex. On all devices except the COP440 series, the next executable address is hex 100. In the COP440 series, hex 100 is the next executable address if EN_4 is reset. If EN_4 is set in these devices, the program counter branches from hex address OFF to hex address 300.
5. EN_1 is reset thereby disabling further interrupts.

4.6.3 Interrupt Handling

Due to hardware considerations, the instruction at hex address OFF **must** be a NOP.

The interrupt status bit remembers if a skip was generated as a result of the completed instruction. In the COP420/COP424C/420L/444L devices, this bit is stored separate from the return address. If set, this bit forces a skip on the first "stack pop" following the interrupt. This means that the use of subroutines or nested interrupts is limited in these devices. An unexpected skip may occur and the original skip status is lost. The user may, of course, defeat this skip by means of an artificial subroutine call, e.g., a JSRP to a RET instruction, followed by a NOP. This will clear the status bit, and subroutines, etc. may be used without restriction. Remember, however, that this procedure destroys the original skip status. No such situation exists in the COP440 series devices. The status bit is saved with the address.

Subroutines may be freely used in the interrupt service routines and nested interrupts are permitted.

Subject to the restraints mentioned above, interrupts may be re-enabled at any time by means of the LEI or CAME instructions. Typically, this re-enabling would occur immediately before the return instruction at the end of the interrupt service routines.

4.6.4 Interrupt Disable

Interrupts are disabled by resetting EN_1 by any valid instruction (LEI or CAME) and by interrupt acknowledge. While EN_1 is low, no interrupt processing of any kind goes on. Thus, a high to low transition at IN_1 which is otherwise valid is not recognized when EN_1 is reset. Furthermore, when EN_1 is set, there is no memory of the event that occurred while EN_1 was reset. The software interrupt disable will prohibit recognition of all interrupt signals which occur subsequent to the disable. Obviously, the interrupt disable instruction cannot disable interrupts which occur before the instruction is executed. More significantly, the interrupt disable instruction also does not disable interrupts which occur during the execution of the instruction. Thus, a valid interrupt signal may occur, and interrupt acknowledge is pending completion of the current instruction. That current instruction may well be an interrupt disable; nonetheless, the interrupt will be acknowledged and the interrupt service routine entered.

Note that in branching to the interrupt routine, the microcontroller saves only the program counter and the skip status. If it is necessary to save other items, the user must do so himself in software. Similarly, the user must restore those values at the end of the interrupt service routine.

4.6.5 Interrupt in the COP440

The COP440 series devices are the only COPS microcontrollers with more than one possible interrupt source. The choice of interrupt is governed by bits EN_4 and EN_5 of the enable register as indicated in Table V. The four possible interrupt sources are as follows:

1. IN_1 negative edge—This is the standard COPS interrupt (EN_5, EN_4 = 00).
2. CKO input—If the CKO input mask option is selected, that input can be selected as an interrupt input. Operation is the same as the IN_1 interrupt (EN_5, EN_4 = 0). If CKO is not selected as an input, selection of CKO as interrupt source has no effect. No interrupt will occur.
3. Zero Crossing on IN_1—IN_1 may be mask programmed to be a zero crossing detect input. Interrupt can be selected to occur at each zero crossing. If the zero cross detect option is not selected, this interrupt source selection will result in an interrupt at every transition of IN_1 (EN_5, EN_4 = 10).
4. T counter overflows—This is an interrupt which can be selected. Interrupt will occur whenever the T counter overflows. All the conditions required for interrupt to be acknowledged, with the obvious exception of input pulse width, are still valid and must be met (EN_5, EN_4 = 11).

The interrupt source should not be changed while the interrupt is enabled (EN_1 = 1). A false interrupt may occur if the interrupt source is changed while EN_1 is a 1. To avoid this problem, the interrupt must be disabled prior to, or at the same time as, the change of the interrupt source. Do not enable the interrupt at the same time as changing the interrupt source. A proper sequence for altering the interrupt source, then, is as follows:

1. Disable interrupt.
2. Change interrupt source (Steps 1 and 2 may be combined).
3. Enable interrupt.

4.7 PROGRAM EFFICIENCY

Three factors are normally involved in determining program efficiency:

1. Program memory (ROM) efficiency, using the least amount of ROM.
2. Data memory (RAM) efficiency, using the least amount of RAM.
3. Execution time efficiency, executing the function in the shortest amount of time.

These three factors, unfortunately, conflict with one another. The most memory efficient implementation of a function is not usually the most execution time efficient implementation. The most RAM efficient implementation is frequently not the most ROM efficient implementation.

Like all single-chip microcontrollers, COPS microcontrollers are memory limited. A premium is therefore placed on general memory efficiency—getting the maximum function in the smallest memory. The reason is simple economics: devices with greater memory capacity are generally more expensive than devices with lesser capacity. Despite the premium on memory efficiency, the application can easily require compromises—sacrifice a little ROM or RAM or both in order to achieve faster execution speed.

Since these conflicting requirements exist, several versions of the standard programs in Chapter 5 are provided. These should help the user to understand the conflict and to make intelligent, informed decisions on any compromises.

4.8 RULES AND TECHNIQUES

4.8.1 Absolute Requirements

There are very few absolute requirements for COPS programming. The restrictions on the instructions are described in Chapter 3. The remaining absolute rules are as follows:

1. The instruction at address 000 must be a CLRA.
2. If interrupts are used, the instruction at hex address 0FF must be a NOP.
3. At least the first instruction of subroutines called with the single byte JSRP must be in Page 2. Note there is no requirement that any other instructions of such subroutines be located in Page 2.

4.8.2. General Guidelines

This section will provide general guidelines to help the programmer write an efficient COPS program. Examples are provided here and in Section 5. Most of these guidelines will reduce memory usage at the expense of execution speed. The programmer may have to make the compromises described in Section 4.7.

Maximize the Use of Subroutines

If a single operation is frequently performed, make that operation a subroutine. If possible, make it a subroutine that can be called with the single-byte JSRP. Try to combine similar operations into a common subroutine, even if it means that an unnecessary operation is performed in some cases. This is "wrong" only if this unnecessary operation interferes in some significant way with achieving the end result. The programmer may use pieces of existing subroutines as new subroutines: multiple entry points are a good thing if code is saved. Consider the following short routine:

```
ENTRY1:  LBI 0,15
ENTRY2:  LD
ENTRY3:  CAB
ENTRY4:  OBD
ENTRY5:  RET
```

It is entirely conceivable that every instruction in this routine is a subroutine entry point. We shall assume that this routine is in Page 2 for maximum savings. A JSRP to ENTRY1 will output the value in RAM (0,15) to D. A JSRP to ENTRY2 will output the RAM digit addressed by B to the D port. A JSRP to ENTRY3 will output the accumulator to the D port. A JSRP to ENTRY4 simply does an OBD. A JSRP to ENTRY5 is, effectively, a NOP but finds usefulness in creating software delays. This is an example of maximizing subroutine usage and sharing commonality or finding commonality where it is not obvious. Page 2 should be filled with subroutine entry points. This will increase the memory efficiency of the program.

Any multibyte instruction can be converted into a single-byte instruction by means of a subroutine. Entry point ENTRY4 in the preceding example illustrates this. If a given multibyte instruction is frequently used in a program, it will probably be beneficial to make a subroutine out of it. Remember, this includes any multibyte instruction. It is common that various branches of a program will jump back to some central location in the program. These jumps can be implemented with a JSRP; the subroutine will consist totally of JMP CENTER, a jump to the central location. This is completely acceptable and will save code. A subroutine does not have to have a return instruction associated with it.

Use the skip feature of successive LBI instructions. This is a very powerful feature that permits code sharing and promotes commonality. It easily lends itself to multiple entry point routines. Consider the following digit right shift routine:

```
RSH0:  LBI   0,15
RSH1:  LBI   1,15
RSH2:  LBI   2,15
RSH3:  LBI   3,15
       CLRA
LOOP:  XDS
       JP    LOOP
       RET
```

Depending on the entry point, this routine will right shift register 0, 1, 2, or 3 one digit. The successive LBI feature finds use in this kind of routine, so the same routine can be used regardless of data location in tests and in "non-obvious" ways. Consider the following:

```
G10:  LBI   0,10
G9:   LBI   0,9
G1:   LBI   0,1
G0::  LBI   0,0
      CBA
      X
      OMG
      X          ;Restore original RAM value
      RET
```

Here, the LBI instruction is being used to establish the G output value. The LBI instruction can be used in many similar ways. The interesting thing about this usage is that the LBI is, in itself, being used to create a value and not to point to a given RAM digit, even though the B register is modified by the instruction.

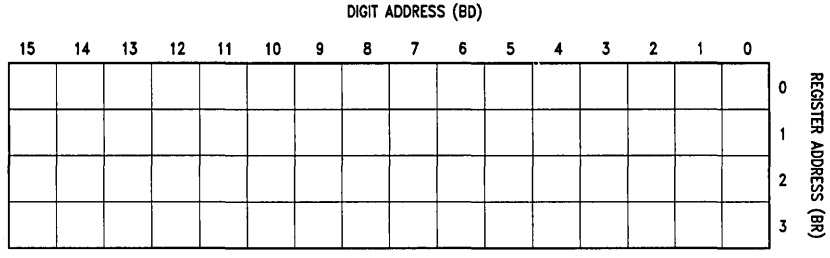
Careful RAM allocation is essential. Careful placement of data in RAM can have significant impact on the amount of program memory required. The use of a RAM map, a visualization of the data placement in RAM, is an invaluable aid. It is nearly impossible to write an efficient program without the use of a RAM map. *Figure 4-1* is a sample RAM map for a COP420. The basic guidelines for data placement in RAM are as follows:

1. Flags should be placed in memory locations addressable by a single-byte LBI.
2. A commonality of bit position within a digit for flags is desirable. This permits the creation of flag testing subroutines like the following:

```
FLAG1:  LBI    3,15
FLAG2:  LBI    3,14
        SKMBZ  1
        RET
        RETSK
```

3. Data should be placed at the "ends" of register to take advantage of the skip features of the XIS and XDS instructions. It takes far less code to exit by "falling of the end" of a register than to test Bd, or some other loop counter, for completion.

The LD, X, XIS, and XDS are associated with the exclusive OR feature whereby Br can be modified. If RAM data and flags are intelligently placed, data manipulation and B register modification can be accomplished in a single instruction thereby saving code. Obviously, the effective use of these instructions goes hand in hand with an effective RAM layout. The basic integer BCD addition below illustrates this feature. The routine is a four-digit BCD addition, adding register 0 to register 3; result to register 0.



TL/DD/8800–47

FIGURE 4-1. COP420 RAM Map

```
BCDADD: LBI   3,12
LOOP:   LD    3     ;fetch data and point to
                    R0
        AISC  6     ;decimal adjust to force
                    carry if A=9
        ASC         ;add
        ADT          ;decimal correct
:       XIS   3     ;place digit in R0,
                    increment Bd, point to R3
        JP    LOOP  ;XIS skip indicates finish
        RET
```

In the above routine, the B register is being continuously modified but there is no LBI instruction other than the one required at the start of the routine.

The table look-up instructions, LQID and LID, can save both code and execution time. Tables can be used in many ways: code conversions, arithmetic, data processing, key decoding, etc. If some set of values is to be derived from another set of values, a table will frequently be more efficient than a computation. The look up will also be invariably faster than a computation. Tables greatly facilitate the handling of inputs from non-linear sources e.g., temperature sensors; they make creation of display a trivial task. The use of a table is not a panacea but is frequently a possible solution worth considering.

The indirect jump instruction, JID, should be used with some care. Because of its "two-tier" organization, this instruction does not always save code. JID permits a jump on the basis of data. As such, it is very useful in decode situations. It is not necessarily the most code efficient decoding scheme, but it is always the most time efficient and time uniform decoding scheme.

For execution speed efficiency, do not put unnecessary instructions in loops. Look for ways to move instructions out of loops. It is frequently possible to move seemingly necessary instructions out of program loops. This is a speed improvement that usually costs little or no code.

4.9 STRUCTURED PROGRAMMING TECHNIQUES

The techniques of structured programming or top-down programming are excellent organizational tools and work well on large systems. However, these techniques have a basic implementation problem at the level of single-chip microcontroller in general and COPS microcontrollers in particular. Systems based on COPS devices are generally seeking maximum function with minimum memory.

Efficient COPS programming requires the elimination or minimization of redundant or duplicated code. Maximum sharing of common or related code is necessary. Partial sharing of routines is also common. Most subroutines in an efficient COPS program will have multiple entry points. There are branches into and out of routines that exist solely to reduce memory usage. All of this is in direct conflict with the top-down modular approaches. An efficient COPS program is not written by assembling independent blocks. That technique will use excessive code and could require a user to use a larger device than necessary. It is difficult, in an efficient COPS program, to extract independent modules other than the most basic functions.

The concepts of structured programming are still useful in defining the functions that must be performed and their inter-relationships. When the time comes to write the code within the memory limits of the microcontroller, the concepts fail. At this point, the user should use the approaches and techniques in this manual. Remember, the objective is to write an efficient COPS program thereby obtaining maximum function in minimum memory. Rarely, if ever, is the objective to write an easily readable program with modular, transportable functional blocks that exceed the memory capacity of the device.

5.0 Standard Programs

5.1 INTRODUCTION

This section contains a number of standard programs illustrating various techniques and the implementation of various functions. If the user wishes to use any of these programs, he or she should remember that maximum efficiency will be obtained by tailoring the program to the application. Copying the programs "as is" generally is not efficient.

5.2 MATH PACK

This section includes a variety of arithmetic routines, including the following:

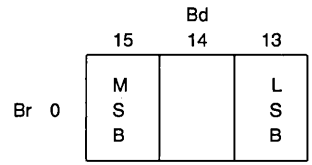
- Increment routines
- Decrement routines
- Integer Addition
- Integer Subtraction
- Binary Multiply
- Basic Arithmetic Package: Add, Subtract, Multiply, Divide
- Square Root
- Binary to BCD Conversion
- BCD to Binary Conversion

Typically, more than one implementation of a function is given.

5.2.1 Basic Increment Routines

Binary Routines

The following three routines have the same function: They perform a binary addition of 1 to a 12-bit binary number. The number is located in register 0, digits 15 through 13.



I

```
INCR:   LBI     0,13
        SC
INCR1:  CLRA
        ASC
        NOP
        XIS
        JP      INCR1
        RET

ROM Words Used: 9
Execution Time: 18
(instruction cycles)
```

II

```
INCR:   LBI     0,13
INCR1:  LD
        AISC    1
        JP      INCR2
        XIS
        JP      INCR1
        RET
INCR2:  X
        RET

        9
Data Dependent 6-16
```

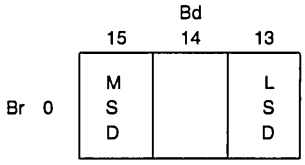
II

```
INCR:   LBI     0,13
        LD
        AISC    1
        AISC    15
        XIS
        LD
        AISC    1
        AISC    15
        XIS
        LD
        AISC    1
        NOP
        X
        RET

        14
        14
```

The preceding three examples illustrate an important point: The most code efficient method of implementing this function takes more time to execute than either of the other two implementations. This is a fairly common characteristic. Implementation II is, on the average, the fastest executing routine. Its main drawback is that its execution time is data dependent. This may not be significant. Implementation I uses and modifies the C register; the other implementations do not. All three routines use the accumulator.

BCD Routines

The following routines have the same function: They increment a three-digit BCD number by one.



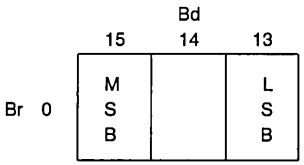

```
                I                          II                         II
INCR:    LBI     0,13       INCR:   LBI     0,13       INCR:   LBI     0,13
         SC                 INCR1:  LD                         LD
INCR1:   CLRA                       AISC    7                  AISC    7
         AISC    6                  JP      INCR2              AISC    9
         ASC                        XIS                        XIS
         ADT                        JP      INCR1              LD
         XIS                        RET                        AISC    7
         JP      INCR1      INCR2:  ADT                        AISC    9
         RET                        X                          XIS
                                    RET                        LD
                                                               AISC    7
                                                               ADT
                                                               X
                                                               RET

ROM Words Used: 9                   10                         14
Execution Time: 21          Data Dependent 7-17                14
(instruction cycles)
```

The same comments made for the binary routines are valid for the BCD routines.

5.2.2 Basic Decrement Routines

Binary Routines

The following routines take a 12-bit binary number and decrements it by one.

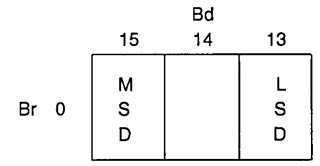



```
                I                          II                         II
DECR:    LBI     0,13       DECR:   LBI     0,13       DECR:   LBI     0,13
         RC                 DECR1:  LD                         LD
DECR1:   CLRA                       AISC    15                 AISC    15
         CASC                       JP      DECR2              XIS
         NOP                        X                          LD
         XIS                        RET                        AISC    15
         JP      DECR1      DECR2:  XIS                        XIS
         RET                        JP      DECR1              LD
                                    RET                        AISC    15
                                                               NOP
                                                               X
                                                               RET

ROM Words Used: 8                   9                          12
Execution Time: 18          Data Dependent 6-17                12
(instruction cycles)
```

As with the increment routines, the routine requiring the least code takes the most time.

BCD Routines

The following routines take a three-digit decimal number and decrement it by one.



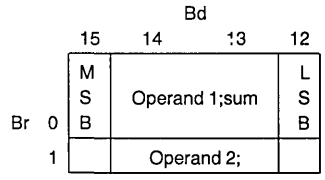

```
                I                          II                         II
DECR:     LBI       0,13       DECR:    LBI       0,13     DECR:    LBI       0,13
          RC                   DECR1:   LD                         LD
DECR1:    CLRA                          AISC      15               AISC      15
          CASC                          JP        DECR2            STII      9
          ADT                           X                          LD
          XIS                           RET                        AISC      15
          JP        DECR1     DECR2:   ADT                        STII      9
          RET                           XIS                        LD
                                        JP        DECR1            AISC      15
                                        RET                        ADT
                                                                   X
                                                                   RET
ROM Words Used: 8                       10                         12
Execution Time: 18                Data Dependent 6-20              12
(instruction cycles)
```

The same pattern is observed here as in the other similar routines.

5.2.3 Integer Addition

Binary Addition

The routine below is the basic addition routine. It illustrates the power of the exclusive OR argument of the LD, XIS, XDS, and X instructions. It also illustrates the conciseness that can come from intelligent data placement in RAM. As written, the routine is a 16-bit binary add, R1 + R0 → R0.



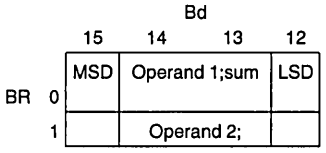

```
BINADD:  LBI    1,12   ;set-up B register
         RC             ;initialize Carry to 0
LOOP:    LD       1     ;fetch data from R1 and point to R0
         ASC            ;add RAM(B) + A + C → A
         NOP            ;defeat skip
         XIS      1     ;store result to R0, increment Bd, point to R1
         JP     LOOP    ;Loop control
         RET            ;all done, exit
```

ROM Words Used: 8

Execution Time: 23 instruction cycle times

BCD Addition

This routine is essentially the same as the binary add routine. A four-digit BCD add is illustrated. Again, R1 + R0 → R0.




```
BINADD:  LBI   1,12   ;initialize B to LSB
         RC           ;initialize Carry to 0
LOOP:    LD      1    ;fetch data from R1 and point to R0
         AISC    6    ;decimal adjust to force carry at 9 → 10
         ASC          ;add
         ADT          ;decimal correct if no carry
         XIS     1    ;store result in R0
         JP    LOOP
         RET
```

ROM Words Used: 9

Execution Time: 23 instruction cycle times

Both of these addition routines can be expanded up to 64 bits or 16 digits merely by changing the starting address, the Bd value in particular. Also note that the data could be placed at the other end of the register and XDS used in place of XIS.

Since the routine is essentially independent of data length and the exclusive OR feature of the LD, XIS, XDS, and X instructions permits easy transportation across data registers, a very versatile and compact routine can be created. Consider the following variation on the BCD addition routine:

```
ADD1:    LBI    3,0   ;R3+R0 → R0, 16-digit add
ADD2:    LBI    0,0   ;R3+R0 → R3, 16-digit add
ADD3:    LBI   1,10   ;R1+R2 → R2, 6-digit add
ADD4:    LBI   2,10   ;R1+R2 → R1, 6-digit add
         RC
LOOP:    LD      3
         AISC    6
         ASC
         ADT
         XIS     3
         JP    LOOP
         RET
```

Here we have the same routine able to work on two different sets of registers with different data lengths. Furthermore, either register in a given set can be the destination for the result. The controlling factor in all of this is simply the value in the B register at the start of the routine. The repeated LBI skip feature proves very useful in creating a multiple entry subroutine such as this one.

Variations on these basic two register additions similar to the techniques shown in the basic increment and decrement routines can be created. This is left as an exercise for the programmer. The most code efficient techniques have been illustrated here.

5.2.4 A Doubling Routine

A routine to double the value in a register is a simple outgrowth from the basic addition routine. This routine is illustrated before for a binary double. Data placement is the same as shown earlier.

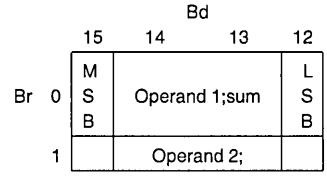
2 x R0 → R0, 16-bit binary

```
DOUBLE:  LBI   0,12
         RC
LOOP:    LD          ;RAM(B) → A, Br not changed
         ASC
         NOP
         XIS         ;A → RAM(B),increment Bd, Br unchanged
         JP    LOOP
         RET
```

The routine for a decimal double is derived from the BCD add routine in the same manner: the exclusive OR argument on the LD and XIS instructions is changed to 0 so that Br is not altered by those instructions.

An Example of the Effect of Data Placement in RAM

If assumed, in either of the addition routines presented earlier, that the data is not optimally placed at the end of registers, the following routine could be the result:



R1 + R0 ⟶ R0, 16-bit binary

```
BINADD:  LBI    1,10   ;initialize B
         RC             ;initialize Carry
LOOP:    LD       1
         ASC
         NOP
         XIS      1
         CBA            ;test for Bd > 13-if yes, done
         AISC     2
         JP     LOOP
         RET
```

ROM Words Used: 10

Execution Time: 31 instruction cycle times

In this example, inefficient data placement resulted in a 25 per cent code increase and a nearly 50 per cent increase in execution time. The message should be clear from this: Placement of data in RAM can have dramatic effects on the program.

5.2.5 Integer Subtract

These routines are the counterparts of the integer addition routines in Section 5.2.3. The RAM maps are the same as in that section.

Binary Subtraction

The routine as written below is a 16-bit binary subtraction (R1 − R0 ⟶ R1).

```
BINSUB:  LBI    0,12   ;initialize B register
         SC             ;set Carry for subtract
LOOP:    LD       1    ;fetch R value and point to R1
         CASC           ;subtract
         NOP
         XIS      1;save result in R1, increment Bd, point to R1
         JP     LOOP
         RET
```

BCD Subtraction

The BCD counterpart of the preceding is the following four-digit subtract routine (R1 − R0 ⟶ R1).

```
BCDSUB:  LBI    0,12   ;initialize B
         SC             ;set C for subtract
LOOP:    LD       1    ;fetch R0 value, point to R1
         CASC           ;subtract
         ADT            ;decimal correct (15 ⟶ 9)
         XIS      1    ;save in R1, increment Bd, point to R0
         JP     LOOP
         RET
```

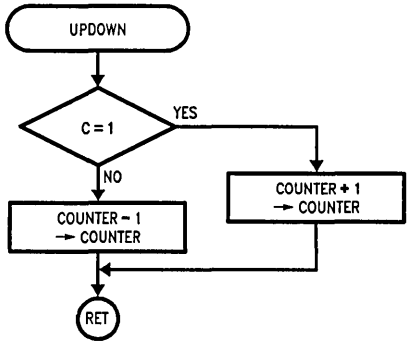
These routines are direct counterparts to the addition routines. The comments in Section 5.2.3 are equally valid for these subtract routines.

5.2.6 Up-Down Counters

The up-down counter routine is an extension or combination of the basic increment or decrement routines. Both an increment and a decrement have, effectively, been combined. The C register is used to distinguish between counting up or counting down. The basic flow for the routine and the RAM map is shown below in *Figure 5-1.*

		Bd		
	15	14	13	12
Br 2		Counter		
3		Not Used		TEMP STORE

The flow chart and RAM map are valid for both the binary and BCD versions of the routine. Two implementations of each are given: The first is a simple combination of the increment decrement routines. The second is a somewhat more sophisticated implementation which saves a little code but uses more RAM (one extra digit which is in TEMP STORE in the RAM map).



TL/DD/8800-11

FIGURE 5-1. Basic Flow for Up-Down Counter Routine

Binary Up-Down Counter

```
                I                                    II

UPDOWN:   LBI     2,12          UPDOWN:   LBI     3,12
          SKC                             CLRA
          JP      DOWN                    SKC
UP:       CLRA                            COMP
          ASC                   COUNT:    X       1;point to 2,12
          NOP
          XIS                   COUNT1:   LDD     3,12
          JP      UP                      ASC
          RET                             NOP
DOWN:     CLRA                            XIS
          CASC                            JP      COUNT1
          NOP                             RET
          XIS
          JP      DOWN
          RET
```

Version II of this routine loads 0 or 15 into a RAM location. Then the state of the carry controls addition or subtraction. Note that the location of the temporary data storage digit was chosen to use the exclusive OR capability of the X instruction to eliminate an instruction.

BCD Up-Down Counter

I

```
UPDOWN:  LBI     2,12
         SKC
         JP      DOWN
UP:      CLRA
         AISC    6
         ASC
         ADT
         XIS
         JP      UP
         RET
DOWN:    CLRA
         CASC
         ADT
         XIS
         JP      DOWN
         RET
```

II

```
UPDOWN:  LBI     3,12
         CLRA
         SKC
         AISC    9
COUNT:   X       1

COUNT1:  LDD     3,12
         AISC    6
         ASC
         ADT
         XIS
         JP      COUNT1
         RET
```

The comparison is the same as the binary routines. Version II here also illustrates another point. As written, Version II will execute (increment or decrement the four-digit counter) in 34 instruction cycle times. By merely moving the AISC 6 instruction from its present location to after the CLRA, the execution time is improved without any penalty.

```
         IIa
UPDOWN:  LBI   3,12
         CLRA
         AISC  6        ;build decimal correct into the stored constant
         SKC
         AISC  9
COUNT:   X     1
COUNT1:  LDD   3,12
         ASC
         ADT
         XIS
         JP    COUNT1
         RET
```

This routine is completely equivalent in function, approach, and amount of code as Version II. It executes faster, however, 31 instruction cycles rather than 34. The reason for the speed improvement is that an instruction, the AISC 6, was moved out of the loop and into the "main body" of the routine.

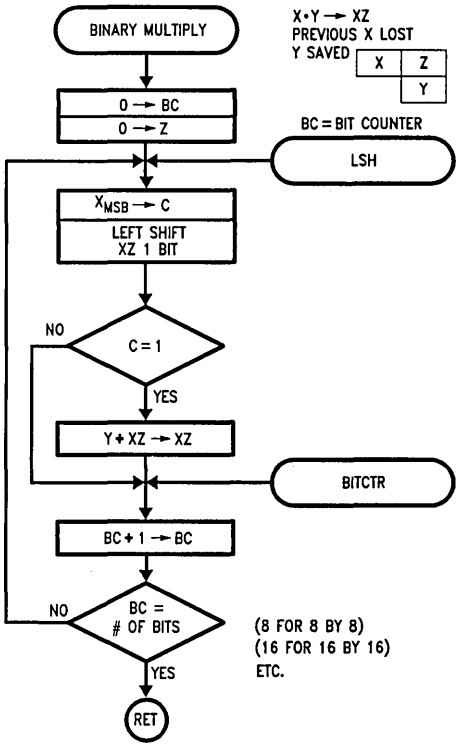
5.2.7 Binary Multiply

A routine for a 16 by 16 bit binary multiply is given below. A 32-bit product is generated. A RAM map for this routine is given below. A flow chart is in *Figure 5-2*.

	Bd 15	14	13	12	11	10	9	8	7
Br 0	X (multiplicand)				Z				BIT COUNTER
1	"0"				Y (multiplier)				NOT USED

The routine does the following:

X x Y → XZ, previous X lost; Y unchanged



TL/DD/8800–12

FIGURE 5-2. Binary Multiply

```
BINMULT:  LBI   0,7
          STII  0         ;clear bit counter and Z
          STII  0
          STII  0
          STII  0
          STII  0
          LBI   1,12
          STII  0
          STII  0
          STII  0
          STII  0
          LBI   0,8
          RC
LSH:      LD              ;left shift XZ 1 bit, putting XMSB into C
          ASC
          NOP
EX:       XIS
          JP    LSH
          SKC
          JP    BITCTR
BINADD:   RC
          LBI   1,8       ;Y + XZ → XZ
ADD:      LD    1
          ASC
          NOP
          XIS   1
          JP    ADD
BITCTR:   LBI   0,7       ;increment bit counter and test if done
          LD
          AISC  1
          JP    EX
          RET
```

5.2.8 Basic Arithmetic Package

This section includes the basic arithmetic functions, add, subtract, multiply, and divide. The routines are written as a cohesive unit. They are for eight-digit floating-point fully algebraic arithmetic. *Figures 5-4* through *5-7* are the RAM map and flow chart for these routines.

Both decimal and binary (hexadecimal) versions of these routines are provided. The flow charts and RAM map are valid for both these versions.

The routines listed in *Figures 5-3* and *5-8* have an arbitrary error handling routine; the error is merely flagged by setting the decimal point and sign position to 15. The user can modify this to a perhaps more useful arrangement.

```
COP CROSS ASSEMBLER     PAGE:    1
MAINPR

 1                ;BASIC BCD FLOATING POINT ARITHMETIC ROUTINES
 2                ;
 3                ;REGISTER 0 = X,REGISTER 1 = Y, REGISTER 2 = Z
 4                ;
 5                ;THE ROUTINES ARE FOR 8 DIGIT,BCD,FULLY ALGEBRAIC ADD,SUBTRACT
 6                ;MULTIPLY AND DIVIDE. ALL ROUTINES ARE FULLY FLOATING POINT.
 7                ;THE ROUTINES ASSUME AN 8 DIGIT MANTISSA, A SIGN DIGIT, AND A
 8                ;DECIMAL POINT DIGIT.THE DECIMAL POINT DIGIT IS A DECIMAL POINT
 9                ;POSITION INDICATOR,I.E.,A DEC. PT.POSITION OF 0 INDICATES
10                ;THAT THE DECIMAL POINT IS PLACED AFTER THE LSD OF THE NUMBER;
11                ;DEC.PT. POSITION OF 7 INDICATES THAT THE DECIMAL POINT IS
12                ;PLACED AFTER THE MSD OF THE NUMBER. OTHER NUMBERS CORRESPOND
13                ;IN THE SAME MANNER TO INTERMEDIATE DIGIT.
14                ;
15                ;THE ROUTINES ALSO ASSUME THAT THERE IS A GUARD OR OVERFLOW
16                ;DIGIT FOR THE NUMBERS.THE MANTISSA IS 8 DIGITS PLUS THE GUARD
17                ;DIGIT FOR A TOTAL OF 9 DIGITS.THE GUARD DIGIT IS FOR INTERNAL
18                ;USE ONLY AND IS NOT AVAILABLE ON INPUT OR OUTPUT.
19                ;
20                ;THE ROUTINES CAN BE MODIFIED FOR HEX OR BINARY ARITHMETIC.
21                ;AS THE ALGORITHMS ARE NOT NUMBER BASE DEPENDENT(EXCEPT FOR
22                ;OBVIOUS THINGS LIKE OVERFLOW TESTS, ETC. WHICH WOULD HAVE TO
23                ;BE MODIFIED TO ACCOMMODATE THE NUMBER BASE USED).
24                ;
25                ;THE CODE AS WRITTEN SHOULD WORK IN COP420 AND LARGER DEVICES.
26                ;THE ROUTINES ARE WRITTEN AS SUBROUTINES CALLED BY A MAIN
27                ;PROGRAM. ONE LEVEL OF SUBROUTINE IS USED BY THE ARITHMETIC
28                ;ROUTINES. COMPARABLE ROUTINES CAN BE WRITTEN FOR THE COP410
29                ;BUT SOME CHANGES ARE REQUIRED. THE ALGORITHM IS STILL VALID
30                ;ALTHOUGH THE IMPLEMENTATION IS SOMEWHAT DIFFERENT.
31                ;
32     0022       SAVE1   =       2,2
33     000F       XGUARD  =       0,15
34     000E       XMSD    =       0,14
35     0007       XLSD    =       0,7
36     0006       ROUND   =       0,6
37     0001       XSIGN   =       0,1
38     0000       XDP     =       0,0
39     001F       YGUARD  =       1,15
40     001E       YMSD    =       1,14
41     0017       YLSD    =       1,7
42     0011       YSIGN   =       1,1
43     0010       YDP     =       1,0
44     002F       ZGUARD  =       2,15
45     002E       ZMSD    =       2,14
46     0027       ZLSD    =       2,7
47     0021       ZSIGN   =       2,1
48     0020       ZDP     =       2,0
49     003F       FLAGS   =       3,15
50     0030       OFLOW   =       3,0
51                ;       =       3,0
```

FIGURE 5-3. BCD Arithmetic Package (Sheet 1 of 9)

```
COP CROSS ASSEMBLER    PAGE:    2
MAINPR
 52     0000            .PAGE   0
 53 000 00               CLRA
 54 001 0F               LBI     0,0     ;CLEAR ALL THE RAM
 55 002 53               AISC    3
 56 003 12      RAMCLR: XABR
 57 004 81               JSRP    CLEAR
 58 005 12               XABR
 59 006 5F               AISC    15
 60 007 C9               JP      TESTG
 61 008 C3               JP      RAMCLR
 62             ;
 63             ;*************************************************************
 64             ;FOLLOWING CODE-TO NEXT LINE OF **-IS FOR CONTROL ONLY
 65             ;
 66 009 335F    TESTG:  OGI     15      ;PUT G LINES HIGH FOR READING G
 67             ;
 68                   ;USING G LINES FOR PRIMITIVE CONTROL TO SELECT ADD,SUB
 69                   ;MULTIPLY OR DIVIDE-WILL ENTER NUMBERS IN BREAKPOINT
 70                   ;MODE USING MODIFY COMMAND
 71                   ;
 72 00B 3301            SKGBZ   0
 73 00D D3              JP      TESTG1
 74 00E 3E              LBI     FLAGS
 75 00F 70              STII    0       ;RESET BIT 2 FOR ADD
 76 010 6840    JSRALN: JSR     ALIGN
 77 012 C9              JP      TESTG
 78 013 3311    TESTG1: SKGBZ   1
 79 015 D9              JP      TESTG2
 80 016 3E              LBI     FLAGS   ;SET SUBTRACT BIT
 81 017 74              STII    4       ;SET BIT 2 FOR SUBTRACT
 82 018 D0              JP      JSRALN
 83 019 3303    TESTG2: SKGBZ   2
 84 01B DF              JP      TESTG3
 85 01C 3E              LBI     FLAGS
 86 01D 70              STII    0       ;RESET BIT 3 FOR DIVIDE
 87 01E E4              JD      JSMD
 88 01F 3313    TESTG3: SKGBZ   3
 89 021 C9              JP      TESTG
 90 022 3E              LBI     FLAGS
 91 023 78              STII    8       ;SET BIT 3 FOR MULTIPLY
 92 024 6940    JSMD:   JSR     MULDIV
 93 026 C9              JP      TESTG
 94             ;
 95             ;PRECEDING CODE FOR CONTROL ONLY,HAS NOTHING TO DO WITH THE
 96             ;ARITHMETIC ALGORITHMS
 97             ;*************************************************************
 98             ;
 99     0040            .PAGE   1
100             ; THIS IS THE ALIGN ROUTINE FOR ADD/SUBTRACT.  IT MAKES THE
101             ; DECIMAL POSITIONS OF THE TWO NUMBERS EQUAL BEFORE ADD OR
102             ; SUBTRACT TAKES PLACE.  THE ROUTINE ASSUMES THAT THE NUMBERS
```

FIGURE 5-3. BCD Arithmetic Package (Sheet 2 of 9)

```
COP CROSS ASSEMBLER    PAGE:   3
MAINPR

                ; ARE RIGHT JUSTIFIED ON ENTRY.  DECIMAL POINT POSITION VALUES
                ; ARE RESTRICTED TO NUMBERS BETWEEN 0 - 8 (SINCE WE ARE ONLY
                ; DOING 8 DIGIT ROUTINES).  ROUTINE ONLY REQUIRED FOR FLOATING
                ; POINT ADD/SUBTRACT ALGORITHMS
                ;
040 0F      ALIGN:  LBI     XDP
041 15              LD      1
042 21              SKE             ;TEST DP0=DP1(DPX=DPY)
043 C6              JP      ALIGN2
044 6100            JMP     ADDSUB  ;IF EQUAL,PROCEED TO ADD/SUBTRACT
046 10      ALIGN2: CASC            ;TEST DP0 > DP1
047 D6              JP      DP0GT1  ;YES
048 0D      DP0LT1: LBI     XMSD    ;DP0<DP1.IF XMSD NOT 0,RIGHT SHIFT
049 00              CLRA            ;M1,ELSE LEFT SHIFT M0
04A 21              SKE
04B D1              JP      R1RSFT
04C 87      R0LSFT: JSRP    LSFTR0
04D 0F              LBI     XDP
04E 1F      DPPL1:  LBI     YDP
04F B5              JSRP    PLUS1   ;MODIFY DP AFTER SHIFT
050 C0              JP      ALIGN
051 8E      R1RSFT: JSRP    RSFTR1
052 1F              LBI     YDP
053 0F      DPMIN1: LBI     XDP
054 AD              JSRP    MINUS1
055 C0              JP      ALIGN
056 1D      DP0GT1: LBI     YMSD    ;TESTING MSD OF M1 NOT 0
057 00              CLRA
058 21              SKE
059 DC              JP      R0RSFT
05A 85      R1LSFT: JSRP    LSFTR1
05B CE              JP      DPPL1
05C 8D      R0RSFT: JSRP    RSFTR0
05D D3              JP      DPMIN1
                ;
0080                .PAGE   2
                ;THESE ARE THE BASIC REQUIRED SUBROUTINES FOR THE ARITHMETIC
                ;ROUTINES-COP420 AND LARGER CODE
                ;
080 0F      CLEAR0: LBI     0,0
081 00      CLEAR:  CLRA
082 04              XIS
083 81              JP      CLEAR
084 48              RET
085 3397    LSFTR1: LBI     YLSD
087 3387    LSFTR0: LBI     XLSD
089 00      LSFTX:  CLRA
08A 04      LSFT:   XIS
08B 8A              JP      LSFT
08C 48              RET
08D 0E      RSFTR0: LBI     0,15
08E 1E      RSFTR1: LBI     1,15
```

FIGURE 5-3 BCD Arithmetic Package (Sheet 3 of 9)

```
COP CROSS ASSEMBLER    PAGE:   4
MAINPR

155 08F 00     RSFTRX: CLRA
156 090 07     RSFT:   XDS
157 091 23A2           XAD     SAVE1   ;SAVE VALUE TEMPORARILY
158 093 4E             CBA             ;ONLY WANT 8 DIGIT SHIFT
159 094 59             AISC    9
160 095 99             JP      DONE
161 096 2322           LDD     SAVE1   ;FETCH SAVED VALUE
162 098 90             JP      RSFT
163 099 2322   DONE:   LDD     SAVE1
164 09B 48             RET
165 09C 32     BCDADD: RC
166 09D 15     BCD1:   LD      1       ;TWO REGISTER BCD ADDITION
167 09E 56             AISC    6
168 09F 30             ASC
169 0A0 4A             ADT
170 0A1 14             XIS     1
171 0A2 9D             JP      BCD1
172 0A3 48             RET
173 0A4 22     BCDSUB: SC              ;TWO REGISTER BCD SUBTRACTION
174 0A5 15     BCDS1:  LD      1
175 0A6 10             CASC
176 0A7 4A             ADT
177 0A8 14             XIS     1
178 0A9 A5             JP      BCDS1
179 0AA 48             RET
180 0AB 2F     ZDPMN1: LBI     ZDP
181 0AC 3F     OFLMN1: LBI     OFLOW
182 0AD 05     MINUS1: LD              ;SUBTRACT 1 FROM MEMORY
183 0AE 5F             AISC    15
184 0AF 44             NOP
185 0B0 06     PLUS1A: X
186 0B1 48             RET
187 0B2 0F     XDPPL1: LBI     XDP
188 0B3 3F     OFLPL1: LBI     OFLOW
189 0B4 2F     ZDPPL1: LBI     ZDP
190 0B5 05     PLUS1:  LD              ;ADD 1 TO MEMORY
191 0B6 51             AISC    1
192 0B7 B0             JP      PLUS1A  ;WILL SKIP IF GREATER THAN 15
193 0B8 06             X
194 0B9 49             RETSK
195 0BA 25     XFER2:  LD      2
196 0BB 24             XIS     2
197 0BC BA             JP      XFER2
198 0BD 48             RET
199            ;

200     0100           .PAGE   4
201            ;THIS IS THE ADD/SUBTRACT ROUTINE.  ROUTINE IS FOR 8 DIGITS.
202            ;FLOATING POINT, FULLY ALGEBRAIC.
203            ;
204 100 3E     ADDSUB: LBI     FLAGS
205 101 03             SKMBZ   2       ;TEST IF SHOULD SUBTRACT
206 102 D0             JP      CHNGMO  ;CHANGE SIGN R0(X) IF SUBTRACT
```

FIGURE 5-3. BCD Arithmetic Package (Sheet 4 of 9)

COP CROSS ASSEMBLER PAGE: 5
MAINPR

```
207 103 3381   ADSB1:  LBI     XSIGN   ;NOW TEST FOR SIGNS EQUAL
208 105 15             LD      1
209 106 21             SKE
210 107 D7             JP      SUB     ;NOT EQUAL,HENCE SUBTRACT
211 108 3387   ADD:    LBI     XLSD
212 10A 9C             JRP     BCDADD  ;R1+RO --> R1,(Y+X --> Y)
213 10B 1E     ERRCHK: LBI     YGUARD  ;TEST FOR OVERFLOW
214 10C 00             CLRA            ;IF 1,15(YGUARD) NOT 0,UNDERFLOW
215 10D 21             SKE
216 10E ED             JP      UNDRFL
217 10F 48             RET
218 110 3381   CHNGMO: LBI     XSIGN   ;CHANGE SIGN OF RO(X)
219 112 05             LD
220 113 58             AISC    8
221 114 44             NOP
222 115 06             X
223 116 C3             JP      ADSB1
224 117 3387   SUB:    LBI     XLSD
225 119 A4             JSRP    BCDSUB  ;R1-RO --> R1,(Y-X --> Y)
226 11A 20             SKC             ;SEE IF MUST COMPLEMENT
227 11B DD             JP      COMPL
228 11C CB             JP      ERRCHK
229 11D 3397   COMPL:  LBI     YLSD    ;NEGATIVE RESULT,COMPLEMENT
230 11F 22             SC
231 120 00     COMPL1: CLRA
232 121 06             X
233 122 10             CASC
234 123 4A             ADT
235 124 04             XIS
236 125 E0             JP      COMPL1
237 126 3391           LBI     YSIGN   ;NOW CHANGE SIGN OF R1(Y)
238 128 05             LD
239 129 58             AISC    8
240 12A 44             NOP
241 12B 06             X
242 12C CB             JP      ERRCHK
243 12D 8E     UNDRFL: JSRP    RSFTR1  ;DO AN UNDERFLOW
244 12E 1F             LBI     YDP     ;ERROR IF YDP IS 0 WHEN UNDERFLOW
245 12F AD             JSRP    MINUS1
246 130 5F             AISC    15
247 131 F3             JP      ERROR
248 132 48             RET
249 133 1F     ERROR:  LBI     YDP
250 134 7F             STII    15      ;15 --> YDP & YSIGN FOR ERROR
251 135 7F             STII    15
252 136 48             RET
253     0140           .PAGE   5
254            ;MULTIPLY,DIVIDE ROUTINES. FLOATING POINT,8 DIGIT
255            ;
256 140 3387   MULDIV: LBI     XLSD
257 142 BA             JSRP    XFER2   ;MO --> M2,X --> Z, THEN CLEAR X
258 143 25             LD      2       ;TRANSFER DP AND SIGN ALSO
```

FIGURE 5-3. BCD Arithmetic Package (Sheet 5 of 9)

COP CROSS ASSEMBLER PAGE: 6
MAINPR

```
259 144 24            XIS     2
260 145 25            LD      2
261 146 26            X       2
262 147 80            JSRP    CLEARO  ;CLEAR MO
263 148 3E            LBI     FLAGS   ;NOW TEST IF MULTIPLY OR DIVIDE
264 149 13            SKMBZ   3
265 14A 61C0          JMP     MULPLY
266             ;
267 14C 22    DIVIDE: SC              ;MO/M1 --> MO,(X/Y --> X)
268 14D 1F            LBI     YDP     ;DP2-DP0 --> DP2,(DPZ-DPX --> DPZ)
269 14E 35            LD      3
270 14F 10            CASC
271 150 44            NOP
272 151 06            X
273 152 3F            LBI     OFLOW   ;15 TO OFLOW DIGIT IF BORROW,ELSE 0
274 153 00            CLRA
275 154 20            SKC
276 155 40            COMP
277 156 06            X
278 157 3397  DIV1A:  LBI     YLSD
279 159 A4            JSRP    BCDSUB  ;MO - M1 TO MO,M1 SAVED
280 15A 20            SKC             ;PART OF THE REPEATED SUBTRACT FEATURE
281 15B E2            JP      DIV3A
282 15C 33A7  DIV3:   LBI     ZLSD    ;DIVIDE BY 0 CHECK
283 15E B5            JSRP    PLUS1
284 15F D7            JP      DIV1A   ;ALL OK,CONTINUE
285 160 6189          JMP     DIVBY0
286 162 3397  DIV3A:  LBI     YLSD
287 164 9C            JSRP    BCDADD  ;RESTORE VALUE
288 165 0F            LBI     XDP
289 166 05            LD
290 167 57            AISC    7
291 168 617F          JMP     DIV1B   ;TESTING DP FOR FINISHED
292 16A 2E    DIV4:   LBI     ZGUARD
293 16B 00            CLRA
294 16C 21            SKE
295 16D 61E5          JMP     MDEND1
296 16F 3F            LBI     OFLOW
297 170 21            SKE             ;TEST OVERFLOW DIGIT
298 171 F8            JP      DIV4A
299 172 2F            LBI     ZDP     ;TEST DP2(ZDP) >= 9
300 173 05            LD
301 174 57            AISC    7
302 175 F8            JP      DIV4A
303 176 61E5          JMP     MDEND1
304 178 B4    DIV4A:  JSRP    ZDPPL1  ;DP2+1 --> DP2,(ZDP+1 --> ZDP)
305 179 6181          JMP     DIV1B2
306 17B B3            JSRP    OFLPL1  ;INCREMENT OVERFLOW DIGIT
307 17C 44            NOP             ;DEFEAT SKIP
308 17D 6181          JMP     DIV1B2
309 17F B2    DIV1B:  JSRP    XDPPL1  ;DP0 + 1 --> DP0
310 180 44            NOP
```

FIGURE 5-3. BCD Arithmetic Package (Sheet 6 of 9)

```
COP CROSS ASSEMBLER    PAGE:   7
MAINPR

311 181 33A7   DIV1B2: LBI     ZLSD
312 183 89             JSRP    LSFTX
313 184 3387           LBI     XLSD
314 186 8A             JSRP    LSFT
315 187 6157           JMP     DIV1A
316 189 621B   DIVBY0: JMP     MDERR

317     01C0           .PAGE   7
318 1C0 32     MULPLY: RC              ;DP1+DP2 --> DP2,(DPY+DPZ --> DPZ)
319 1C1 1F             LBI     YDP
320 1C2 35             LD      3
321 1C3 30             ASC
322 1C4 44             NOP
323 1C5 06             X
324 1C6 00             CLRA            ;1 TO OFLOW IF CARRY,ELSE 0
325 1C7 20             SKC
326 1C8 CA             JP      MUL1A
327 1C9 51             AISC    1
328 1CA 3F     MUL1A:  LBI     OFLOW
329 1CB 06             X
330 1CC 33A7   MUL1:   LBI     ZLSD
331 1CE 05             LD
332 1CF 5F             AISC    15      ;LSD CONTROLLING REPEATED ADDS
333 1D0 D6             JP      MUL2
334 1D1 06             X
335 1D2 3397           LBI     YLSD    ;M0 + M1  -->  M0,(X+Y --> X)
336 1D4 9C             JSRP    BCDADD
337 1D5 CC             JP      MUL1
338 1D6 8D     MUL2:   JSRP    RSFTR0
339 1D7 2E             LBI     ZGUARD
340 1D8 90             JSRP    RSFT
341 1D9 B2             JSRP    XDPPL1
342 1DA 58             AISC    8
343 1DB CC             JP      MUL1    ;PRECEDING IS DP ADJUST
344 1DC 78     MUL3:   STII    8
345 1DD 3387           LBI     XLSD
346 1DF 00     MUL3X:  CLRA            ;TEST M0=0(X=0)
347 1E0 21             SKE
348 1E1 6212           JMP     MUL5
349 1E3 04             XIS
350 1E4 DF             JP      MUL3X
351 1E5 2E     MDEND1: LBI     ZGUARD
352 1E6 00             CLRA
353 1E7 21             SKE
354 1E8 ED             JP      MDX
355 1E9 2F             LBI     ZDP
356 1EA 05             LD
357 1EB 57             AISC    7       ;TEST >= 9
358 1EC F7             JP      MDEND2
359 1ED 2E     MDX:    LBI     ZGUARD
360 1EE 8F             JSRP    RSFTRX
361 1EF 3386           LBI     ROUND   ;SAVE VALUE FOR ROUNDING
362 1F1 06             X
```

FIGURE 5-3. BCD Arithmetic Package (Sheet 7 of 9)

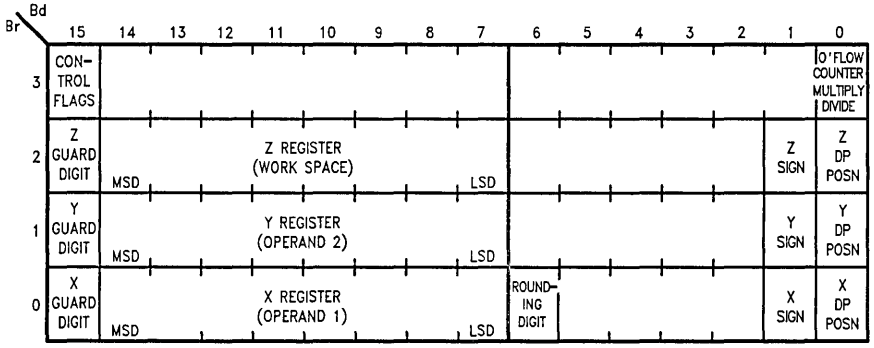
COP CROSS ASSEMBLER PAGE: 8
MAINPR

```
363 1F2 AB             JSRP    ZDPMN1
364 1F3 05             LD              ;TEST DP2(ZDP) = 15
365 1F4 51             AISC    1
366 1F5 F7             JP      MDEND2
367 1F6 AC             JSRP    OFLMN1  ;SUBTRACT 1 FROM OVERFLOW DIGIT
368 1F7 2F     MDEND2: LBI     2,0     ;TRANSFER R2 TO R0
369 1F8 BA             JSRP    XFER2
370 1F9 3391           LBI     YSIGN   ;ADD SIGNS AND PUT TO M0(X)
371 1FB 15             LD      1
372 1FC 31             ADD
373 1FD 06             X
374 1FE 3F             LBI     OFLOW   ;TEST OVERFLOW DIGIT
375 1FF 05             LD
376 200 51             AISC    1
377 201 C3             JP      MDEND4  ;NOT 15
378 202 DB             JP      MDERR   ;IS 15,NUMBER TOO BIG
379 203 00     MDEND4: CLRA            ;NOW TEST DIGIT > 0
380 204 21             SKE
381 205 80             JSRP    CLEAR0  ;IS NON ZERO,CLEAR M0
382 206 3387   MDRJ:   LBI     XLSD    ;RIGHT JUSTIFY THE RESULT
383 208 00             CLRA
384 209 21             SKE
385 20A 48             RET             ;IF LSD NON ZERO,STOP
386 20B 0F             LBI     XDP     ;IF DP PSN = 0,STOP
387 20C 05             LD
388 20D 5F             AISC    15
389 20E 48             RET
390 20F 06             X               ;ELSE,DECREMENT BY 1 AND CONTINUE
391 210 8D             JSRP    RSFTR0
392 211 C6             JP      MDRJ
393 212 2F     MUL5:   LBI     ZDP     ;TEST DP2(ZDP) = 0
394 213 00             CLRA
395 214 21             SKE
396 215 D8             JP      MUL3A
397 216 40             COMP            ;15 TO DP2(ZDP)
398 217 06             X
399 218 AD     MUL3A:  JSRP    MINUS1  ;DP2(ZDP) - 1 --> DP2(ZDP)
400 219 61D6           JMP     MUL2
401 21B 0E     MDERR:  LBI     0,15
402 21C 7F             STII    15
403 21D 7F             STII    15
404 21E 48             RET
405                    .END
```

FIGURE 5-3. BCD Arithmetic Package (Sheet 8 of 9)

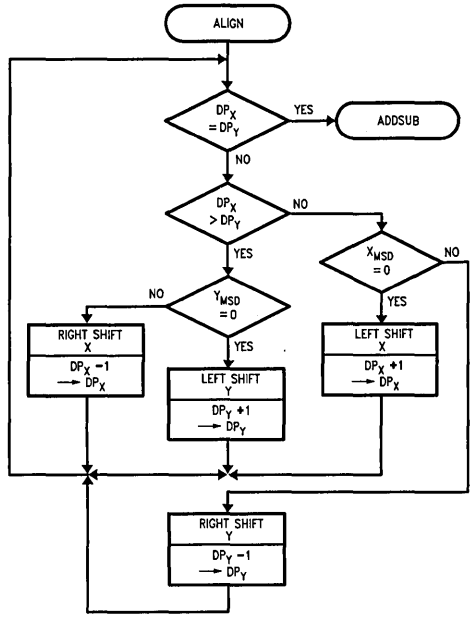
```
COP CROSS ASSEMBLER     PAGE:    9
MAINPR

ADD     0108 *   ADDSUB 0100     ADSB1  0103     ALIGN  0040
ALIGN2  0046     BCD1   009D     BCDADD 009C     BCDS1  00A5
BCDSUB  00A4     CHNGM0 0110     CLEAR  0081     CLEAR0 0080
COMPL   011D     COMPL1 0120     DIV1A  0157     DIV1B  017F
DIV1B2  0181     DIV3   015C *   DIV3A  1062     DIV4   016A *
DIV4A   0178     DIVBY0 0189     DIVIDE 014C *   DONE   0099
DP0GT1  0056     DP0LT1 0048 *   DPMIN1 0053     DPPL1  004E
ERRCHK  010B     ERROR  0133     FLAGS  003F     JSMD   0024
JSRALN  0010     LSFT   008A     LSFTR0 0087     LSFTR1 0085
LSFTX   0089     MDEND1 01E5     MDEND2 01F7     MDEND4 0203
MDERR   021B     MDRJ   0206     MDC    01ED     MINUS1 00AD
MUL1    01CC     MUL1A  01CA     MUL2   01D6     MUL3   01DC *
MUL3A   0218     MUL3X  01DF     MUL5   0212     MULDIV 0140
MULPLY  01C0     OFLMN1 00AC     OFLOW  0030     OFLPL1 00B3
PLUS1   00B5     PLUS1A 00B0     ROLSFT 004C *   RORSFT 005C
R1LSFT  005A *   R1RSFT 0051     RAMCLR 0003     ROUND  0006
RSFT    0090     RSFTR0 008D     RSFTR1 008E     RSFTRX 008F
SAVE1   0022     SUB    0117     TESTG  0009     TESTG1 0013
TESTG2  0019     TESTG3 001F     UNDRFL 012D     XDP    0000
XDPPL1  00B2     XFER2  00BA     XGUARD 000F *   XLSD   0007
XMSD    000E     XSIGN  0001     YDP    0010     YGUARD 001F
YLSD    0017     YMSD   001E     YSIGN  0011     ZDP    0020
ZDPMN1  00AB     ZDPPL1 00B4     ZGUARD 002F     ZLSD   0027
ZMSD    002E *   ZSIGN  0021 *
NO ERROR LINES
    356 ROM WORDS USED
COP     420     ASSEMBLY
SOURCE CHECKSUM = FBE9
INPUT FILE      ABDUL10:ARITH.SRC    VN:    20
```

FIGURE 5-3. BCD Arithmetic Package (Sheet 9 of 9)



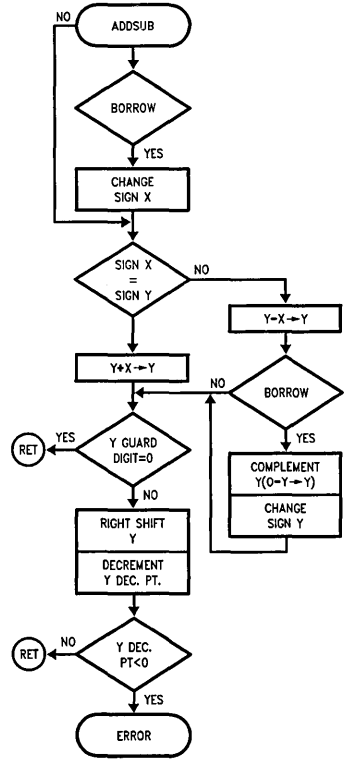
TL/DD/8800–13

FIGURE 5-4. RAM Map—Basic Arithmetic Routines



TL/DD/8800-14

FIGURE 5-5. Align Routine for Add/Subtract



TL/DD/8800-15

FIGURE 5-6. Fully Algebraic Add/Subtract

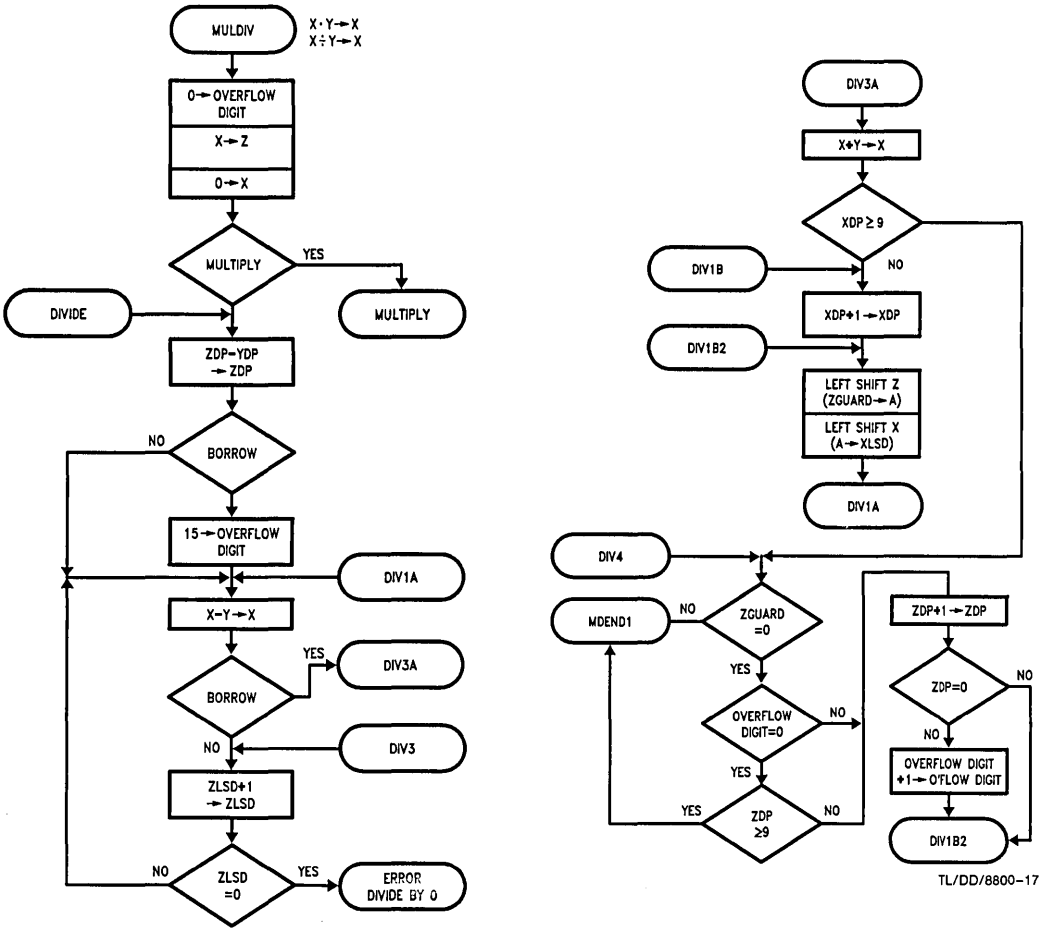
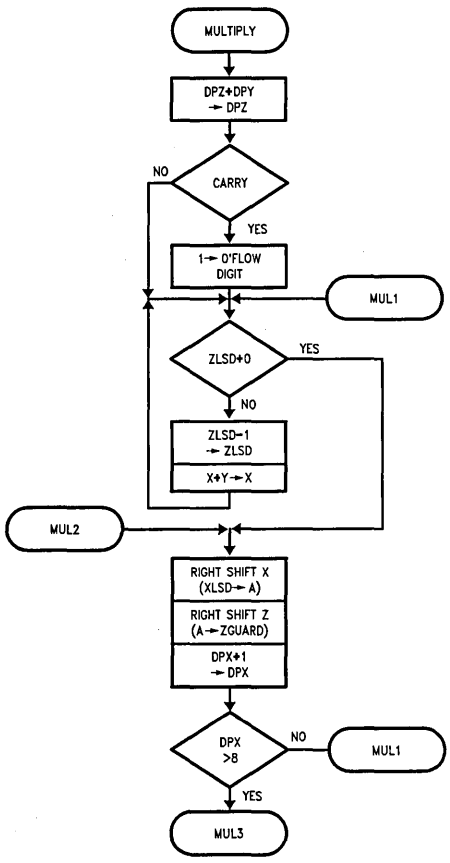
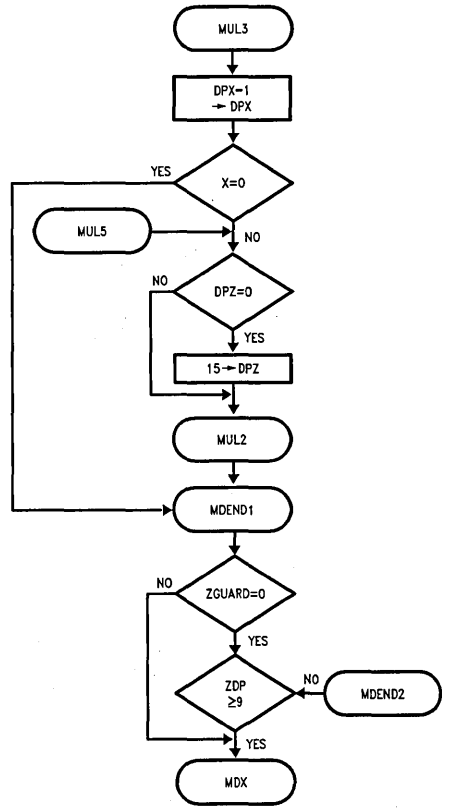


FIGURE 5-7. Multiply/Divide (Sheet 1 of 3)

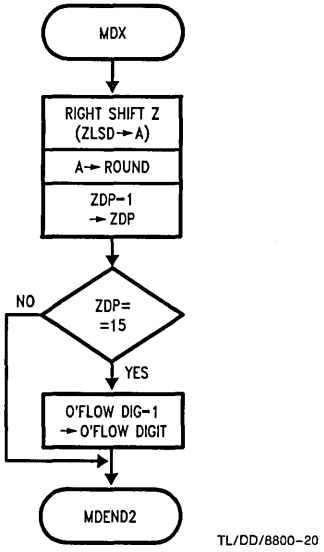


TL/DD/8800-18

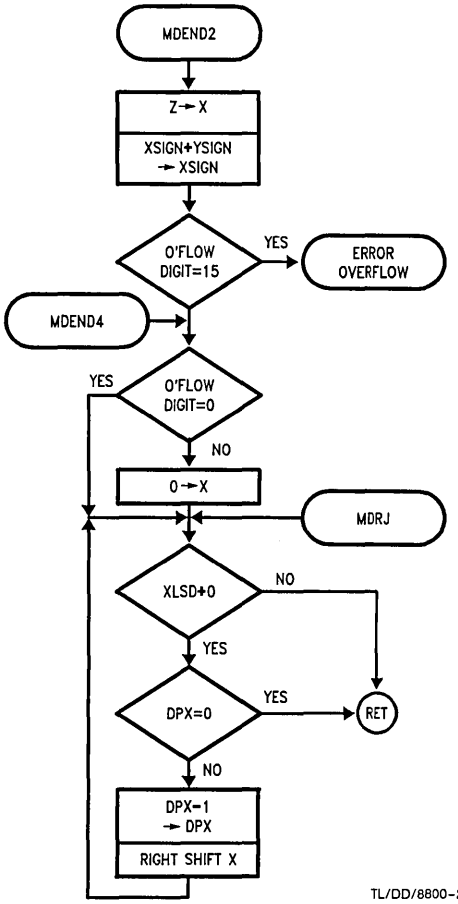


TL/DD/8800-19

FIGURE 5-7. Multiply/Divide (Sheet 2 of 3)



TL/DD/8800-20



TL/DD/8800-21

FIGURE 5-7. Multiply/Divide (Sheet 3 of 3)

```
COP CROSS ASSEMBLER     PAGE:    1
HXMATH     HEXADECIMAL ARITHMETIC

 1              ;BASIC HEXADECIMAL(BINARY) FLOATING POINT ARITHMETIC ROUTINES
 2              ;
 3              ;REGISTER 0 = X,REGISTER 1 = Y, REGISTER 2 = Z
 4              ;
 5              ;THE ROUTINES ARE FOR 8 DIGIT,HEX,FULLY, ALGEBRAIC ADD,SUBTRACT
 6              ;MULTIPLY AND DIVIDE. ALL ROUTINES ARE FULLY FLOATING POINT.
 7              ;THE ROUTINES ASSUME AN 8 DIGIT MANTISSA, A SIGN DIGIT, AND A
 8              ;HEX POINT DIGIT. THE HEX POINT DIGIT IS A HEX POINT
 9              ;POSITION INDICATOR,I.E.,A HEX PT.POSITION OF 0 INDICATES
10              ;THAT THE HEX POINT IS PLACED AFTER THE LSD OF THE NUMBER;
11              ;HEX POINT POSITION OF 7 INDICATES THAT THE HEX POINT IS
12              ;PLACED AFTER THE MSD OF THE NUMBER. OTHER NUMBERS CORRESPOND
13              ;IN THE SAME MANNER TO INTERMEDIATE DIGITS.
14              ;
15              ;THE ROUTINES ALSO ASSUME THAT THERE IS A GUARD OR OVERFLOW
16              ;DIGIT FOR THE NUMBERS.THE MANTISSA IS 8 DIGITS PLUS THE GUARD
17              ;DIGIT FOR A TOTAL OF 9 DIGITS.THE GUARD DIGIT IS FOR INTERNAL
18              ;USE ONLY AND IS NOT AVAILABLE ON INPUT OR OUTPUT.
19              ;
20              ;THE ROUTINES ARE USABLE AS IS FOR BINARY ARITHMETIC DUE TO
21              ;THE OBVIOUS RELATIONSHIP BETWEEN HEX AND BINARY. THE ONLY
22              ;ADVERSE EFFECT IS THAT THE RAM IS NOT OPTIMALLY USED IF
23              ;BINARY ARITHMETIC IS DESIRED.  NONETHELESS THE ROUTINES
24              ;ARE FULLY FUNCTIONAL FOR BINARY ARITHMETIC.
25              ;
26              ;THE CODE AS WRITTEN SHOULD WORK IN COP420 AND LARGER DEVICES.
27              ;THE ROUTINES ARE WRITTEN AS SUBROUTINES CALLED BY A MAIN
28              ;PROGRAM.  ONE LEVEL OF SUBROUTINE IS USED BY THE ARITHMETIC
29              ;ROUTINES.  COMPARABLE ROUTINES CAN BE WRITTEN FOR THE COP410
30              ;BUT SOME CHANGES ARE REQUIRED. THE ALGORITHM IS STILL VALID
31              ;ALTHOUGH THE IMPLEMENTATION IS SOMEWHAT DIFFERENT.
32              ;
33              ;IT WILL BE NOTED THAT THESE HEX ROUTINES DIFFER ONLY SLIGHTLY
34              ;FROM THE EQUIVALENT DECIMAL ROUTINES.  WITH THIS INFORMATION
35              ;IT WOULD BE POSSIBLE, IF IT WERE NECESSARY, TO WRITE A
36              ;COMMON ROUTINE AND DO THE ARITHMETIC IN HEX OR DECIMAL WHICH-
37              ;EVER WAS REQUIRED.  THE EXTRA CODE TO DO THIS WOULD NOT BE
38              ;SIGNIFICANT IF THE APPLICATION GENUINELY REQUIRED THIS DUAL
39              ;CAPABILITY.
40              ;
41                      .TITLE  HXMATH,'HEXADECIMAL ARITHMETIC'
42     0022     SAVE1   =       2,2
43     000F     XGUARD  =       0,15
44     000E     XMSD    =       0,14
45     0007     XLSD    =       0,7
46     0006     ROUND   =       0,6
47     0001     XSIGN   =       0,1
48     0000     XDP     =       0,0
49     001F     YGUARD  =       1,15
50     001E     YMSD    =       1,14
51     0017     YLSD    =       1,7
52     0011     YSIGN   =       1,1
```

FIGURE 5-8. Binary (Hexadecimal) Arithmetic Package (Sheet 1 of 9)

```
COP CROSS ASSEMBLER    PAGE:    2
HXMATH    HEXADECIMAL ARITHMETIC

 53      0010   YDP     =       1,0
 54      002F   ZGUARD  =       2,15
 55      002E   ZMSD    =       2,14
 56      0027   ZLSD    =       2,7
 57      0021   ZSIGN   =       2,1
 58      0020   ZDP     =       2,0
 59      003F   FLAGS   =       3,15
 60      0030   OFLOW   =       3,0
 61             ;
 62      0000           .PAGE   0
 63 000 00              CLRA
 64 001 0F              LBI     0.0     ;CLEAR ALL THE RAM
 65 002 53              AISC    3
 66 003 12      RAMCLR: XABR
 67 004 81              JSRP    CLEAR
 68 005 12              XABR
 69 006 5F              AISC    15
 70 007 C9              JP      TESTG
 71 008 C3              JP      RAMCLR
 72             ;
 73             ;***************************************************************
 74             ;FOLLOWING CODE-TO NEXT LINE OF **-IS FOR CONTROL ONLY
 75             ;
 76 009 335F    TESTG:  OGI     15      ;PUT G LINES HIGH FOR READING G
 77             ;
 78                    ;USING G LINES FOR PRIMITIVE CONTROL TO SELECT ADD,SUB
 79                    ;MULTIPLY OR DIVIDE--WILL ENTER NUMBERS IN BREAKPOINT
 80                    ;MODE USING MODIFY COMMAND
 81                    ;
 82 00B 3301            SKGBZ   0
 83 00D D3              JP      TESTG1
 84 00E 3E              LBI     FLAGS
 85 00F 70              STII    0       ;RESET BIT 2 FOR ADD
 86 010 6840    JSRALN: JSR     ALIGN
 87 012 C9              JP      TESTG
 88 013 3311    TESTG1: SKGBZ   1
 89 015 D9              JP      TESTG2
 90 016 3E              LBI     FLAGS   ;SET SUBTRACT BIT
 91 017 74              STII    4       ;SET BIT 2 FOR SUBTRACT
 92 018 D0              JP      JSRALN
 93 019 3303    TESTG2: SKGBZ   2
 94 01B DF              JP      TESTG3
 95 01C 3E              LBI     FLAGS
 96 01D 70              STII    0       ;RESET BIT 3 FOR DIVIDE
 97 01E E4              JP      JSMD
 98 01F 3313    TESTG3: SKGBZ   3
 99 021 C9              JP      TESTG
100 022 3E              LBI     FLAGS
101 023 78              STII    8       ;SET BIT 3 FOR MULTIPLY
102 024 6940    JSMD:   JSR     MULDIV
103 026 C9              JP      TESTG
104             ;
```

FIGURE 5-8. Binary (Hexadecimal) Arithmetic Package (Sheet 2 of 9)

```
COP CROSS ASSEMBLER    PAGE:   3
HXMATH    HEXADECIMAL ARITHMETIC

                ;PRECEDING CODE IS FOR CONTROL ONLY,HAS NOTHING TO DO WITH THE
                :ARITHMETIC ALGORITHMS
                ;*************************************************************
                ;
0040                    .PAGE   1
                ; THIS IS THE ALIGN ROUTINE FOR ADD/SUBTRACT.  IT MAKES THE
                ; HEX POSITIONS OF THE TWO NUMBERS EQUAL BEFORE ADD OR
                ; SUBTRACT TAKES PLACE.  THE ROUTINE ASSUMES THAT THE NUMBERS
                ; ARE RIGHT JUSTIFIED ON ENTRY.  HEX POINT POSITION VALUES
                ; ARE RESTRICTED TO NUMBERS BETWEEN 0 - 8 (SINCE WE ARE ONLY
                ; DOING 8 DIGIT ROUTINES).  ROUTINE ONLY REQUIRED FOR FLOATING
                ; POINT ADD/SUBTRACT ALGORITHMS
                ;
040 0F  ALIGN:  LBI     XDP
041 15          LD      1
042 21          SKE     ;TEST DP0=DP1(DPX=DPY)
043 C6          JP      ALIGN2
044 6100        JMP     ADDSUB  ;IF EQUAL,PROCEED TO ADD/SUBTRACT
046 10  ALIGN2: CASC            ;TEST DP0 > DP1
047 D6          JP      DP0GT1  ;YES
048 0D  DP0LT1: LBI     XMSD    ;DP0<DP1.IF XMSD NOT 0,RIGHT SHIFT
049 00          CLRA            ;M1,ELSE LEFT SHIFT M0
04A 21          SKE
04B D1          JP      R1RSFT
04C 87  ROLSFT: JSRP    LSFTR0
04D 0F          LBI     XDP
04E 1F  DPPL1:  LBI     YDP
04F B4          JSRP    PLUS1   ;MODIFY DP AFTER SHIFT
050 C0          JP      ALIGN
051 83  R1RSFT: JSRP    RSFTR1
052 1F          LBI     YDP
053 0F  DPMIN1: LBI     XDP
054 AC          JSRP    MINUS1
055 C0          JP      ALIGN
056 1D  DP0GT1: LBI     YMSD    ;TESTING MSD OF M1 NOT 0
057 00          CLRA
058 21          SKE
059 DC          JP      RORSFT
05A 85  R1LSFT: JSRP    LSFTR1
05B C3          JP      DPPL1
05C 8D  RORSFT: JSRP    RSFTR0
05D D3          JP      DPMIN1
                ;
0080                    .PAGE   2
                ;THESE ARE THE BASIC REQUIRED SUBROUTINES FOR THE ARITHMETIC
                ;ROUTINES-COP420 AND LARGER CODE
                ;
080 0F  CLEAR0: LBI     0,0
081 00  CLEAR:  CLRA
082 04          XIS
```

FIGURE 5-8. Binary (Hexadecimal) Arithmetic Package (Sheet 3 of 9)

```
COP CROSS ASSEMBLER    PAGE:    4
HXMATH    HEXADECIMAL ARITHMETIC

155 083 81             JP      CLEAR
156 084 48             RET
157 085 3397   LSFTR1: LBI     YLSD
158 087 3387   LSFTR0: LBI     XLSD
159 089 00     LSFTX:  CLRA
160 08A 04     LSFT:   XIS
161 08B 8A             JP      LSFT
162 08C 48             RET
163 08D 0E     RSFTR0: LBI     0,15
164 08E 1E     RSFTR1: LBI     1,15
165 08F 00     RSFTRX: CLRA
166 090 07     RSFT:   XDS
167 091 23A2           XAD     SAVE1   ;SAVE VALUE TEMPORARILY
168 093 4E             CBA             ;ONLY WANT 8 DIGIT SHIFT
169 094 59             AISC    9
170 095 99             JP      DONE
171 096 2322           LDD     SAVE1   ;FETCH SAVED VALUE
172 098 90             JP      RSFT
173 099 2322   DONE:   LDD     SAVE1
174 09B 48             RET
175 09C 32     BINADD: RC
176 09D 15     BIN1:   LD      1       ;TWO REGISTER BINARY ADDITION
177 09E 30             ASC
178 09F 44             NOP
179 0A0 14             XIS     1
180 0A1 9D             JP      BIN1
181 0A2 48             RET
182 0A3 22     BINSUB: SC              ;TWO REGISTER BINARY SUBTRACTION
183 0A4 15     BINS1:  LD      1
184 0A5 10             CASC
185 0A6 44             NOP
186 0A7 14             XIS
187 0A8 A4             JP      BINS1
188 0A9 48             RET
189 0AA 2F     ZDPMN1: LBI     ZDP
190 0AB 3F     OFLMN1: LBI     OFLOW
191 0AC 05     MINUS1: LD              ;SUBTRACT 1 FROM MEMORY
192 0AD 5F             AISC    15
193 0AE 44             NOP
194 0AF 06     PLUS1A: X
195 0B0 48             RET
196 0B1 0F     XDPPL1: LBI     XDP
197 0B2 3F     OFLPL1: LBI     OFLOW
198 0B3 2F     ZDPPL1: LBI:    ZDP
199 0B4 05     PLUS1:  LD              ;ADD 1 TO MEMORY
200 0B5 51             AISC    1
201 0B6 AF             JP      PLUS1A  ;WILL SKIP IF GREATER THAN 15
202 0B7 06             X
203 0B8 49             RETSK
204 0B9 25     XFER2:  LD      2
205 0BA 24             XIS     2
206 0BB B9             JP      XFER2
207 0BC 48             RET
208            ;
```

FIGURE 5-8. Binary (Hexadecimal) Arithmetic Package (Sheet 4 of 9)

COP CROSS ASSEMBLER PAGE: 5
HXMATH HEXADECIMAL ARITHMETIC

```
209      0100            .PAGE   4
210                ;THIS IS THE ADD/SUBTRACT ROUTINE.  ROUTINE IS FOR 8 DIGITS.
211                ;HEXADECIMAL,FLOATING POINT, FULLY ALGEBRAIC.
212                ;
213 100 3E     ADDSUB: LBI     FLAGS
214 101 03             SKMBZ   2       ;TEST IF SHOULD SUBTRACT
215 102 D0             JP      CHNGMO  ;CHANGE SIGN RO(X) IF SUBTRACT
216 103 3381   ADSB1:  LBI     XSIGN   ;NOW TEST FOR SIGNS EQUAL
217 105 15             LD      1
218 106 21             SKE
219 107 D7             JP      SUB     ;NOT EQUAL,HENCE SUBTRACT
220 108 3387   ADD:    LBI     XLSD
221 10A 9C             JSRP    BINADD  ;R1+RO → R1,(Y+X → Y)
222 10B 1E     ERRCHK: LBI     YGUARD  ;TEST FOR OVERFLOW
223 10C 00             CLRA            ;IF 1,15(YGUARD) NOT 0,UNDERFLOW
224 10D 21             SKE
225 10E ED             JP      UNDRFL
226 10F 48             RET
227 110 3381   CHNGMO: LBI     XSIGN   ;CHANGE SIGN OF RO(X)
228 112 05             LD
229 113 58             AISC    8
230 114 44             NOP
231 115 06             X
232 116 C3             JP      ADSB1
233 117 3387   SUB:    LBI     XLSD
234 119 A3             JSRP    BINSUB  ;R1-RO → R1,(Y-X → Y)
235 11A 20             SKC             ;SEE IF MUST COMPLEMENT
236 11B DD             JP      COMPL
237 11C CB             JP      ERRCHK
238 11D 3397   COMPL:  LBI     YLSD    ;NEGATIVE RESULT,COMPLEMENT
239 11F 22             SC
240 120 00     COMPL1: CLRA
241 121 06             X
242 122 10             CASC
243 123 4A             ADT
244 124 04             XIS
245 125 E0             JP      COMPL1
246 126 3391           LBI     YSIGN   ;NOW CHANGE SIGN OF R1(Y)
247 128 05             LD
248 129 58             AISC    8
249 12A 44             NOP
250 12B 06             X
251 12C CB             JP      ERRCHK
252 12D 8E     UNDRFL: JSRP    RSFTR1  ;DO AN UNDERFLOW
253 12E 1F             LBI     YDP     ;ERROR IF YDP IS 0 WHEN UNDERFLOW
254 12F AC             JSRP    MINUS1
255 130 5F             AISC    15
256 131 F3             JP      ERROR
257 132 48             RET
258 133 1F     ERROR:  LBI     YDP
259 134 7F             STII    15      ;15 → YDP & YSIGN FOR ERROR
260 135 7F             STII    15
```

FIGURE 5-8. Binary (hexadecimal) Arithmetic Package (Sheet 5 of 9)

```
COP CROSS ASSEMBLER    PAGE:    6
HXMATH    HEXADECIMAL ARITHMETIC

261 136 48              RET

262     0140            .PAGE   5
263             ;MULTIPLY,DIVIDE ROUTINES. FLOATING POINT,8 DIGIT
264             ;
265 140 3387    MULDIV: LBI     XLSD
266 142 B9              JSRP    XFER2   ;M0 --> M2,X --> Z, THEN CLEAR X
267 143 25              LD      2       ;TRANSFER DP AND SIGN ALSO
268 144 24              XIS     2
269 145 25              LD      2
270 146 26              X       2
271 147 80              JSRP    CLEAR0  ;CLEAR M0
272 148 3E              LBI     FLAGS   ;NOW TEST IF MULTIPLY OR DIVIDE
273 149 13              SKMBZ   3
274 14A 61C0            JMP     MULPLY
275             ;
276 14C 22      DIVIDE  SC              ;M0/M1  -->  M0,(X/Y --> X)
277 14D 1F              LBI     YDP     ;DP2-DP0 --> DP2,(DPZ-DPX --> DPZ)
278 14E 35              LD      3
279 14F 10              CASC
280 150 44              NOP
281 151 06              X
282 152 3F              LBI     OFLOW   ;15 TO OFLOW DIGIT IF BORROW,ELSE 0
283 153 00              CLRA
284 154 20              SKC
285 155 40              COMP
286 156 06              X
287 157 3397    DIV1A:  LBI     YLSD
288 159 A3              JSRP    BINSUB  ;M0 - M1 TO M0,M1 SAVED
289 15A 20              SKC             ;PART OF THE REPEATED SUBTRACT FEATURE
290 15B E3              JP      DIV3A
291 15C 33A7    DIV3:   LBI     ZLSD    ;DIVIDE BY 0 CHECK
292 15E B4              JSRP    PLUS1
293 15F D7              JP      DIV1A   ;ALL OK,CONTINUE
294 160 6189            JMP     DIVBY0
295 162 3397    DIV3A:  LBI     YLSD
296 164 9C              JSRP    BINADD  ;RESTORE VALUE
297 165 0F              LBI     XDP
298 166 05              LD
299 167 57              AISC    7
300 168 617F            JMP     DIV1B   ;TESTING DP FOR FINISHED
301 16A 2E      DIV4:   LBI     ZGUARD
302 16B 00              CLRA
303 16C 21              SKE
304 16D 61E5            JMP     MDEND1
305 16F 3F              LBI     OFLOW
306 170 21              SKE             ;TEST OVERFLOW DIGIT
307 171 F8              JP              DIV4A
308 172 2F              LBI     ZDP     ;TEST DP2(ZDP) >= 9
309 173 05              LD
310 174 57              AISC    7
311 175 F8              JP      DIV4A
312 176 61E5            JMP     MDEND1
```

FIGURE 5-8. Binary (Hexadecimal) Arithmetic Package (Sheet 6 of 9)

```
COP CROSS ASSEMBLER     PAGE:    7
HXMATH     HEXADECIMAL ARITHMETIC

313 178 B3      DIV4A:  JSRP    ZDPPL1  ;DP2+1 --> DP2,(ZDP+1 --> ZDP)
314 179 6181            JMP     DIV1B2
315 17B B2              JSRP    OFLPL1  ;INCREMENT OVERFLOW DIGIT
316 17C 44              NOP             ;DEFEAT SKIP
317 17D 6181            JMP     DIV1B2
318 17F B1      DIV1B:  JSRP    XDPPL1  ;DP0 + 1 --> DP0
319 180 44              NOP
320 181 33A7    DIV1B2: LBI     ZLSD
321 183 89              JSRP    LSFTX
322 184 3387            LBI     XLSD
323 186 8A              JSRP    LSFT
324 187 6157            JMP     DIV1A
325 189 621B    DIVBY0: JMP     MDERR
326     01C0            .PAGE   7
327 1C0 32      MULPLY: RC              ;DP1+DP2 --> DP2,(DPY+DPZ --> DPZ)
328 1C1 1F              LBI     YDP
329 1C2 35              LD      3
330 1C3 30              ASC
331 1C4 44              NOP
332 1C5 06              X
333 1C6 00              CLRA            ;1 TO OFLOW IF CARRY,ELSE 0
334 1C7 20              SKC
335 1C8 CA              JP      MUL1A
336 1C9 51              AISC    1
337 1CA 3F      MUL1A:  LBI     OFLOW
338 1CB 06              X
339 1CC 33A7    MUL1:   LBI     ZLSD
340 1CE 05              LD
341 1CF 5F              AISC    15      ;LSD CONTROLLING REPEATED ADDS
342 1D0 D6              JP      MUL2
343 1D1 06              X
344 1D2 3397            LBI     YLSD    ;M0 + M1 --> M0,(X+Y --> X)
345 1D4 9C              JSRP    BINADD
346 1D5 CC              JP      MUL1
347 1D6 8D      MUL2:   JSRP    RSFTR0
348 1D7 2E              LBI     ZGUARD
349 1D8 90              JSRP    RSFT
350 1D9 B1              JSRP    XDPPL1
351 1DA 58              AISC    8
352 1DB CC              JP      MUL1    ;PRECEDING IS DP ADJUST
353 1DC 78      MUL3:   STII    8
354 1DD 3387            LBI     XLSD
355 1DF 00      MUL3X   CLRA            ;TEST M0=0(X=0)
356 1E0 21              SKE
357 1E1 6212            JMP     MUL5
358 1E3 04              XIS
359 1E4 DF              JP      MUL3X
360 1E5 2E      MDEND1: LBI     ZGUARD
361 1E6 00              CLRA
362 1E7 21              SKE
```

FIGURE 5-8. Binary (Hexadecimal) Arithmetic Package (Sheet 7 of 9)

COP CROSS ASSEMBLER PAGE: 8
HXMATH HEXADECIMAL ARITHMETIC

```
363 1E8 ED              JP      MDX
364 1E9 2F              LBI     ZDP
365 1EA 05              LD
366 1EB 57              AISC    7       ;TEST >= 9
367 1EC F7              JP      MDEND2
368 1ED 2E      MDX:    LBI     ZGUARD
369 1EE 8F              JSRP    RSFTRX
370 1EF 3386            LBI     ROUND   ;SAVE VALUE FOR ROUNDING
371 1F1 06              X
372 1F2 AA              JSRP    ZDPMN1
373 1F3 05              LD              ;TEST DP2(ZDP) = 15
374 1F4 51              AISC    1
375 1F5 F7              JP      MDEND2
376 1F6 AB              JSRP    OFLMN1  ;SUBTRACT 1 FROM OVERFLOW DIGIT
377 1F7 2F      MDEND2: LBI     2,0     ;TRANSFER R2 TO R0
378 1F8 B9              JSRP    XFER2
379 1F9 3391            LBI     YSIGN   ;ADD SIGNS AND PUT TO M0(X)
380 1FB 15              LD      1
381 1FC 31              ADD
382 1FD 06              X
383 1FE 3F              LBI     OFLOW   ;TEST OVERFLOW DIGIT
384 1FF 05              LD

385 200 51              AISC    1
386 201 C3              JP      MDEND4  ;NOT 15
387 202 DB              JP      MDERR   ;IS 15,NUMBER TOO BIG
388 203 00      MDEND4: CLRA            ;NOW TEST DIGIT > 0
389 204 21              SKE
390 205 80              JSRP    CLEAR0  ;IS NON ZERO,CLEAR M0
391 206 3387    MDRJ:   LBI     XLSD    ;RIGHT JUSTIFY THE RESULT
392 208 00              CLRA
393 209 21              SKE
394 20A 48              RET             ;IF LSD NON ZERO,STOP
395 20B 0F              LBI     XDP     ;IF DP PSN = 0,STOP
396 20C 05              LD
397 20D 5F              AISC    15
398 20E 48              RET
399 20F 06              X               ;ELSE,DECREMENT BY 1 AND CONTINUE
400 210 8D              JSRP    RSFTR0
401 211 C6              JP      MDRJ
402 212 2F      MUL5:   LBI     ZDP     ;TEST DP2(ZDP) = 0
403 213 00              CLRA
404 214 21              SKE
405 215 D8              JP      MUL3A
406 216 40              COMP            ;15 TO DP2(ZDP)
407 217 06              X
408 218 AC      MUL3A:  JSRP    MINUS1  ;DP2(ZDP) - 1 → DP2(ZDP)
409 219 61D6            JMP     MUL2
410 21B 0E      MDERR:  LBI     0,15
411 21C 7F              STII    15
412 21D 7F              STII    15
413 21E 48              RET
414                     .END
```

FIGURE 5-8. Binary (Hexadecimal) Arithmetic Package (Sheet 8 of 9)

```
COP CROSS ASSEMBLER     PAGE:    9
HXMATH     HEXADECIMAL ARITHMETIC

ADD     0108 *   ADDSUB 0100     ADSB1  0103     ALIGN  0040
ALIGN2 0046      BIN1   009D     BINADD 009C     BINS1  00A4
BINSUB 00A3      CHNGM0 0110     CLEAR  0081     CLEAR0 0080
COMPL  011D      COMPL1 0120     DIV1A  0157     DIV1B  017F
DIV1B2 0181      DIV3   015C *   DIV3A  1062     DIV4   016A *
DIV4A  0178      DIVBY0 0189     DIVIDE 014C *   DONE   0099
DP0GT1 0056      DP0LT1 0048 *   DPMIN1 0053     DPPL1  004E
ERRCHK 010B      ERROR  0133     FLAGS  003F     JSMD   0024
JSRALN 0010      LSFT   008A     LSFTR0 0087     LSFTR1 0085
LSFTX  0089      MDEND1 01E5     MDEND2 01F7     MDEND4 0203
MDERR  021B      MDRJ   0206     MDX    01ED     MINUS1 00AC
MUL1   01CC      MUL1A  01CA     MUL2   01D6     MUL3   01DC *
MUL3A  0218      MUL3X  01DF     MUL5   0212     MULDIV 0140
MULPLY 01C0      OFLMN1 00AB     OFLOW  0030     OFLPL1 00B2
PLUS1  00B4      PLUS1A 00AF     ROLSFT 004C *   RORSFT 005C
R1LSFT 005A *    R1RSFT 0051     RAMCLR 0003     ROUND  0006
RSFT   0090      RSFTR0 008D     RSFTR1 008E     RSFTRX 008F
SAVE1  0022      SUB    0117     TESTG  0009     TESTG1 0013
TESTG2 0019      TESTG3 001F     UNDRFL 012D     XDP    0000
XDPPL1 00B1      XFER2  00B9     XGUARD 000F *   XLSD   0007
XMSD   000E      XSIGN  0001     YDP    0010     YGUARD 001F
YLSD   0017      YMSD   001E     YSIGN  0011     ZDP    0020
ZDPMN1 00AA      ZDPPL1 00B3     ZGUARD 002F     ZLSD   0027
ZMSD   002E *    ZSIGN  0021 *
NO ERROR LINES
    355 ROM WORDS USED
COP     420     ASSEMBLY
SOURCE CHECKSUM = 7921
INPUT FILE     ABDUL10:HEXARITH.SRC     VN:     5
```

FIGURE 5-8. Binary (Hexadecimal) Arithmetic Package (Sheet 9 of 9)

5.2.9 Square Root

Two square root routines are provided: an integer square root and full floating-point square root. Both routines are based on the mathematical relationship:

$$\sum_{i=1}^{n} (2i - 1) = n^2$$

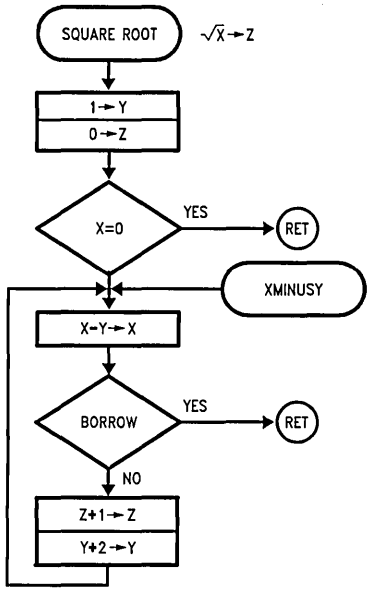
Therefore, if sequential odd numbers were subtracted from a value, the square root of that value is given by the number of odd numbers that must be subtracted from the original value to reduce that original value to "0" (or at least to reduce the integer part to 0).

Integer Square Root—BCD

A simple routine is provided that computes the integer portion of the square root of an integer. The technique is the simple subtraction of odd numbers as described above. The flow chart for this routine is given in *Figure 5-9*. The code and RAM map for a four-digit routine is given below.

	Bd 15	14	13	12
0	MSD	X		LSD
Br 1		Y		
2		Z		

The subroutines are assumed to be located in Page 2.



TL/DD/8800-22

FIGURE 5-9. Integer Square Root

```
SQROOT:   LBI   1,12      ; → Z
          STII  1         ;1 → Y,Bd+1 → Bd
          JSRP  CLEAR
          LBI   2,12      ;0 → Z
          JSRP  CLEAR
          LBI   0,12      ;test X = 0,if so exit
TSTZRO:   CLRA
          SKE
          JP    XMINUSY   ;X=0
          XIS
          JP    TSTZRO
          RET
XMINUSY:  JSRP  SUB       ;X-Y → X
          SKC             ;test borrow,C=0 if borrow
          RET             ;if borrow,exit-finished
          JSRP  ZPLUS1    ;Z+1 → Z
          JSRP  YPLUS1    ;Y+2 → Y
          JSRP  YPLUS1    ;
          JP    XMINUSY
```

The following subroutines, assumed to be in Page 2, are used by the square root routine above:

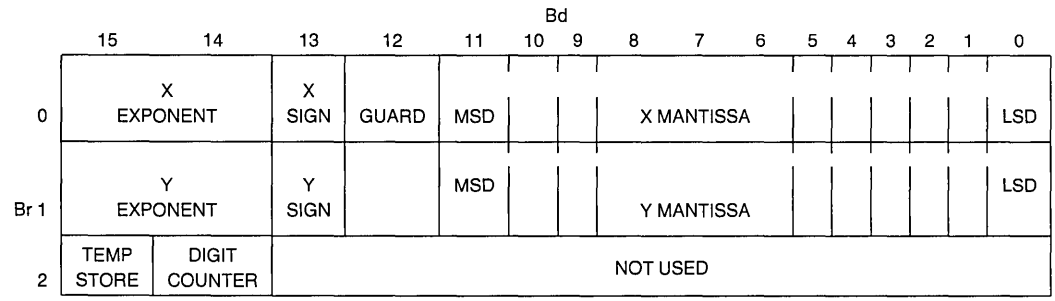
```
CLEAR:  CLRA          ; simple register clear
        XIS
        JP    CLEAR
        RET
SUB:    LBI   1,12
        LD    1       ; this is the basic BCD subtract routine as given
                      ; earlier
SUB1:   CASC
        ADT
        XIS   1
        JP    SUB1
        RET

ZPLUS:  LBI   2,12    ; this is the basic BCD increment routine as
                      ; given earlier, with the repeated LBI skip
                      ; feature
YPLUS:  LBI   1,12
        SC
PLUS1:  CLRA
        AISC  6
        ASC
        ADT
        XIS
        JP    PLUS1
        RET
```

Floating-Point Square Root—BCD

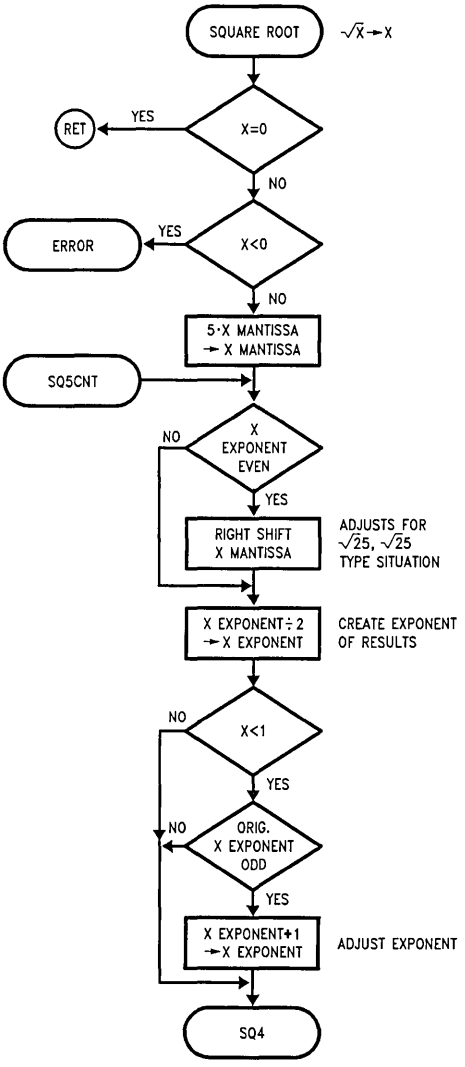
A full floating-point BCD square root routine is provided. As written, the routine works on a 12-digit number with a two-digit signed exponent. Although substantially more complex than the integer square root routine seen earlier, this routine has the same conceptual basis—the subtraction of odd numbers.

The first part of the routine creates the exponent of the result of dividing the original exponent by two. Note that this is accomplished by first multiplying the exponent by 5, via repeated additions, and then dividing it by 10 by means of a right-digit shift.

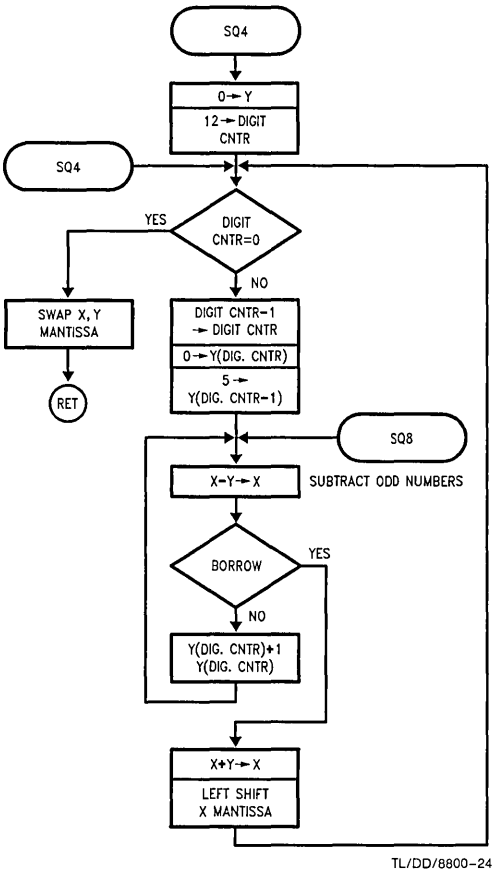
Two flow charts are provided, a generalized flow chart *(Figure 5-10)* and a detailed flow chart *(Figure 5-10a)*, to help clarify the routine. The RAM map for the routine is indicated below.



The routine performs $\sqrt{x} \rightarrow x$.



TL/DD/8800-23



TL/DD/8800-24

FIGURE 5-10. Square Root—General Flow Chart

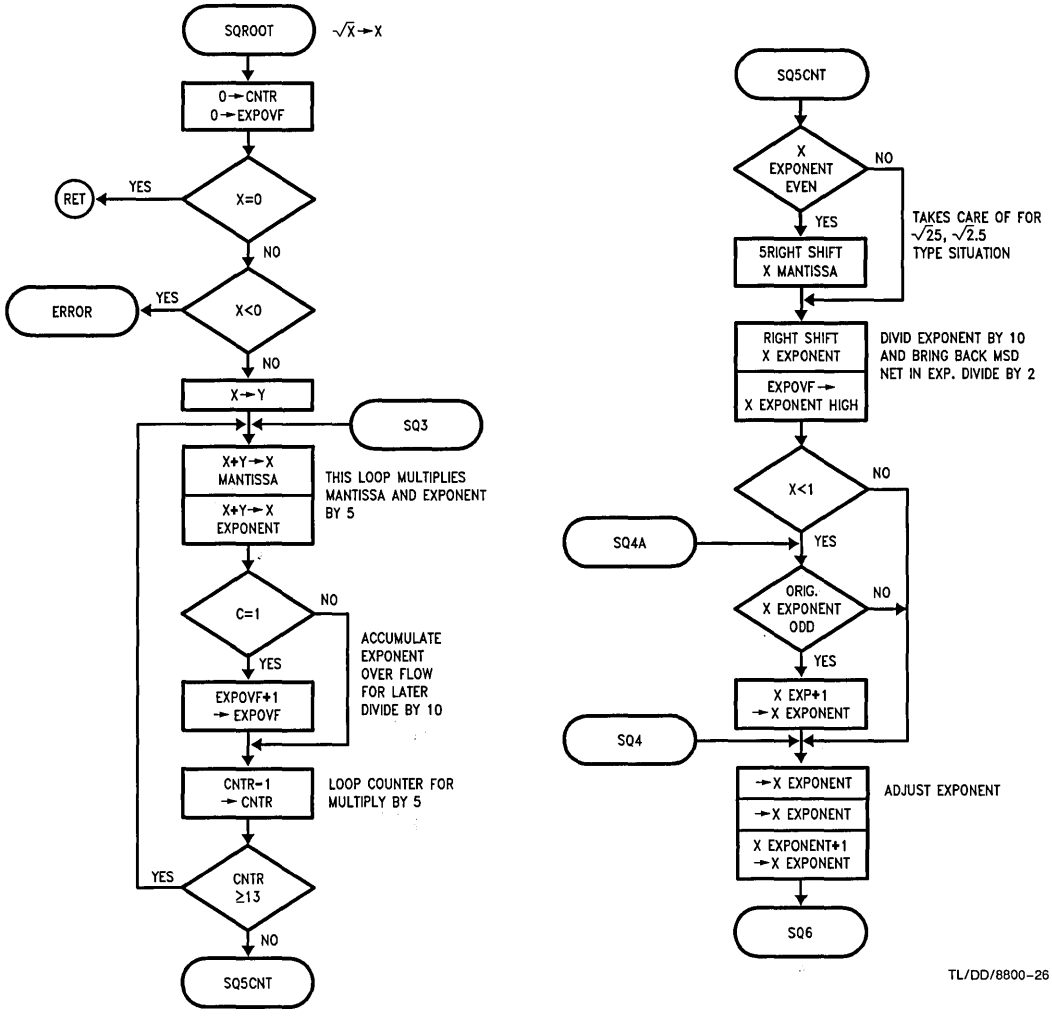
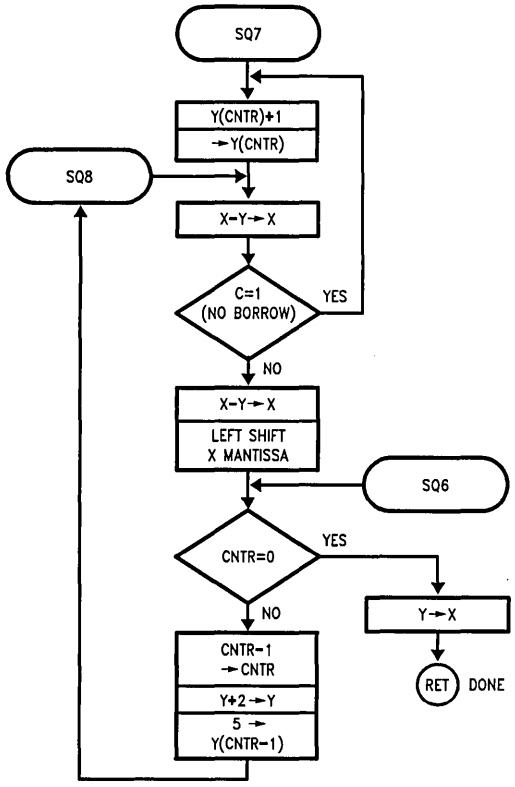


FIGURE 5-10a. Square Root—Detailed Flow Chart (Sheet 1 of 2)



TL/DD/8800-27

FIGURE 5-10a. Square Root—Detailed Flow Chart (Sheet 2 of 2)

```
COP CROSS ASSEMBLER     PAGE:    1
SQROOT     SQUARE ROOT ROUTINE

 1                ;THIS PROGRAM IS A FLOATING POINT SQUARE ROOT ROUTINE. THE
 2                ;ROUTINE ASSUMES THAT THE NUMBER X(REGISTER 0) IS IN
 3                ;SCIENTIFIC NOTATION FORMAT, I.E., SIGNED EXPONENT AND
 4                ;MANTISSA.  AS WRITTEN THE ROUTINE ASSUMES A 12 DIGIT BCD
 5                ;MANTISSA AND GENERATES A 12 DIGIT BCD RESULT.  THE
 6                ;EXPONENT IS APPROPRIATELY HANDLED.  BY CHANGING ONLY THE
 7                ;START VALUE IN THE DIGIT COUNTER, SMALLER MANTISSAS CAN
 8                ;BE EASILY HANDLED. THE STRUCTURE OF THE ROUTINE DOES NOT
 9                ;CHANGE.
10                ;
11                ;THE ROUTINE ASSUMES THAT THE LSD OF THE NUMBER IS LOCATION
12                ;0. THE MSD OF THE NUMBER IS IN LOCATION 11.  LOCATION 13
13                ;CONTAINS THE SIGN INFORMATION FOR BOTH THE EXPONENT AND THE
14                ;MANTISSA.  BIT 0 OF LOCATION 13 IS THE EXPONENT SIGN; BIT
15                ;3 IS THE MANTISSA SIGN; BITS 1 AND 2 ARE NOT USED.  A TWO
16                ;DIGIT EXPONENT IS CONTAINED IN LOCATIONS 14 AND 15 WITH
17                ;LOCATION 15 BEING THE MOST SIGNIFICANT DIGIT OF THE EXPONENT.
18                ;LOCATION 12 IS THE MANTISSA GUARD DIGIT AND IS USED IN THE
19                ;COMPUTATION BUT CONTAINS NO INFORMATION ON ENTRY TO
20                ;OR EXIT FROM THE ROUTINE.
21                ;
22                ;THE ROUTINE FURTHER ASSUMES THAT THE DECIMAL POINT IS LOCATED
23                ;TO THE RIGHT OF THE MSD OF THE MANTISSA,I.E.,ALL NUMBERS ARE
24                ;OF THE FORM 1.2345 x 10**EXPONENT.
25                ;
26      0000      XLSD    =       0,0
27      000B      XMSD    =       0,11
28      000C      XGUARD  =       0,12
29      000D      XSIGN   =       0,13
30      000E      XEXPLO  =       0,14
31      000F      XEXPHI  =       0,15
32      0010      YLSD    =       1,0
33      001C      YGUARD  =       1,12
34      001E      YEXPLO  =       1,14
35      001F      YEXPHI  =       1,15
36      002E      CNTR    =       2,14    ;THIS IS MANTISSA DIGIT COUNTER
37      002F      EXPOVF  =       2,15    ;EXPONENT OVERFLOW DIGIT
38                ;
39                        .TITLE  SQROOT,'SQUARE ROOT ROUTINE'
40                ;
41      0000              .PAGE   0
42                ;************************************************************
43                ;CODE FROM HERE TO NEXT LINE OF *'S IS NOT PART OF SQUARE ROOT
44                ;ROUTINE.  IT IS FOR TEST ONLY
45                ;
46 000 00         START;  CLRA
47 001 12         STRT1:  XABR
48 002 00         STRT2:  CLRA
49 003 04                 XIS             ;CLEAR ALL THE RAM FOR CONTROL
50 004 C2                 JP      STRT2
```

FIGURE 5-11. Square Root Routine (Sheet 1 of 6)

```
COP CROSS ASSEMBLER     PAGE:    2
SQROOT    SQUARE ROOT ROUTINE

 51 005 12              XABR
 52 006 5D              AISC    13
 53 007 C1              JP      STRT1
 54             TESTSQROOT:
 55 008 44              NOP
 56 009 44              NOP
 57 00A 335F            OGI     15      ;THIS JUST FOR TEST,DETECTING ERROR
 58 00C 6900            JSR     SQROOT  ;RETURN AND SKIP IF ERROR,SO G WILL BE
 59 00E 3350            OGI     0       ;SET TO 0 IF NO ERROR
 60 010 44              NOP
 61 011 44              NOP
 62 012 C8              JP      TESTSQROOT
 63             ;
 64             ;NOTE THAT THE PRECEDING CODE IS NOT PART OF THE SQUARE ROOT
 65             ;ROUTINE.  IT IS FOR CONTROL AND TEST ONLY
 66             ;*********************************************************
 67             ;
 68             ;THE FOLLOWING CODE IS PART OF SQUARE ROOT ROUTINE.THE SUB-
 69             ;ROUTINES ARE INCLUDED. IN A SYSTEM REQUIRING SQUARE ROOT, IT
 70             ;IS HIGHLY PROBABLE THAT AT LEAST SOME OF THE OTHER BASIC
 71             ;ARITHMETIC FUNCTIONS WOULD ALSO BE REQUIRED. THE SUBROUTINES
 72             ;BELOW WOULD ALSO BE USABLE IN THOSE ROUTINES.
 73             ;
 74             ;*********************************************************
 75             ;
 76     0080            .PAGE   2
 77 080 00      CLEAR:  CLRA            ;CLEAR A REGISTER
 78 081 04              XIS
 79 082 80              JP      CLEAR
 80 083 48              RET
 81 084 1F      ADDXY:  LBI     YLSD    ;X + Y --> X,13 DIGITS(MANTISSA AND
 82 085 32      ADDXYE: RC              ;GUARD DIGIT)
 83 086 15      ADLOOP: LD      1
 84 087 56              AISC    6       ;DECIMAL ADJUST
 85 088 30              ASC
 86 089 4A              ADT             ;DECIMAL CORRECT
 87 08A 14              XIS     1
 88 08B 8D              JP      ADLP2
 89 08C 48              RET
 90 08D 4E      ADLP2:  CBA             ;NOW TEST IF DONE
 91 08E 53              AISC    3       ;IF BD >= 13,DONE
 92 08F 86              JP      ADLOOP
 93 090 48              RET
 94             ;
 95 091 05      PLUS1:  LD              ;MEMORY LOCATION PLUS 1
 96 092 51              AISC    1
 97 093 44              NOP
 98 094 06      XRET:   X
 99 095 48              RET
100 096 2D      CTRMN1: LBI     CNTR    ;DIGIT COUNTER MINUS 1
101 097 05      MINUS1: LD              ;MEMORY LOCATION MINUS 1
102 098 5F              AISC    15
```

FIGURE 5-11. Square Root Routine (Sheet 2 of 6)

```
COP CROSS ASSEMBLER     PAGE:    3
SQROOT     SQUARE ROOT ROUTINE

103 099 94              JP      XRET
104 09A 94              JP      XRET
105             ;
106 09B 22      SUBXY:  SC
107 09C 1F              LBI     YLSD
108 09D 15      SUBLOOP:LD      1       ;X - Y --> X,13 DIGITS (MANTISSA AND
109 09E 10              CASC            ;GUARD DIGIT)
110 09F 4A              ADT
111 0A0 14              XIS     1
112 0A1 4E              CBA
113 0A2 53              AISC    3
114 0A3 9D              JP      SUBLOOP
115 0A4 48              RET
116             ;
117 0A5 0B      RSHX:   LBI     XGUARD  ;RIGHT SHIFT X MANTISSA
118 0A6 00              CLRA
119 0A7 07      RSHX1:  XDS
120 0A8 A7              JP      RSHX1
121 0A9 48              RET
122             ;
123 0AA 0F      LSHX:   LBI     XLSD    ;LEFT SHIFT X
124 0AB 00              CLRA
125 0AC 04      LSHX1:  XIS
126 0AD AC              JP      LSHX1
127 0AE 48              RET
128             ;
129 0AF 15      XFER1:  LD      1       ;REGISTER TRANSFER
130 0B0 14              XIS     1
131 0B1 AF              JP      XFER1
132 0B2 48              RET
133             ;
134     0100            .PAGE   4
135             ;
136             ;*************************************************************
137             ;
138             ;THE FOLLOWING IS THE BODY OF THE SQUARE ROOT ROUTINE
139             ;X MUST BE NORMALIZED ON ENTRY, I.E.,NO LEADING ZEROES
140             ;THUS IF MSD IS 0,THE MANTISSA IS 0
141             ;
142 100 2D      SQROOT: LBI     CNTR
143 101 80              JSRP    CLEAR   ;CLEAR DIGIT COUNTER AND EXPONENT
144                                     ;OVERFLOW DIGIT-CNTR & EXPOVF
145 102 0A      SQ1:    LBI     XMSD    ;TEST FOR X = 0,IF YES,RETURN
146 103 05              LD              ;IF XMSD 0,X IS 0
147 104 5F              AISC    15
148 105 48              RET
149 106 0C      SQ2:    LBI     XSIGN   ;ERROR IF X IS NEGATIVE
150 107 13              SKMBZ   3
151 108 49      ERROR:  RETSK           ;RETURN AND SKIP FOR ERROR
152 109 0F              LBI     X;XLSD
153 10A AF              JSRP    XFER1   ;X --> Y FOR SUBSEQUENT ADDS
154             ;
```

FIGURE 5-11. Square Root Routine (Sheet 3 of 6)

```
COP CROSS ASSEMBLER    PAGE:    4
SQROOT    SQUARE ROOT ROUTINE

155 10B 84     SQ3:    JSRP    ADDXY    ;THIS LOOP MULTIPLIES X MANTISSA BY
156 10C 1D             LBI     YEXPLO   ;5 AND THE X EXPONENT BY 5
157 10D 85             JSRP    ADDXYE
158 10E 86             JSRP    ADLOOP
159 10F 20             SKC
160 110 D3             JP      TSTCTR
161 111 2E             LBI     EXPOVF   ;EXTRA DIGIT FOR THE EXPONENT MULTIPLY
162 112 91             JSRP    PLUS1
163 113 96     TSTCTR: JSRP    CTRMN1   ;CNTR IS USED AS LOOP COUNTER HERE
164 114 05             LD
165 115 53             AISC    3        ;IF CNTR IS < 13,MULTIPLY IS COMPLETE
166 116 D8             JP      SQ5CNT
167 117 CB             JP      SQ3
168            ;
169 118 0D     SQ5CNT  LBI     XEXPLO   ;TEST X EXPONENT EVEN(IF ORIGINAL X
170 119 05             LD               ;EXPONENT EVEN,5 TIMES IT WILL RESULT
171 11A 5F             AISC    15       ;IN XEXPLO BEING 0)
172 11B A5             JSRP    RSHX     ;FOR SQRT 25,SQRT 2.5 TYPE CASE
173            ;
174 11C 2E             LBI     EXPOVF   ;RIGHT SHIFT X EXP WITH 0'FLOW DIGIT
175 11D 25             LD      2        ;RESULTS IN NET EXPONENT DIVIDE BY 2
176 11E 07             XDS              ;WHICH IS DESIRED RESULT FOR SQUARE
177 11F 07             XDS              ;ROOT
178 120 01             SKMBZ   0        ;SEE IF X < 1
179 121 E9             JP      SQ4A     ;YES
180 122 1F     SQ4:    LBI     YLSD     ;CLEAR Y,WILL CREATE ANSWER IN Y
181 123 80             JSRP    CLEAR
182 124 0C             LBI     XSIGN    ;MOVE SIGN,EXPONENT TO Y
183 125 AF             JSRP    XFER1
184 126 2D             LBI     CNTR     ;LOAD DIGIT COUNTER WITH NUMBER
185 127 7C             STII    12       ;OF DIGITS
186 128 FC             JP      SQ6
187 129 0D     SQ4A:   LBI     XEXPLO   ;TEST ORIGINAL EXPONENT ODD,5 TIMES IT
188 12A 5B             AISC    11       ;RESULTS IN A, AT THIS POINT,=5 IF
189 12B E2             JP      SQ4      ;ORIGINAL X EXPONENT ODD
190 12C 91             JSRP    PLUS1    ;ORIGINAL EXPONENT WAS ODD * X<1,
191 12D 57             AISC    7        ;CORRECT EXPONENT BY ADDING 1
192 12E E2             JP      SQ4      ;IF LSD EXPONENT WAS < 9,STOP
193 12F 70             STII    0        ;WAS = 9,SO SET TO 0 AND INCREMENT
194 130 91             JSRP    PLUS1    ;MSD OF EXPONENT
195 131 E2             JP      SQ4
196            ;
197            ;EXPONENT COMPUTATION COMPLETE AT THIS POINT.THE REST OF THE
198            ;CODE COMPUTES THE MANTISSA OF THE RESULT BY THE TECHNIQUE
199            ;OF SUBTRACTION OF ODD NUMBERS. SINCE THE MANTISSA HAS BEEN
200            ;MULTIPLIED BY 5, 5 TIMES THE VARIOUS ODD NUMBERS WILL BE SUB-
201            ;TRACTED. THUS TO SUBTRACT 1,2,5,7,... FROM THE ORIGINAL WE
202            ;SUBTRACT 5,15,25,35,...FROM 5 TIMES THE ORIGINAL VALUE.
203            ;
204 132 2D     SQ7:    LBI     CNTR
205 133 35             LD      3        ;INCREMENT Y(CNTR)
206 134 50             CAB
207 135 91             JSRP    PLUS1
208 136 9B     SQ8:    JSRP    SUBXY    ;THIS IS THE REPEATED SUBTRACT
```

FIGURE 5-11. Square Root Routine (Sheet 4 of 6)

```
COP CROSS ASSEMBLER     PAGE:    5
SQROOT     SQUARE ROOT ROUTINE

209 137 20              SKC             ;IF WE BORROW,NEED TO SHIFT
210 138 FA              JP      SQ8A
211 139 F2              JP      SQ7
212 13A 84      SQ8A:   JSRP    ADDXY   ;RESTORE VALUE
213 13B AA              JSRP    LSHX
214 13C 96      SQ6:    JSRP    CTRMN1  ;DECREMENT DIGIT COUNTER
215 13D 05              LD
216 13E 51              AISC    1       ;SEE IF IT WAS 0 FOR EXIT
217 13F C3              JP      SQ6A
218 140 1F      DONE:   LBI     YLSD    ;DONE,TRANSFER RESULT IN Y TO X
219 141 AF              JSRP    XFER1
220 142 48              RET
221 143 35      SQ6A:   LD      3       ;0 --> Y(CNTR)
222 144 50              CAB
223 145 00              CLRA
224 146 07              XDS
225 147 75              STII    5       ;5 --> Y(CNTR - 1)
226 148 6136            JMP     SQ8
227             ;
228                     .END
```

FIGURE 5-11. Square Root Routine (Sheet 5 of 6)

```
COP CROSS ASSEMBLER     PAGE:    6
SQROOT     SQUARE ROOT ROUTINE

ADDXY   0084      ADDXYE 0085      ADLOOP 0086      ADLP2   008D
CLEAR   0080      CNTR   002E      CTRMN1 0096      DONE    0140 *
ERROR   0108 *    EXPOVF 002F      LSHX   00AA      LSHX1   00AC
MINUS1  0097 *    PLUS1  0091      RSHX   00A5      RSHX1   00A7
SQ1     0102      SQ2    0106 *    SQ3    010B      SQ4     0122
SQ4A    0129      SQ5CNT 0118      SQ6    013C      SQ6A    0143
SQ7     0132      SQ8    0136      SQ8A   013A      SQROOT  0100
START   0000 *    STRT1  0001      STRT2  0002      SUBLOO  009D
SUBXY   009B      TESTSQ 0008      TSTCTR 0113      XEXPHI  000F *
XEXPLO  000E      XFER1  00AF      XGUARD 000C      XLSD    0000
XMSD    000B      XRET   0094      XSIGN  000D      YEXPHI  001F *
YEXPLO  001E      YGUARD 001C *    YLSD   0010

NO ERROR LINES
    144     ROM WORDS USED
COP     420     ASSEMBLY
SOURCE CHECKSUM = 46ED
INPUT FILE      ABDUL10:SQROOT.SRC      VN:     9
```

FIGURE 5-11. Square Root Routine (Sheet 6 of 6)

5.2.10 Binary to BCD Conversion

Several methods of performing a binary to BCD conversion are illustrated. These different approaches illustrate algorithms and different programming techniques.

A Simple 8-bit Binary to BCD Routine

This is a simple routine for converting an eight-bit binary number to its three-digit BCD equivalent. The conversion is the straightforward scheme of adding the respective powers of two. However, this is reduced if the eight-bit number is treated as a two-digit hex number: then we merely expand the number by the powers of 16. Thus we have:

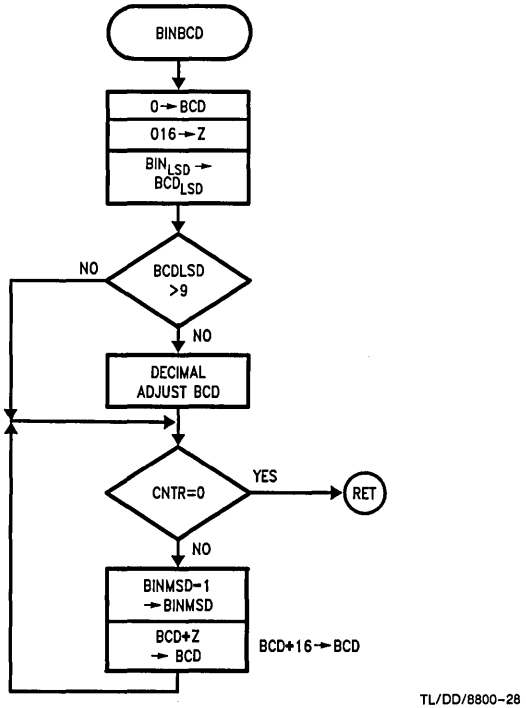
$$1110\ 0111_2 = E7_{16} = 14_{10} * 16^1_{10} + 7_{10} * 16^0_{10} = 231_{10}$$

$$0101\ 1111_2 = 5F_{16} = 5_{10} * 16^0_{10} + 15_{10} * 16^0_{10} = 95_{10}$$

The flow chart for this routine is given in *Figure 5-12*. The RAM map is given below.

	Bd 15	14	13
2	Work Space - Z		
Br 1	MSD	BCD Result	LSD
0	Binary Number		Not Used

The routine converts the binary number in R0 to the BCD number in R1. R2 is used as work space. The binary value is destroyed. Four implementations of this routine are presented.



TL/DD/8800-28

FIGURE 5-12. Eight-Bit Binary to BCD Conversion

Version I

```
BINBCD:  LBI   0,14
         XDS   1       ;BIN_LSD → BCD_LSD,0 to other digits in BCD
         XIS
         STII  0
         STII  0
         LBI   2,13    ;016 → Z
         STII  6
         STII  1
         STII  0
         LBI   1,13    ;test BCD_LSD > 9, if so, decimal adjust
         LD
         AISC  6
         JP    TEST
         XIS
         STII  1
TEST:    LBI   0,15    ;test BIN_MSD = 0, if yes exit
         LD            ;conversion complete
         AISC  15
         RET           ;decrement BIN_MSD by 1
         X
         LBI   2,13
         RC
LOOP:    LD    3       ;add BCD+16 (BCD+Z) → BCD
         AISC  6
         ASC
         ADT
         XIS   3
         JP    LOOP
         JP    TEST
```

Version II

```
RIBCD:   LBI   0,14
         XDS   1       ; BIN_LSD → BCD_LSD, 0 to other digits in BCD
         XIS
         STII  0
         STII  0
         LBI   2,13    ; equivalent of 016 → Z
         STII  12      ; have incorporated AISC 6 into constant for subsequent BCD addition
         STII  7
         STII  6
         LBI   0,13    ; test BCD_LSD → 9, if so decimal adjust
         LD
         AISC  6
         JP    TEST
         XIS
         STII  1
TEST:    LBI   0,15    ; test BIN_MSD = 0, if yes exit - conversion complete
         LD
         AISC  15
         RET
         X             ; decrement BIN_MSD by 1
         LBI   2,13
         RC
LOOP:    LD    3       ;BCD+16 → BCD
         ASC
         ADT
         XIS   3
         JP    LOOP
         JP    TEST
```

```
Version III
BINBCD:  LBI   0,14  ; BIN_LSD → BCD_LSD, 0 to other digits in BCD
         XDS   1
         XIS
         STII  0
         STII  0
         LBI   0,13  ; test BCD_LSD > 9, if so decimal adjust the number
         LD
         AISC  6
         JP    TEST
         XIS
         STII  1
TEST:    LBI   0,15  ; test BIN_MSD = 0, if yes exit
         LD          ; conversion complete
         AISC  15    ; else decrement BIN_MSD by 1
         RET
         X
         LBI   2,13  ; straight line BCD+16 → BCD using no additional RAM
         RC
         CLRA
         AISC  12
         ASC
         ADT
         XIS
         CLRA
         AISC  7
         ASC
         ADT
         XIS
         CLRA
         AISC  6
         ASC
         ADT
         X
         JP    TEST  ; loop back to TEST
Version IV
BINBCD:  LBI   0,14
         XDS   1     ; BIN_LSD → BCD_LSD, 0 to other digits in BCD
         XIS
         STII  0
         STII  0
         LBI   2,13
         STII  0     ; clear Z
         STII  0
         STII  0
LOOP:    JSRP  BCDADD; decimal adjust first time, add 16 in all subsequent times
         LBI   2,13
         STII  6
         STII  1
TEST:    LBI   0,15
         LD
         AISC  15
         RET
         X
         JR    LOOP
```

The routine uses the following subroutine, assummed to be located in Page 2.

```
BCDADD:  LBI   2,13
         RC
ADLOOP:  LD    3
         AISC  6
         ASC
         ADT
         XIS   3
         JP    ADLOOP
         RET
```

Note: By using the same kind of "trick" as was illustrated in Version II, the total ROM count can be reduced by one word and the execution speed improved.

Let us now consider these four programs. They all do precisely the same thing: convert an eight-bit binary number to a three-digit BCD number using the same algorithm. The differences are in implementation only. Version I takes 29 ROM words, uses 8 RAM digits (two for input binary number, three for BCD result, and three for scratch pad), has a worst case execution time of 409 instruction cycles, and uses no subroutines. Version II takes 28 ROM words, also uses 8 RAM digits, has a worst case execution time of 364 instruction cycles, and also uses no subroutines. Version III takes 34 ROM words, uses only 5 RAM digits, has a worst case execution time of 360 instruction cycles, and uses no subroutines. Version IV uses 28 ROM words, including 9 words in a subroutine; uses 8 RAM digits; and has a worst case execution time of 474 instruction cycles. Other variations on these routines are possible which will affect ROM, RAM, and execution time.

Version I is the straightforward implementation of the flow chart with few tricks. It is fairly representative of the amount of code required for the task; uses the maximum RAM for the function and is about midrange in execution speed. Version II makes a very slight change to Version I. It sets up a constant with a decimal adjust factor built in. The result is that Version II uses one less ROM word and the same amount of RAM as Version 1; however, Version II executes considerably faster, about a 10 percent speed improvement. Version III uses the minimum RAM for the function, uses the most ROM, and has the fastest execution time. This has been achieved by straight line coding the BCD add 16. This both maximizes speed and reduces RAM usage but the penalty is ROM code. RAM usage is reduced by storing the constant "16" in ROM rather than RAM. Version IV is preferable in cases where a BCD addition subroutine already exists in the program. Not counting the subroutine Version IV uses only 19 ROM words. However, Version IV has the slowest execution time. By the addition of two ROM words, as shown in Version IVA, the speed of Version IV can be significantly improved. Version IVA is the same as Version IV but achieves faster speed, by moving some code out of the main loop, with a small ROM penalty.

Version IVA

```
BINBCD   LBI   0,14     ; BIN_LSD → BCD_LSD, 0 to other
         XDS   1        ; digits in BCD
         XIS
         STII  0
         STII  0
         LBI   2,13     ; clear Z
         STII  0
         STII  0
         STII  0
         JSRP  BCDADD   ; decimal adjust BCD
         LBI   2,13     ; 16 → Z
         STII  6
         STII  1
         JP    TEST
LOOP:    JSRP  BCDADD   ; basic loop
TEST:    LBI   0,15
         LD
         AISC  15
         RET
         X
         JP    LOOP
```

Version IVA uses 21 ROM words (not counting the subroutine), uses 8 RAM digits, and executes in 454 instruction cycle times.

Binary to BCD Conversion—Doubling Methods

If we have a binary number expressed as $b_n b_{n-1} \ldots b_2 b_1 b_0$ where bx is either 1 or 0, the standard expansion to produce the decimal (BCD) number is as follows:

$$b_n{*}2^n + b_{n-1}{*}2^{n-1} + \ldots + b_2{*}2^2 + b_1{*}2 + b_0 = \sum_{i=0}^{n} b_i{*}2^i$$

For simplicity, a six-bit binary number is used.

$b_5 b_4 b_3 b_2 b_1 b_0$

The expansion for this number for its decimal equivalent, is then

$b_5{*}2^5 + b_4{\cdot}\,2^4 + b_3{\cdot}2^3 + b_2{\cdot}2^2 + b_1{\cdot}2 + b_0$

This can be rewritten as

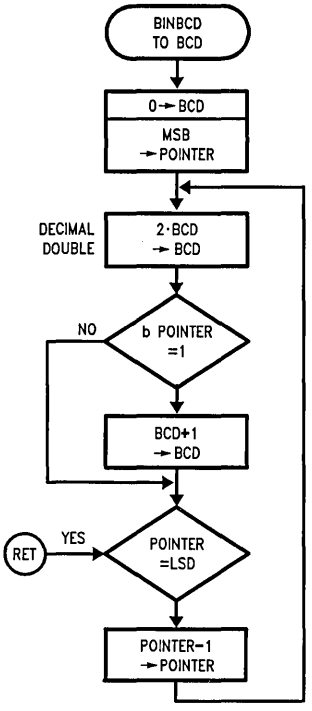
$2\,\{2\,[2 < 2\,(2\,(b_5\,(+b_4) + b_3 > + b_2] + b_1\} + b_0$

This expression, although apparently more complex, points out one means of conversion that is easy to implement because it is iterative. The first step is to set the BCD number equal to the most significant bit, here it is b_5. Then the value is doubled and one is added if the next bit is one. The value is then doubled again and one is added if the next bit is one. The cycle continues until the LSB is added to the result. *Figure 5-13* is the flow chart for this general approach.

The Straightforward Implementation

This implementation is the straightforward implementation of the flow chart of *Figure 5-13.* As written, it converts a 16-bit binary number to its five-digit BCD counterpart. The routine expands by merely changing the pertinent LBI instructions. *Figure 5-14* is the RAM map for this routine. The routine uses one subroutine level.

The routine uses the following subroutines assumed to be located in Page 2.



TL/DD/8800-29

FIGURE 5-13. Binary to BCD Conversion—Basic Doubling Algorithm

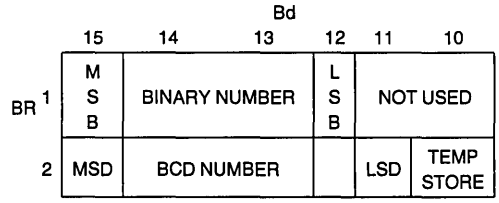


FIGURE 5-14. RAM Map for Doubling Algorithm Straight-Forward Implementation

```
DOUBLE:  LBI   2,11     ; decimal double of BCD number
DBLA:    LD            ; modify the LBI for larger numbers
         AISC  6
         ASC
         ADT
         XIS           ; sips on Bd increment past 15
         JP    DBLA
SETUP:   RC            ; falls into SETUP routine-which
         LBI   2,10    ; is also called independently
         LD    3       ; fetch digit position of binary number
         CAB           ; and load Bd with it. Br adjusted by
         RET           ; LD 3 instruction
```

The basic routine is as follows:

```
BINBCD:  LBI   2,10    ; load pointer with position of binary MSB
         STII  15
CLEAR:   CLRA          ; 0 → BCD
         XIS
         JP    CLEAR
BINDEC:  JSRP  SETUP   ; point to proper digit in binary number
         SKMBZ 3       ;
         SC            ; march down the digit;set C
         JSRP  DOUBLE  ; if addressed bit is 1. Then do
         SKMBZ 2       ; double to get 2* BCD + b_x → BCD
         SC            ; (C is reset on exit from SETUP and DOUBLE)
         JSRP  DOUBLE
         SKMBZ 1
         SC
         JSRP  DOUBLE
         SKMBZ 0
         SC
         JSRP  DOUBLE
         LBI   2,10    ; test for done, here finished
         LD
         AISC  3       ; if Bd ≤ 12
         RET
         AISC  12      ; else, decrement Bd by 1 and
         X
         JP    BINDEC  ; continue
```

This routine used 25 words plus 12 words in the subroutine page for a total of 37 words. The routine executes in 690 instruction cycles. The execution time is data independent. The routine preserves the binary number. The routine uses 10 RAM digits; 4 for the original number, 5 for the BCD result, and 1 scratch pad.

Variation I—The Doubling Algorithm—"Shift 1"

The straightforward implementation can be modified in a simple way by using some left bit shifting on the binary number. The basic flow chart is the same but a detailed modified flow chart is shown in *Figure 5-15*. *Figure 5-16* is the RAM map for this variation.

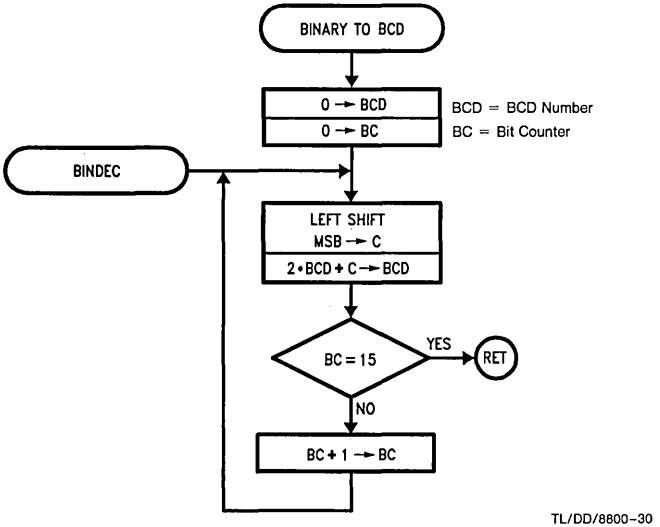


FIGURE 5-15. Flow Chart for Variation 1

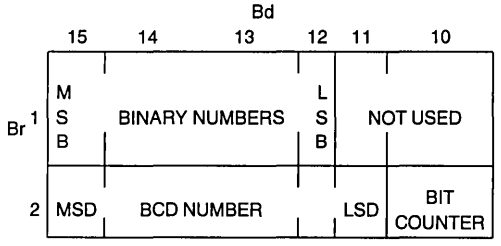


FIGURE 5-16. RAM Map for Variation 1 on the Doubling Algorithm

The code for this implementation is as follows:

```
BINBCD:   LBI   2,10
CLEAR:    CLRA          ; 0 → bit counter, BCD number
          XIS
          JP    CLEAR
BINDEC:   LBI   1,12    ; left-shift binary number one
          RC            ; bit with MSB going into C
LOOP1:    LD
          ASC           ; left-shift by means of binary double
          NOP
          XIS
          JP    LOOP1
DCDBL     LBI   2,11    ; double BCD number
LOOP2:    LD
          AISC  6
          ASC
          ADT
          XIS
          JP    LOOP2
TEST      LBI   2,10    ; test if finished
          LD
          AISC  1       ; done if bit counter = 15
          JP    X1
          RET
X1:       X
          JP    BINDEC
```

This routine uses no subroutines, takes 25 ROM words, and uses 10 digits of RAM—just as the first method. The existence of a CLEAR subroutine and/or a decimal double and/or a binary double routine in the program would further reduce the code required for this routine. As written, the routine does not preserve the binary number. The routine executes in 954 instruction cycles. Thus, it uses code, overall, than the previous routine but executes substantially slower.

Variation 2—The Shifting Algorithm

The left bit shift shown in Section 5.2.9 is simply a binary double. The primary algorithm still requires the add one to the doubled BCD number when the binary bit is a one. This routine does the doubling a little differently: A binary double is performed on the BCD number with a subsequent decimal correct. The RAM map for this scheme is given in *Figure 5-17.* The flow chart is in *Figure 5-18.* The flow chart is general for the algorithm. Judicious placement of data in RAM eliminates the need for the digit counter, DC, shown in the flow chart.

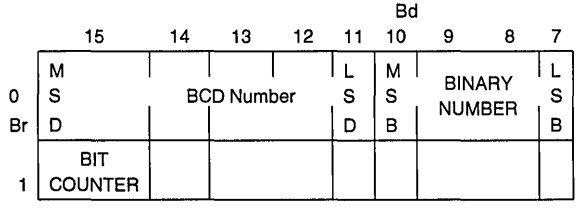


FIGURE 5-17. RAM Map for Binary to BCD Conversion

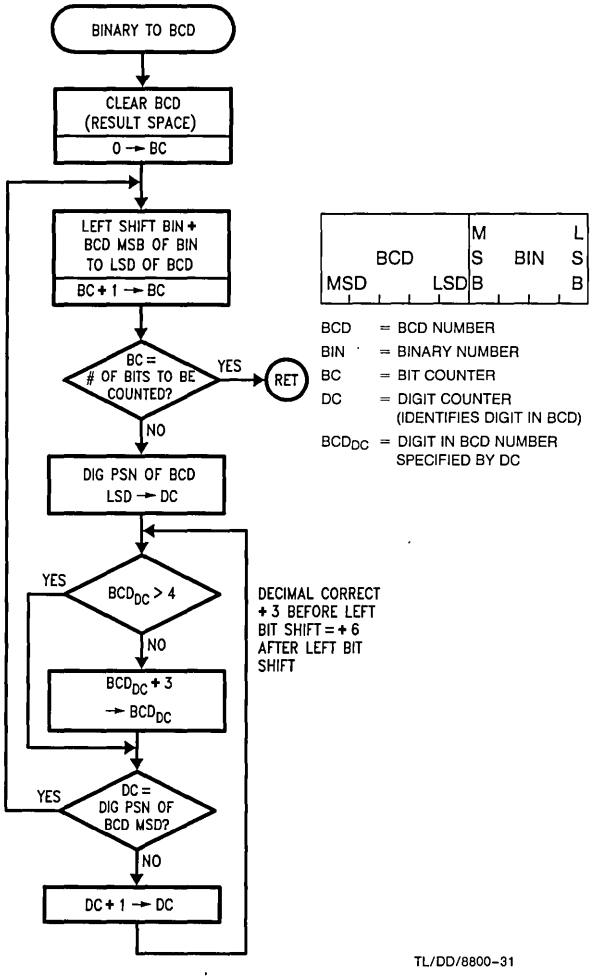


FIGURE 5-18. Binary to BCD Conversion—Shifting Algorithm

The routine as written uses no subroutines. However, the presence of a binary double routine in the program would reduce the code required in this routine.

The code is as follows:

```
BINBCD:  LBI   0,11     ; 0 → BCD
CLEAR:   CLRA
         XIS
         JP    CLEAR
         LBI   1,15     ; 0 → bit counter
         STII  0
SHIFT:   RC             ; left bit shift binary number
         LBI   0,7
LOOP1:   LD             ; and BCD number
         ASC
         NOP
         XIS
         JP    LOOP1
FINISH:  LBI   1,15     ; test if done
         LD
         AISC  1
         JP    RDCADJ   ; no done, BCD adjust
         RET
BCDADJ:  X
         LBI   0,11
DECADJ:  LD             ; decimal adjust before left shift
         AISC  11       ; number > 9 after shift is > 4 before shift
         JP    LESS5
         AISC  8        ; number > 4; do net + 3(=+6 after shift)
         NOP
         X
LESS5:   LD
         XIS            ; go through the whole number
         JP    DECADJ
         JP    SHIFT
```

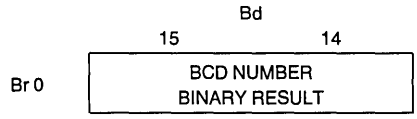
The routine takes 31 ROM words, uses 10 RAM digits, and has a worst case execution time of 1525 instruction cycle times. Unlike the other routines, this routine is data dependent.

5.2.11 BCD to Binary Conversion

Several methods of performing a BCD to binary conversion are presented. The methods, like the binary to BCD routines in Section 5.2.9, illustrate different algorithms and different programming techniques.

An Efficient Two-Digit BCD to Eight-Bit Binary Routine

Figure 5-19 is the flow chart for a very efficient routine for converting a two-digit BCD number to its binary equivalent. The routine uses only two digits of RAM, replacing the BCD number with its binary representation. The simple RAM map is indicated below:

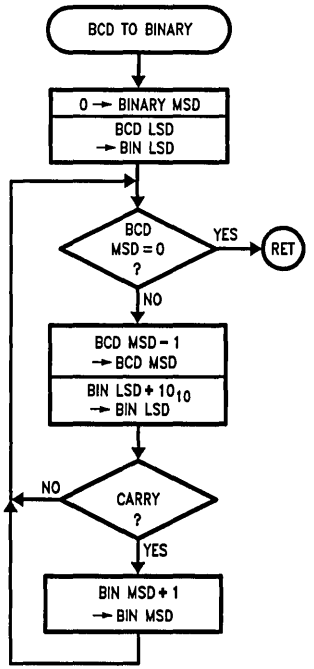


The routine is as follows:

```
BCDBIN:   LBI   0,15     ; clear MSD of BCD number, save value in A
          CLRA
NOCARY:   X
          LBI   0,14     ; point to LSD of BCD number
          AISC  15       ; if MSB (saved in A) = 0, done
          RET
          X              ; else subtract 1 from it
          AISC  10       ; add 10 to the ""binary'' number
          JP    NOCARY
          XIS            ; if carry, must add 1 to binary ""MSB''
          X
          AISC  1
          JP    NOCARY   ; loop until done. This word never
                         ; can be skipped since max BCD number = 99
```

This is a simple routine implemented in an "obscure" manner. The routine takes only 13 words, uses 2 RAM digits only, and has a worst case execution time of 104 instruction cycles. The execution time is data dependent. The minimum execution time is 6 instruction cycles.

Some attention should be given to this routine. It is a good example of code sharing, efficient use of memory, and clever use of the instructions. The routine uses the accumulator both as temporary storage and as work space for the arithmetic. The two RAM digits also serve multifunctions such as accumulate the result, temporary storage, and the input number. To be sure, a great deal of this routine's efficiency comes from the fact that we are working with only two digits. However, the techniques with the accumulator illustrated in this routine have much broader applicability.



TL/DD/8800-32

FIGURE 5-19. Two-Digit BCD to Binary Conversion

BCD to Binary Conversion—Multiply By 10

This routine is the counterpart to the binary to BCD conversion by the doubling technique. As written, the routine will convert a five-digit number (≤ 65535) to its 16-bit binary equivalent. The routine again comes from the standard expansion

$d_4d_3d_2d_1d_0 = d_4 10^4 + d_3 10^3 + d_2 10^2 + d_1 10^1 + d_0$

d_n = digit within a decimal number.

The preceding expression can be rewritten in the following form:

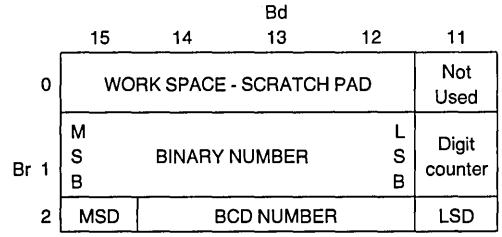
$d_4d_3d_2d_1 = 10\{10[10 < 10(d_4) + d_3 > + d_3 > + d_2] + d_1\} + d_0$

By merely evaluating the right-hand expression above (and adding and multiplying by 10 in binary), the conversion is accomplished. For general information, the scheme is number base independent and can, therefore, be used to convert a decimal number to any desired number base: merely carry out the adds and multiplies in the desired base.

Analysis of the expression above yields the following iterative procedure: Multiply MSD of decimal number by 10, add the next MSD to that quantity, multiply the result by 10, add the next MSD to that result, multiply the result by 10, etc. until the LSB of the decimal number is added. The conversion is complete after this final addition.

The Straightforward Implementation

The flow chart of *Figure 5-20* is the direct expression of the preceding math. The first example will be a straightforward implementation of that flow chart. The routine is written to convert a five-digit BCD number (< 65536) to a 16-bit binary number. The RAM map is shown below.



The routine uses the following subroutines, assumed to be located in Page 2.

```
CLEAR:    CLRA
CLEAR2:   XIS
          JP    CLEAR
          RET
BINADD:   RC            ; R1 + R0 → R1, 16 bit
          LBI   0,12
BINAD1:   LD    1
          ASC
          NOP
          XIS   1
          JP    BINAD1
          RET
BINDBL:   LBI   1,12    ; 2 x R1 → R1, binary
          RC
DBL:      LD
          ASC
          NOP
          XIS
          JP    DBL
          RET
```

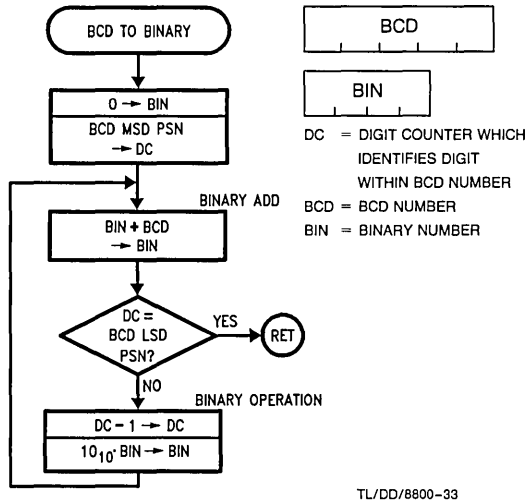


FIGURE 5-20. BCD to Binary Conversion—Multiply by 10

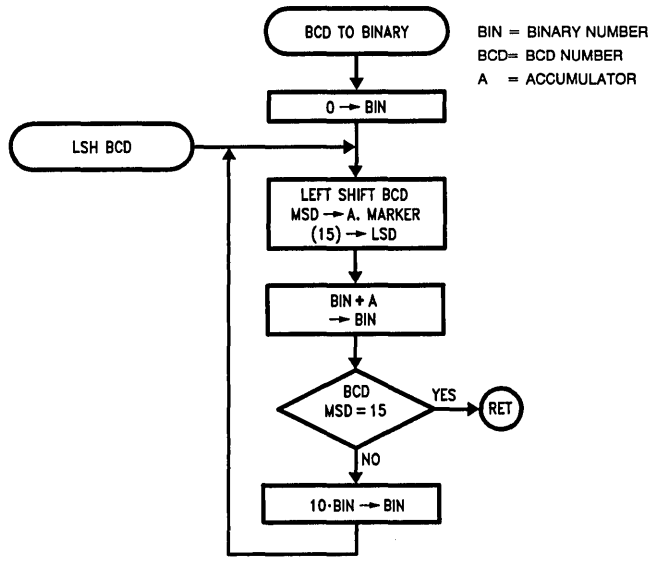
The main body of the routine is as follows:

```
BCDBIN:   LBI   1,11     ; load digit counter
          STII  15       ; position of BCD MSD
          JSRP  CLEAR    ; 0 → binary number
DECBIN:   LBI   1,11
          LD    3
          CAB            ; point to digit in BCD number
          LD
          LBI   0,12     ; and put it into R0
          JSRP  CLEAR2   ; note call into middle of subroutine
          JSRP  BINADD   ; add the digit to the rest of the number
          LBI   1,11     ; now test if finished
          LD
          AISC  4        ; if digit counter < 12 done
          RET
          AISC  11       ; this results in net subtract 1
          X              ; save new value of digit counter
          JSRP  BINDBL   ; now multiply by 10, first do 2 x R1 → R1
          LBI   1,12
XFER:     LD    1        ; transfer R1 → R0, could be a subroutine
          XIS   1
          JP    XFER1
          JSRP  BINDBL   ; 2 x R1 → R1
          JSRP  BINDBL   ; 2 x R1 → R1
          JSRP  BINADD   ; result after all this is 10 x R1 → R1, binary
          JP    DECBIN
```

The routine uses 25 ROM words, not counting the 20 words of subroutine. The routine uses 14 RAM digits and executes in 709 instruction cycles. The original number is preserved and the routine uses one subroutine level.

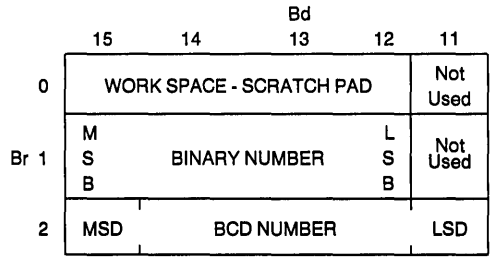
The Shifting Approach

Consider a slightly modified version of the preceding routine. *Figure 5-21* shows this modification. The important differences are 1) there is no digit counter or pointer and 2) the original number is lost. The RAM map follows *Figure 5-21.*



TL/DD/8800-34

FIGURE 5-21. BCD to Binary Conversion—Multiply by 10—the Shifting Approach



This version of the routine uses the same subroutines as given in the preceding section.

```
BCDBIN:   LBI   1,12     ; clear binary number
          JSRP  CLEAR
LSHBCD:   CLRA           ; left shift BCD number, MSD → A;
          COMP           ; marker to LSD
          LBI   2,11
LSH:      XIS
          JP    LSH      ; A → R0, digit 12, 0 to rest of R0
          LBI   0,12
          JSRP  CLEAR2
          JSRP  BINADD   ; add the digit to converted value
          LBI   2,15
          LD             ; test for done, if BCD MSD=15, finished
          AISC  1
          JP    TIMES10
          RET
TIMES10:  JSRP  BINDBL   ;multiply by 10
          LBI   1,12
```

```
XFER:   LD    1
        XIS   1
        JP    XFER1
        JSRP  BINDBL
        JSRP  BINDBL
        JSRP  BINADD
        JP    LSHBCD
```

This outline uses 24 ROM words, not counting the 20 subroutine words and 13 RAM digits. It executes in 731 instruction cycles.

This second approach, a slightly different implementation of the same basic algorithm, uses slightly less memory and executes slightly slower than the straightforward approach.

A "Paper and Pencil" Method and a Common Mistake

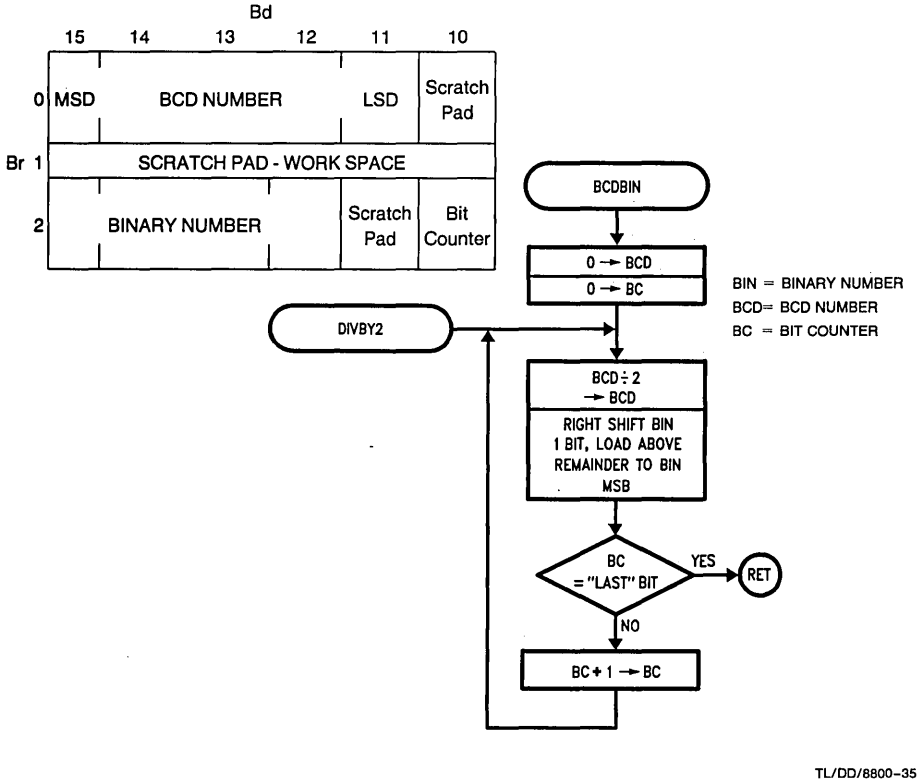
One of the standard methods for base conversion, at least on paper, is the technique of successively dividing the original number by the destination base. The remainders constitute the digits of the converted number. Thus to convert from BCD to binary, simply divide the BCD number by two repeatedly. See the simple example below:

	NUMBER	REMAINDER	
2	17		
	8	1	LSB
	4	0	
	2	0	
	1	0	
	0	1	MSB

Thus, $17_{10} = 1000_{12}$. The technique is well established and useful in instruction.

A conversion sheme using this algorithm is presented. This scheme is presented for comparison and for illustration techniques, e.g., a decimal divide by two without a divide routine. This particular scheme for BCD to binary conversion is not recommended since it is neither more memory efficient nor faster than techniques previously shown. This approach is simply the implementation of a well known conversion technique and serves to illustrate the effect of the algorithm itself on the code.

The flow chart for this algorithm is given in *Figure 5-22*. The RAM map is given below.



TL/DD/8800-35

FIGURE 5-22. BCD to Binary Conversion by Successive Divide by Two

The routine uses the following subroutines, assumed to be located in Page 2.

```
RSH:     XDS          ; a simple 5- or 6-digit right shift
RSH2:    XDS
         XDS
         XDS
         XDS
         X
         RET
BINDBL:  RC           ; 2 x R2 → R2; binary
         LBI  2,11
BDLOOP:  LD
         ASC
         NOP
         XIS
         JP   BDLOOP
         RET
ADD:     RC           ; R1+R0 → R0, decimal
         LBI  1,10
ADLOOP:  LD   1
         AISC 6
         ASC
         ADT
         XIS  1
         JP   ADLOOP
         RET
```

The main body of the routine follows:

```
BCDBIN:  LBI  2,10    ; 0 → binary number and initialize
CLEAR:   CLRA
         XIS          ; bit counter
         JP   CLEAR
DIVBY2:  LBI  0,15    ; divide BCD number by 2, by first
         CLRA         ; divide by 10 (right digit shift) and
         JSRP RSH     ; then multiply by 5
         LBI  0,10
XFER1:   LD   1       ; R0 → R1, for subsequent adds
         XIS  1
         JP   XFER1
         JSRP ADD     ; 2 x R0
         JSRP ADD     ; 3 x R0
         JSRP ADD     ; 4 x R0
         JSRP ADD     ; 5 x R0, therefore have net divide by 2
         CLRA
         LBI  0,10    ; fetch the remainder
         RMB  2       ; make sure it is only 0 or 1
         X
         LBI  2,15    ; load remainder to binary number and
         JSRP RSH2    ; shift right 4 bits
         JSRP BINDBL  ; now shift left 3 bits
         JSRP BINDBL
         JSRP BINDBL  ; net effect is 1-bit right shift of
                      ; binary number with divide remainder going into MSB
TSTFIN:  LBI  2,10
         LD           ; test bit counter for done, if not
         AISC 1       ; increment bit counter
         JP   TFIN2
         RET
TFIN2:   X
         JP   DIVBY2
```

This routine takes 31 ROM words, plus 24 words in the subroutine page; uses 18 RAM digits; and executes in approximately 5500 instruction cycles.

There is little reason to recommend this routine over the others presented. It takes significantly more memory and executes significantly slower. The only benefit, if it can be so termed, is that it implements a commonly known procedure. The lesson is obvious: Do not assume that the "standard" procedure will yield the most efficient implementation. The programmer should be prepared to investigate various algorithms and approaches in the interest of efficiency.

5.3 TIMEKEEPING ROUTINES

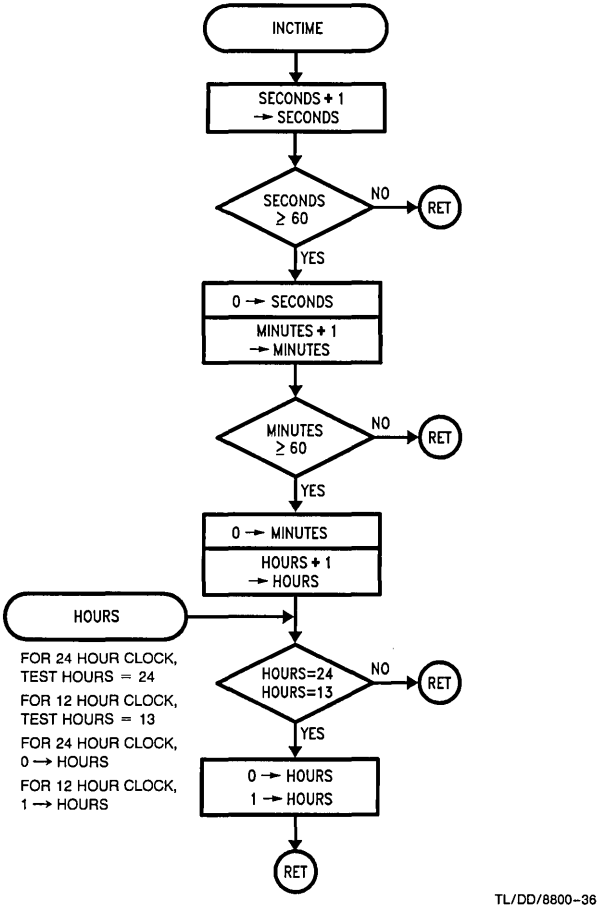
Several routines for keeping time are presented. These include routines for a 12- or 24-hour clock based on internal or external timing references.

5.3.1 Basic Clock Routines - External Input

The following two routines implement a basic clock. The two routines do the same thing. One is written as a 12-hour clock, the other as a 24-hour clock. This, however, is not the significant difference between them. Both routines use the RAM map below, and it is assumed that both the routines are called once per second on the basis of a 1-Hz input signal.

	Bd					
	15	14	13	12	11	10
Br 3	Hours MSD	Hours LSD	Minutes MSD	Minutes LSD	Seconds MSD	Seconds LSD

The flow chart of *Figure 5-23* applies to both routines. The flow chart indicates the minor differences when implementing a 12- or 24-hour clock. Note that both routines have implemented the same flow chart in different ways.



TL/DD/8800-36

FIGURE 5-23. Basic Block Flow Chart

The first implementation, Version I, uses a master increment loop which increments seconds, minutes, and hours as required. The loop handles the overflow from 60 to 00 in the seconds and minutes. Version I is written as a 24-hour clock.

```
INCTIME:  LBI   3,10     ; point to seconds LSD
PLUS1:    SC             ; add 1
          CLRA
          AISC  6
          ASC
          ADT
          XIS            ; LSD incremented, point to MSD
          CLRA
          AISC  6        ; increment saved in C
          ASC            ; increment MSD of seconds, minute or hours
          ADT
          X              ;
                         ; test = 6, if so correct to 0 and move
          LD             ; to next digit. If not, exit
          AISC  10
          JP    HOURS    ; will always escape loop here if get to hours
          STII  0
          JP    PLUS1
HOURS:    LBI   3,15     ; test if hours need to be corrected
          LD             ; here testing for hours ≥ 24
          AISC  14
          RET            ; hours ≥ 20
          LBI   3,14
          LD
          AISC  12
          RET            ; hours < 24
          STII  0        ; hours ≥, therefore set to 0
          STII  0
          RET
```

This routine takes 28 ROM words, 6 RAM digits, and has a worst case execution time of 58 instruction cycle times. The routine uses no subroutines and execution time is data dependent. Minimum execution time is 19 instruction cycle times.

The second implementation, Version II, is a more direct implementation of the flow chart shown in *Figure 5-23.* It moves sequentially through the clock data, incrementing and adjusting as required.

```
INCTIME:  LBI   3,10     ; point to seconds LSD
          JSRP  PLUS1    ; 2 digit BCD increment
          LD
          AISC  10       ; mod 6 correct
          RET            ; seconds MSD < 6, exit
          XIS
          JSRP  PLUS1    ; increment minutes
          LD             ; mod 6 correct
          AISC  10
          RET            ; minutes MSD < 6, exit
          XIS
          JSRP  PLUS1    ; increment hours
          LD             ; now do hours adjust
          AISC  15
          RET            ; exit if hours MSD = 0
          LBI   3,14     ; hours MSD = 1, test hours LSD < 3
LD
AISC            13
          RET            ; hours ≤ - exit
          STII  1        ; hours = 13, set to 01 and exit
          STII  0
          RET
```

The routine uses the following subroutines, assumed to be located in Page 2.

```
PLUS1:    SC           ; 2 digit BCD increment
          CLRA
          AISC  6
          ASC
```

```
ADT
XIS
CLRA
AISC  6
ASC
ADT
X
RET
```

The routine takes a total of 34 ROM words, 22 in the main routine and 12 in the subroutine; uses 6 RAM digits; and has a worst case execution time of 58 instruction cycle times. Execution time is data dependent with the minimum execution time being 17 instruction cycles. The routine uses one subroutine level.

1-Hz Input and 50- or 60-Hz Input

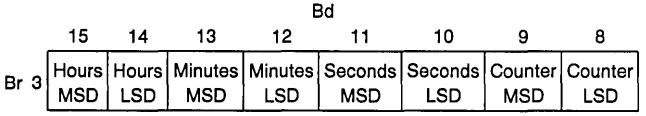
The two routines provided are written assuming they are called as a result of a 1-Hz signal. It is a simple task to modify the routines for a 50- or 60-Hz input signal. As Version I is the more code efficient routine, the necessary modifications will only be illustrated for that implementation.

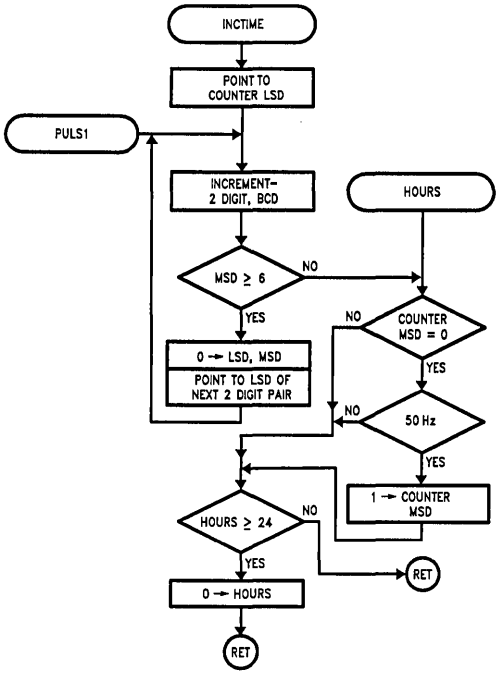
60-Hz Only Input

If the signal source is a 60-Hz signal, the modification is trivial. By simply changing the first LBI from LBI 3,10 to LBI 3,8, the routine becomes a clock increment based on a 60-Hz input. The rest of the routine is completely unchanged. Of course, two extra RAM digits are used, digits 3,9 and 3,8, to count the 60-Hz signal. Also, as should be expected, worst case execution time increases.

A General 50- or 60-Hz Input

It is fairly simple to modify the routine to operate with either a 50-Hz or 60-Hz reference input. The modification will use the characteristic described in the preceding paragraph. For a 50-Hz input, the frequency counter is set to 10 rather than 00. Otherwise, the routine remains the same. The routine arbitrarily selects G2 as the input line to define whether the input is 50-Hz or 60-Hz. *Figure 5-24* is the flow chart shown in *Figure 5-23* modified to indicate the specific implementation and the 50- or 60-Hz feature.





TL/DD/8800-37

FIGURE 5-24. Clock Based on 50- or 60-Hz Input

```
INCTIME:  LBI   3,8      ; point to counter LSD
PLUS1:    SC
          CLRA           ; 2 digit BCD increment by 1
          AISC  6
          ASC
          ADT
          XIS
          CLRA
          AISC  6
          ASC
          ADT
          X
          CLRA
          AISC  6        ; now test MSD ≥ 6
          SKE
          JP    HOURS
          STII  0
          JP    PLUS1    ; is ≥ 6, correct to 0 and continue
HOURS:    LBI   3,9      ; test counter MSD = 0
          CLRA
          SKE
          JP    HOURS2
          OGI   15       ; now test 50- or 60-Hz, set G2 high
          SKGBZ 2
          STII  1        ; if 50-Hz, 1 → counter MSD
HOURS2:   LBI   3,15     ; G2 = 1 indicates 50-Hz input
          LD
          AISC  14
          RET            ; hours MSD < 2
          LBI   3,14     ; hours MSD = 2, test hours LSD < 4
          LD
          AISC  12
          RET            ; hours < 24
          STII  0
          STII  0        ; hour ≥ 24, set to 0
          RET
```

The routine uses 39 ROM words, the extra words being used to read the input and adjust the counter accordingly, and 8 RAM digits. Input $G_2 = 1$ indicates a 50-Hz input signal.

12- or 24-Hour Capability

It is a trivial matter to expand the routine further to give it the option of 12- or 24-hour capability. *Figure 5-23* indicates the differences, which are minor. One need only test another input and alter the hours digits accordingly.

5.3.2 Clock Routines Based on Internal Timer

The internal timer of COPS microcontrollers can be used as the time reference for a clock. Routines using this feature must count timer overflows. These overflows are dependent, of course, on the operating frequency of the microcontroller. This points out a major restriction on this type of clock routine: It is impossible for the clock to be more accurate than the oscillator frequency. Another difficulty is that the selection of operating frequency may give a fractional SKT, timer overflow, frequency. This complicates the routine by requiring compensation for this fractional frequency.

An SKT-Based Timekeeping Routine

The following routine is representative of the worst case conditions when using the internal timer as a clock time base: A common, inexpensive crystal is used for the oscillator and creates a fractional SKT frequency. The following information is essentially a duplication of Section 4.9 of the *COPS Family User's Guide.* It is presented here for completeness.

The routine presented here is a 12-hour clock using the SKT overflow as the time base. The oscillator used will be based on a 3.579545 MHz-crystal, the inexpensive, readily available TV crystal. Therefore, a high-speed part (e.g., COP420) with the divide by 16 option must be used. The SKT overflow frequency is the instruction cycle frequency (here 3.579545 MHz divided by 16) divided by 1024 or, in this case, 218.478 Hz. Therefore, the timekeeping calling routine must execute an SKT instruction at an approximate 218-Hz rate to guarantee detection of every SKT overflow. The routine must compensate for the non-integer SKT overflow frequency to provide timing accuracy.

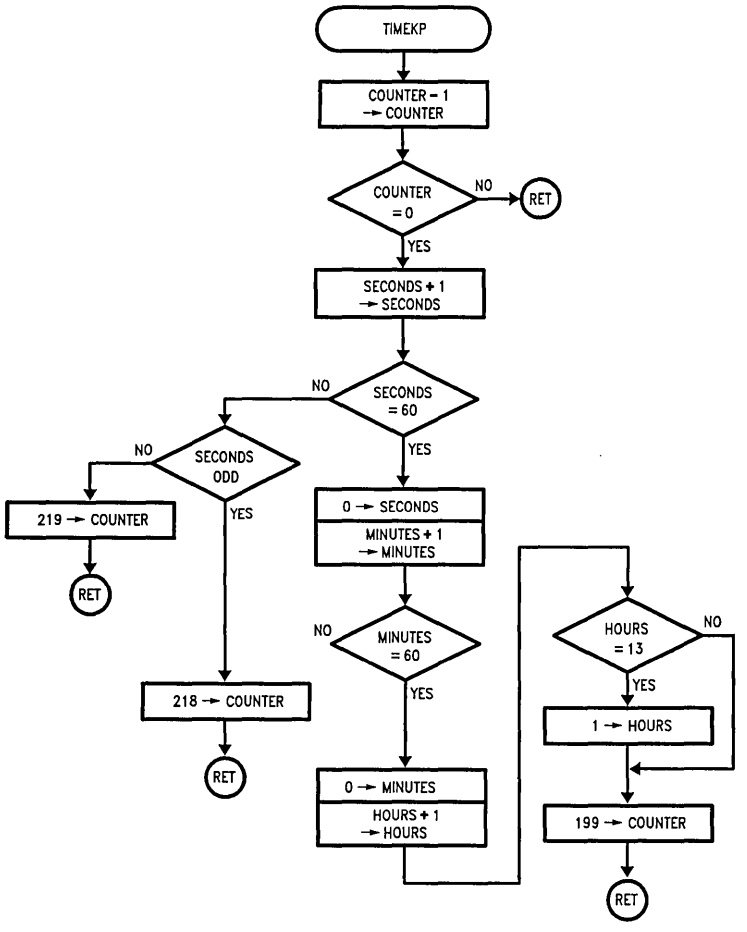
Compensation is achieved by establishing a counter for the SKT overflows. Seconds are incremented when this counter reaches 0. This counter is preset to various values, from which it is counted down, at various points in the routine. The details of the compensation are as follows:

- Every odd second in the range of 0–59 seconds, the counter is set to 218.
- Every even second in the range of 0–59 seconds, the counter is set to 219.
- Every minute in the range of 0–59 minutes, the counter is set to 218.
- Every hour the counter is set to 199.

Regardless of the preset, the counter is decremented every time the SKT instruction skips, i.e., an SKT overfow is detected. The technique previously described will provide accuracy at the end of each hour. The short term inaccuracies during the hour are small. The *COPS Family User's Guide* explains why this particular compensation scheme works and the reader is referred to the manual for explanation.

Figure 5-25 is the flow chart and RAM map for this routine. Note that the counter for SKT overflows is binary. Also note that the hours portion of the clock is binary, to save RAM, and that the minutes and seconds portions of the clock are BCD. The routine is located outside Page 2 and uses a subroutine located in Page 2.

	Bd 15	14	13	12	11	10	9
Br 2	COUNTER FOR SKT OVERFLOWS		HOURS (BINARY)	MINS MSD(BCD)	MINS LSD(BCD)	SEC MSD(BCD)	SECS LSD(BCD)
	MSD	LSD					



TL/DD/8800–38

FIGURE 5-25. Flow Chart for Internal Time Base Clock (Oscillator Frequency = 3.579545 MHz)

```
TIMEKP:   LBI   2,14     ; point to low-order digit of counter
DECR:     LD             ; decrement the counter by 1
          AISC  15
          JP    NEXTDIG
          X              ; counter = 0 return to main routine
          RET
NEXTDIG:  XIS            ; if skip executed, counter is 0
          JP    DECR
SECONDS:  LBI   2,9      ; points to seconds LSD
          JSRP  INC2     ; 2 digit BCD increment with MOD6 adjust
          JP    TSEC     ; seconds < 60, test ODD or EVEN
          STII  0        ; seconds = 60, 0 → seconds, increments mins.
          JSRP  INC2
          JP    C218     ; minutes < 60, set counter = 218
          STII  0        ; 0 → minutes, increment hours
          LD
          AISC  1
          X
          AISC  4        ; test hours > 12
          JP    C199     ; no, set counter to 199
          STII  1        ; yes, set hours to 1 and counter to 199
C199:     LBI   2,14
          STII  7        ; set counter = 199 (binary 12,7)
          STII  12
          RET
TSEC:     LBI   2,9      ; point to seconds LSD to test ODD/EVEN
          SKMBZ 0
          JP    C218     ; seconds ODD, set counter to 218
C219:     LBI   2,14     ; seconds EVEN, set counter to 219
          STII  11       ; 219 = binary 13,11
C21X:     STII  13
          RET
C218:     LBI   2,14     ; 218 = binary 13,10
          STII  10
          JP    C21X
```

This routine uses the following subroutine:

```
INC2:     SC             ; 2-digit BCD increment
          CLRA
          AISC 6
          ASC
          ADT
          XIS
          CLRA
          AISC  6
          ASC
          ADT
          X              ; now test if reached 60
          LD
          AISC  10
          RET            ; 2 digits < 60
          RETSK          ; 2 digits = 60
```

It should be clear that a more convenient choice of oscillator frequency would significantly reduce the code in this routine. An integer SKT overflow frequency would reduce the routine to, essentially, one of the routines shown initially.

5.4 DATA MANIPULATION AND STRING OPERATIONS

5.4.1 Register Transfers

Several routines are provided for transferring data between registers. Some more or less specialized routines are presented along with a completely general routine.

Four Register Blocks

The LD, XIS, XDS, and X instructions have an exclusive OR argument which permits easy data transfer among the registers within a four register block, registers 0–3, 4–7, etc. Moving data across a register block boundary is less efficient and the general purpose routines have to be used. Within the register block, the following routines can be used:

```
XFER1:  LD      1       XFER2:  LD      2       XFER3:  LD      3
        XIS     1               XIS     2               XIS     3
        JP      XFER1           JP      XFER2           JP      XFER3
        RET                     RET                     RET
```

Note: LDS can be used in place of XIS in any of these routines.

Routine XFER1 will transfer data from R0 to R1, R1 to R0, R2 to R3, or R3 to R2. Routine XFER2 will transfer data from R0 to R2, R2 to R0, R1 to R3, R3 to R1. Routine XFER3 will transfer data from R0 to R3, R3 to R0, R1 to R2, or R2 to R1. The direction of the transfer depends only on the status of the B register when the routine is executed. In fact, the routines are commonly preceded by one or more LBI instructions. The successive skip feature of the LBI instruction is very powerful when used in conjunction with these routines.

Register exchanges within the four register blocks are written in much the same way as the following routine indicates.

```
SWAP1:  LD      1
        X       1
        XIS     1
        JP      SWAP1
        RET
```

This routine will exchange the contents of the R0 and R1 or R2 and R3. Similar routines for the other registers can also be written in the same manner as the data transfers. Again, XDS may be used in place of XIS.

Completely General Transfers

A completely general register transfer routine is indicated below. The routine uses a RAM digit for temporary storage. The routine is called by setting up the source register with an LBI and establishing the destination register number in the accumulator. RAM digit TEMP is any convenient digit.

```
LOOP:   XAD     TEMP
        XABR
XFER:   XAD     TEMP    ; XFER is the entry point for the routine
        LD
        XAD     TEMP
        XABR
        XAD     TEMP
        XIS
        JP      LOOP
        RET
```

The calling sequence for the routine is as follows:

```
LBI     SOURCE
CLRA
AISC    N               N defines destination register
JSRP    XFER
```

Obviously, if a transfer from RN to RK is common, the setup can be included in the subroutine.

The routine can be rewritten in the following form and the calling sequence modified as follows:

```
CALLING SEQUENCE:
LBI     TEMP
STII    N               ; destination register
LBI     SOURCE
JSRP    XFER
```

The subroutine is as follows:

```
LOOP:   JSR     EXCH    EXCH:   XAD     TEMP
XFER:   LD                      XABR
        JSR     EXCH            XAD     TEMP
        XIS                     RET
        JP      LOOP
        RET
```

There is no particular benefit in doing this for the simple register transfer but it will result in code savings where register swaps, general purpose swaps, are also required.

The routine for a general purpose register swap and the calling sequence are given below.

```
CALLING SEQUENCE:
LBI     TEMP
STII    N           ; one register number
LBI     SOURCE
JSRP    SWAP
```

The SWAP subroutine is:

```
SWAP2:  JSR   EXCH
SWAP:   LD            ; entry point for the routine
        JSR   EXCH
        X
        JSR   EXCH
        XIS
        JP    SWAP2
        RET
```

Subroutine EXCH is the same routine as indicated in the general purpose transfer.

5.4.2 Shift Routines

Right Digit Shift

The following routines will perform right digit shifts. The first routine shifts right one digit from the starting B address to the end of the register. The second routine shifts an arbitrary four-digit group right one digit. Both routines place a "0" in the starting digit and leave the previous contents of the last digit in the accumulator.

```
                          I
RSHIFT: CLRA          ; to put 0 to first digit
RSH:    XDS
        JP    RSH     ; simple right shift loop, exit on XDS skip
        RET

                        RSHIFT:
        XDS
        XDS           ; shift 4-digit block right one digit
        XDS
        X             ; save value of last digit in A
        RET
```

Left Digit Shift

The following routines will perform left digit shifts. The first routine shifts left one digit from the starting B address to the end of the register. The second routine shifts an arbitrary four-digit group left one digit. Both routines place a "0" in the starting digit and leave the previous contents of the last digit in the accumulator.

```
                          I
LSHIFT: CLRA          ; to put 0 to first digit
LSH:    XIS
        JP    LSH     ; simple left shift loop, exit on XIS skip
        RET

                          II
LSHIFT: CLRA          ; to put 0 to first digit
        XIS
        XIS           ; shift 4-digit block left on digit
        XIS
        X             ; save value of last digit in A
        RET
```

NOTE: The left and right digit shift routines are written in the sense that the direction of increasing Bd value is "left". The direction of decreasing Bd value is "right". It is entirely possible that the user may, for his or her application, wish to reverse this directional sense. This causes no problem and the routines above are merely reversed (i.e., the left shifts become right shifts and vice-versa).

Right Bit Shift

A right bit shift is one of those very few things that COPS microcontrollers do not do well. If the algorithm or approach chosen involves right bit shifting, it is strongly recommended that an alternative approach be used or developed. An alternative nearly always exists and will commonly be COPS code efficient. Rarely, if ever, does the failure to find an alternative to right bit shifting mean that no alternative exists. The programmer should think in broader terms than the specific function of right bit shifting; if an algorithm requires right bit shifting, consider other algorithms for the same function.

However, if there is no choice and right bit shifting must be performed, some routines to perform the shift are presented. Note, right shift has the same directional sense here as in digit right shift; data movement is in the direction of decreasing Bd.

Right Shift Memory Digit 1 Bit

This routine is a simple, straightforward approach to shift a memory digit right one bit. The shifted data is formed in the accumulator and then exchanged into memory. The routine can be written for a simple shift or a right circular shift. Both versions are indicated. The routines take advantage of the bit testing capability of COPS microcontrollers.

```
I - Simple Shift          II - Circular Bit Shift

RBSHIFT: CLRA             RBSHIFT: CLRA
         SKMBZ   3                 SKMBZ   3
         AISC    4                 AISC    4
         SKMBZ   2                 SKMBZ   2
         AISC    2                 AISC    2
         SKMBZ   1                 SKMBZ   1
         AISC    1                 AISC    1
         X                         SKMBZ   0
         RET                       AISC    8
                                   X
                                   RET
```

These routines are not particularly long nor complex and work well. They form the most efficient basis for general right bit shifting in COPS microcontrollers.

Right Shift Using SIO

If the SIO register is not otherwise being used, it can be used to perform a right circular shift of the data in the accumulator. This technique requires that pins SO and SI of the microcontroller be tied together externally. The routine is then reduced.

```
RSHIFT:  XAS    ; SIO must be in shift register mode
         NOP
         NOP
         XAS
         RET
```

The SIO register shifts left one bit each instruction cycle when it is enabled as a shift register. Thus, a right bit shift is achieved by three left bit shifts.

Left Bit Shifts

Left bit shifts are easy to perform even though there is no bit shift instruction. Bit left shift has the same directional sense as digit left shift; data movement is in the direction of increasing Bd.

Left Bit Shift by Means of Binary Double

Left shifting a value by one bit is equivalent to a binary doubling of that value. Thus, a binary doubling routine can be used for left bit shifting. Two routines are provided; one simply left shifts a single digit 1 bit; the other shifts several digits left 1 bit.

```
I - Single Digit          II - Multidigit

LBSHIFT: LD               LBSHIFT: RC
         ADD              LSHFT:   LD
         X                         ASC
         RET                       NOP
                                   XIS
                                   JP      LSHFT
                                   RET
```

These two routines perform the left shift in the same manner. The number is added to itself to do a binary double. The second version remembers the state of the MSB of a given digit in C so shifting can be performed across the digits.

Use of SIO for Left Bit Shifting

The SIO register can be used to shift the data in the accumulator left one bit. In the shift register mode, SIO is always shifting left. This normal operational feature can be used to advantage. The routine is simplicity itself:

```
LBSHIFT: XAS    ; SIO must be in shift register mode
         XAS
         RET
```

A and SIO are simply swapped twice. Since SIO is always shifting (in shift register mode), this results in a net one bit left shift. This routine does not require that SI and SO be tied together and is therefore more or less unrestricted in its use. The user must remember that the state of SI, whatever it may be, is shifted into SIO and that the LSB of the accumulator after this routine will be controlled by the state of SI during the shift. Tying SI to SO will result in a left circular shift of one bit, the MSB of the accumulator will be moved to the LSB as the left bit shift occurs.

5.4.3 Data/String Compare

A routine to compare two strings of data or characters is provided. It is the same routine that would be used to compare two registers (within the four register blocks). The RAM map for this routine is indicated below:

	Bd 15	14	13	12	11
Br 1			String 1		
2			String 2		

The routine is setup as a subroutine. It will simply return if the strings are not equal and return and skip if the two strings are identical. By changing the starting LBI, larger strings can be tested.

```
COMPARE:  LBI   1,11     ; initialize B
CMPR:     LD    3        ; load value to A, point to other register
          SKE            ; test equal
          RET            ; not equal, return
          XIS   3
          JP    CMPR
          RETSK          ; all digits equal, return and skip
```

The preceding routine is excellent if the data is placed so that it can be used. The programmer should strive to place data in RAM so that routines such as the one previously illustrated can be used. However, data is not always located in the most efficient places. Therefore, a general purpose compare routine is provided. This routine will compare a three-digit string located in (1,10), (1,11) and (1,12) to another three-digit string located in (3,7), (3,8), and (3,9).

	Bd 12	11	10	9	8	7
Br 1		String 1			Not Used	
3		Not Used			String 2	

```
COMPARE:  LBI   1,10     ; initialize B register
          LDD   3,7      ; fetch first digit to compare
          SKE
          RET            ; not equal
          XIS            ; point to next digit
          LDD   3,8      ; fetch second digit
          SKE
          RET            ; not equal
          XIS
          LDD   3,9      ; fetch third digit
          SKE
          RET            ; not equal
          RETSK          ; strings equal
```

This routine is general and the two strings could be located anywhere. By merely supplying the proper values in the LBI and LDD instructions, the routine is modified for data in locations other than those indicated here.

5.4.4 String Search

It is often necessary to search data memory for a string of characters. This routine will search register 0 for the three character string located in digits (2,15), (2,14), and (2,13). The routine simply returns if no match and returns and skips if the string is found.

```
SEARCH:   LBI   0,15     ; initialize B register
CHAR1:    LDD   2,15     ; fetch first character
          SKE
          JP    DECR     ; not equal, move B register
          XDS
          JP    CHAR2    ; matched first character, test second
          RET            ; string not found in register 0
CHAR2:    LDD   2,14     ; fetch second character
          SKE
          JP    CHAR1    ; no match
          XDS
          JP    CHAR3
          RET            ; string not found in register 0
```

```
CHAR3:    LDD   2,13     ; fetch third character
          SKE
          JP    INCR
          RETSK          ; string found
DECR:     LD             ; no match, move Bd down
          XDS
          JP    CHAR1    ; and start over
          RET            ; moved over the end, string not found
INCR      LD
          XIS
          JP    CHAR1
```

Remember, the routine is searching for the contiguous three-digit group and exists via RETSK when that group is found.

5.4.5 RAM Clear Routines

Routines that clear the data memory are commonly required in programs. Some of the more standard techniques are indicated here.

Single Register Clear

The following routines will clear all or part of a register. They are normally preceded by an LBI instruction.

```
            I                          II
CLEARX:   LBI     START    CLEARX:   LBI     START
CLR:      CLRA             CLR:      CLRA
          XIS                        XDS
          JP      CLR                JP      CLR
          RET                        RET
```

The routines are equivalent. Routine I clears the data in the register from the digit defined by START up to and including digit 15. Routine II clears the data in the register from the digit defined by START down to and including digit 0.

Clearing Entire RAM

It is a common requirement that the entire RAM be cleared at power up or on the basis of a master clear operation or both. This can be done by calling the register clear instructions provided previously. It will usually be more code efficient to use the routine provided here.

```
MCLEAR:   LBI   0,0
          CLRA
          AISC  N        ; N = highest number of register in device
                         ; N = 3 for COP420, N = 7 for COP444L, etc.
LOOP:     XABR
CLR:      CLRA           ; these three words could be replaced
          XIS            ; with a subroutine call to CLR
          JP    CLR      ; subroutine defined above
          XARR
          AISC  15       ; decrement BR
          JP    LOOP
```

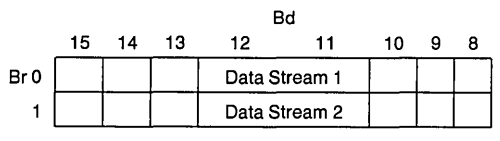
The routine merely establishes the maximum value of BR allowed in the device—or desired to be cleared—and successively clears each register.

5.5 INPUT/OUTPUT

This section deals with the techniques for getting data in and out of COPS microcontrollers. Some of this is straightforward since COPS devices have independent instructions for input and output.

5.5.1 Table Look Up

The LQID instruction makes outputting converted data very simple. It is powerful in its own right as a table look-up instruction but that power is increased if it is necessary to output the table values. A routine to output information is shown below. The table is not shown but is obviously required. Note that the table may be any kind of code conversion: BCD to Seven Segment, ASCII conversion, etc. The output is not affected by the table contents. By virtue of the successive LBI feature, the routine is set up to output either of two data streams.



```
OUTDS1:   LBI   0,8      ; this entry point will output data stream 1
OUTDS2:   LBI   1,8      ; this entry point will output data stream 2
          CLRA
          AISC  N        ; setup accumulator for Table location
OUTPUT:   OBD            ; output digit position on D lines
          LQID
          X              ; this allows movement through
          XIS            ; the data without disturbing the data
          JP    OUTPUT   ; or the accumulator
          RET
```

The routine assumes that the L drivers have been enabled prior to calling the routine. Note that the LQID instruction loads the Q register. The L drivers must be enabled to output the data in Q. Remember also that the LQID instruction uses a subroutine level in some COPS microcontrollers.

5.5.2 Microbus I/O

Microbus I/O is, of course, relevant only to those COPS microcontrollers which have the Microbus option implemented. This option makes the code required for the interface simplicity itself. Only one caution is necessary: Do not enable the L drivers, i.e., do not set EN_2, on Microbus parts. COPS Microbus devices are structured to be peripheral devices for some host processor. The host has control over the L drivers via the chip select, read strobe, and write strobe.

As stated earlier G0 is the handshake line for the Microbus interface. It is the responsibility of the COPS program to set G0 to a 1 level to indicate the COPS device is ready for access by the host. A write to the COPS Microbus peripheral by the host will set G0 low. A typical sequence for this is as follows:

```
          .
          .
          .
          OGI   1      ; G0 assumed low, 0 prior to this
                       ; set G0 high to indicate COPS ready
WAIT:     SKGBZ 0      ; wait for a write by host
          JP    WAIT
          CQMA         ; G0 was low, a write was performed
          .            ; read the data and continue with
          .            ; the program
          .
```

Note that when the host processor writes to a COPS Microbus device, the host writes directly into the Q register. The COPS microcontroller then merely reads the Q register.

A read by the host is equally simple. Upon seeing G0 high, the host will execute a read operation which takes the Q data out to the eight-bit bus. The only possible difficulty is that the COPS microcontroller does not know that a read has been performed. If it is necessary for the microcontroller to know a read has been performed, the following sequence is recommended.

```
          .
          .
          .
          CAMQ         ; load Q; could use LQID
          OGI   1      ; set G0 high to indicate data ready
WAIT:     SKGBZ 0
          JP    WAIT   ; host acknowledges ready by a dummy write
          JMP   MAIN
          .
          .
          .
```

This sequence outputs the data to Q and then sets G0 high to indicate ready. The host reads the data and then does a dummy write to indicate the data has been read. The microcontroller detects this and then returns to the main loop where G0 is set high and the device waits for the next write.

The procedure above is, of course, not necessary if there is no requirement that the COPS microcontroller know that a read operation by the host has taken place.

5.5.3 Serial I/O—MICROWIRE

Routines for handling serial I/O are provided. Two versions of output routines are provided: a destructive output and a nondestructive output. The routines are written for 16-bit transmissions but are trivially expandable up to 64-bit transmissions by merely changing the initial LBI instruction. The routines are written using the XIS instruction, but the XDS instruction could be used equally well.

The routines arbitrarily select register 0 as the I/O register. It is assumed that the external device requires a logic low chip select. It is further assumed that chip select is high, SK is low, and SO is low on entry to the routines. The routines exit with chip select high, SK low, and SO low. G0 is arbitrarily chosen as the chip select for the external device.

Destructive Data Output

This routine outputs the data under the conditions specified above. The output data is destroyed after it is transmitted.

```
OUT1:   LBI   0,12  ; point to start of data word
        SC          ; set C to enable SK clock
        OGI   14    ; select external device by 0 → G0
        LEI   8     ; enable shift register output
SEND:   LD
        XAS         ; data transmission loop, first
        XIS         ; XAS turns on SK clock
        JP    SEND
        RC
        XAS         ; turn off SK clock, transmission done
        OGI   15    ; deselect external device
        LEI   0     ; set SO to 0
        RET
```

Note that this is a general purpose routine and handles all the overhead except loading the data into R0. The routine takes a total of 17 ROM words and can undoubtedly be reduced in specific applications.

Nondestructive Data Output

This routine is identical to the destructive data output routine except that the transmitted data is preserved in the microcontroller.

```
OUT2:   LBI   0,12  ; point to start of data word
        SC
        OGI   14    ; select the external device
        LEI   8     ; enable shift register mode
        JP    SEND2
SEND1:  XAS
SEND2:  LD          ; data output loop
        XIS
        JP    SEND1
        XAS         ; send last data
        RC          ; wait 4 cycles to data to get out
        CLRA
        NOP
        XAS         ; turn SK clock off
        OGI   15    ; deselect the device
        LEI   0     ; turn SO low
        RET
```

The nondestructive routine takes 21 ROM words, four more than the destructive routine. Again, this is a general purpose routine which can probably be reduced in specific applications.

Serial Data Input

The code for reading serial data is almost the same as the write code. This should be expected because of the nature of the SIO register and the XAS instruction.

The first routine enables shift register mode, selects the external device, and reads the data in. Register 0 is the input register and the routine, as written, is for a 16-bit data stream. As before, the routine is trivially expandable up to 64 bits. G0 is arbitrarily selected as the chip select for the external device. SK is 0, and G0 is high or entry to the routine.

```
READ:   LEI   0     ; enable shift register mode, SO is 0
        OGI   14
        SC
        XAS         ; turn on the clock
        LBI   0,13  ; initialize the B register
        NOP         ; NOPs to preserve the timing
LOOP:   NOP
        XAS
        XIS         ; read all but last four bits in this loop
        JP    LOOP
        RC
        XAS         ; turn off the clock and read last four bits
        OGI   15    ; deselect the device
        RET
```

The routine exits with the data in digits 0,13, 0,14, 0,15, and the accumulator.

A variation on this routine which places the input data in digits 0,12 through 0,15 is presented below. This routine uses one subroutine level.

```
READ:   LEI   0      ; enable shift register mode 0 → SO
        OGI   14     ; select external device
        LBI   0,12   ; initialize B register
LOOP:   JSRP  SIO
        LD           ; data read loop
        XIS
        JP    LOOP
        OGI   15     ; deselect the devices
        RET
```

The following subroutine is used:

```
SIO:    SC           ; turn on SK clock
        XAS
        RC           ; wait 4 cycles for the data to full SIO and
        NOP          ; turn off the clock
        NOP
        XAS
        X            ; put data to memory
        RET
```

These are two implementations of the same basic routine. The first version reads the data in one continuous stream; the second version reads the data in four-bit groups. The second routine uses a little more code. The choice of routine is entirely governed by the application, the peripheral devices used, and not by the microcontroller.

It is fairly common that the peripheral device must be sent some command or instruction directing it to output some data to the MICROWIRE interface. A typical routine of this type is given below. G0 is again chosen as the chip select. It is assumed that the peripheral device requires a start bit followed by four bits of instruction information. Location 0,0 is arbitrarily selected for storage of the instruction data. The routine is again written for 16 data bits. The input portion of the routine is essentially the same routine as the first version above. There is a subtle difference: the data is all placed in RAM and four extra clocks are generated. This is not normally a problem, but if it is, use another form of the input routine. There is no requirement that the input routine must be in this form:

```
READ:   OGI   14     ; select the device
        LEI   8      ; enable shift register
        CLRA         ; setup start bit in A
        AISC  1
        SC           ; turn on clock and send start bit
        XAS
        LDD   0,0    ; fetch command/instruction
        LBI   0,12   ; initialize B register
        XAS          ; send command/instruction
        NOP          ; wait 4 cycles for data to get to
        CLRA         ; the peripheral
        XAS          ; just maintaining the timing, send 0s
        NOP          ; delay - typical required 0 to 3 instruction cycles
        NOP          ; now wait 4 cycles for data to fill SIO
        NOP
LOOP:   CLRA
        XAS          ; data read loop
        XIS
        JP    LOOP
        RC
        XAS          ; turn off the clock
        OGI   15
        LEI   0      ; deselect the device and turn SO off
        RET
```

5.5.4 SI as a General Purpose Input

When not used as part of the MICROWIRE interface, SI can be used as a general purpose input. There are two ways in which this can be done:

1. Leave SIO in shift register mode. SO may be enabled or disabled depending on system requirements. Then reading SI is simple:

```
        CLRA        ; this clear not absolutely necessary
        XAS
        AISC  15    ; test SIO for 0, if 0 SI=0, else SI=1
        JP    SIEQ0
SIEQ1   .
        .
        .
```

2. Put SIO in counter mode. Then SI will capture pulses that meet minimum width requirements. Load SIO with 0 and test for 15. Sample code for this is as follows:

```
        CLRA        ; CLRA required here
        XAS
             AISC  1
             JP    NOPULSE
        PULSE .
             .
```

Remember that this mode captures and remembers the occurrences of a high to low transition at SI input. SIO is in binary counter mode for this method to work.

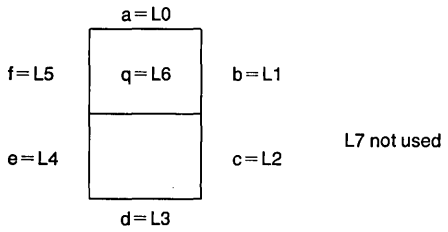
Some devices have the SKSZ instruction. This makes testing SI, or SIO, particularly easy. SKSZ tests the contents of SIO without affecting those contents and generates a skip if SIO is 0. This is essentially the same test as above except that it is a single instruction.

5.6 DISPLAY CONTROL

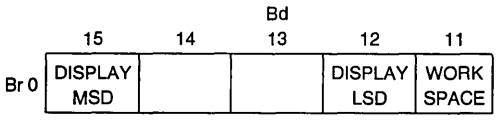
It is frequently required to control a display as part of an application using COPS microcontrollers. There are several approaches to this and this section will attempt to illustrate those approaches.

5.6.1 A Four-Digit Multiplexed Display

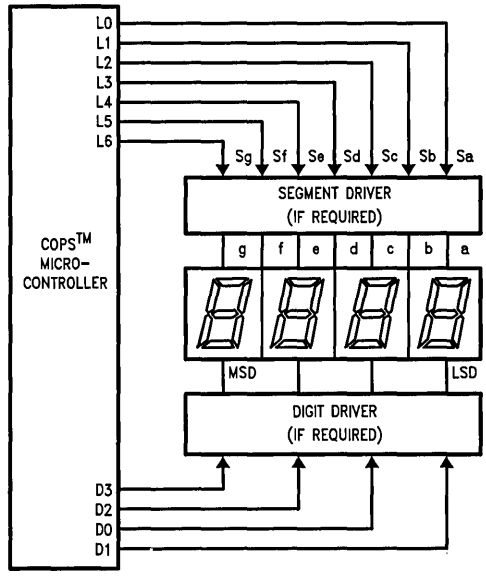
This routine will output a four-digit number to a standard seven segment display. The D lines will be the digit strobes, with D3 being the most significant display digit. The L lines will provide the segment data with the following format:



The interconnect and flow chart are shown in *Figures 5-26* and *5-27*. The code is written independently and simply displays the data. In a real application, the routine would have to be merged with the main code. The routine provides both segment and digit interdigit blanking. A simple delay routine is used to control display ON/OFF time.

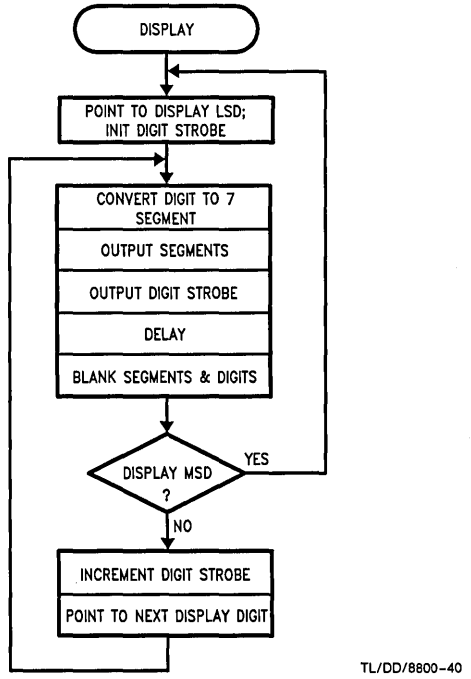


The RAM map for this routine is shown above. The display data is in BCD.



TL/DD/8800–39

FIGURE 5-26. Interconnect for Sample and Multiplexed Display Code



TL/DD/8800–40

FIGURE 5-27. Multiplexed Display Flow Chart

```
DISPLAY:  LBI   0,11     ; initialize digit strobe
          STII  1
          JSR   OUT      ; output first digit - LSD
          LBI   0,13
          JSR   OUT      ; second digit
          LBI   0,14
          JSR   OUT      ; third digit
          LBI   0,15
          JSR   OUT      ; fourth digit - MSD
          JP    DISPLAY
```

The subroutine OUT does most of the work:

```
OUT:      CLRA           ; set up address for table
          AISC  4
          LQID
          LDD   0,11     ; output digit strobe
          CAB
          OBD
          LEI   4        ; enable segment outputs
          LBI   0,11
          LD
          ADD            ; shift the strobe to next digit
          X
WAIT:     CLRA           ; delay time arbitrary for display
          SKT            ; on time
          JP    WAIT
          LBI   0,15     ; turn off the digits; all high
          OBD
          LEI   0        ; turn off the segments; L drives off
          RET            ; return for the next digit
```

The preceding routine uses a subroutine level. A routine that performs the same function but does not use a subroutine level is indicated below. As the RAM map indicates, an extra RAM digit is used in this implementation of the multiplexed display routine.

BD

	15	14	13	12	11	10
Br 0	DISPLAY MSD			DISPLAY LSD	DIGIT STROBE	DISPLAY POSITION

As before, the data is assumed to be in BCD.

```
DISPLAY   LBI   0,10     ; initialize display pointer and digit strobe
          STII  12
          STII  1
DSP1:     CLRA
          AISC  4        ; set up address for table
          LQID           ; loop up segments
          LDD   0,11     ; output digit strobe
          CAB
          OBD
          LEI   4        ; enable L to output segment data
          LBI   0,11     ; increment digit strobe (left shift)
          LD
          ADD
          X
WAIT:     CLRA           ; delay arbitrary for display ON time
          SKT
```

```
JP        WAIT
          LBI   0,10     ; increment display pointer
          LD
          AISC  1
          JP    DSP2
          JP    DISPLAY ; have outputted MSD, start over
DSP2:     X
          LD
          CAB
          JP    DSP1
```

This routine is completely equivalent to the preceding routine but does not have a subroutine call. Both routines use the following BCD to seven-segment code conversion table:

```
.=0140          ; set up table location - address
                ; starts at 140 hex
.WORD    03F    ; 0
.WORD    006    ; 1
.WORD    05B    ; 2
.WORD    04F    ; 3
.WORD    066    ; 4
.WORD    06D    ; 5
.WORD    07D    ; 6
.WORD    007    ; 7
.WORD    07F    ; 8
.WORD    067    ; 9
```

Both routines assume that the L drivers are off and that the digit strobes are high on entry to the routine. Some display types do not require both digit and segment blanking. If this is the case, the routines can be shortened by removing the unnecessary blanking code. Note that the routines do not alter the BCD data. Remember, also, that the LQID instruction uses a subroutine level on some COPS microcontrollers. Also note that the delay time included in the routine may not be necessary for some display types. In these cases, that code may be eliminated. The delay, if required at all, may be implemented in any convenient manner.

5.6.2 Peripheral Display Drivers

Several display drivers are available which are compatible with the COPS MICROWIRE and remove the burden of display control from the microcontroller to an inexpensive driver.

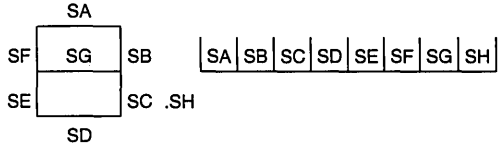
The COP470 and COP472

The COP470 is a four-digit multiplexed vacuum fluorescence display driver. The device is loaded with 32 bits of segment data and controls the display directly. Updating the display merely requires loading the new data. Note that any required code conversion must be performed by the microcontroller.

The COP472 is a similar device intended for use with a multiplexed (three backplane) liquid crystal display. The COP472 is a 4½ digit driver and can drive 36 segments of data. Again, any required code conversion must be done in the microcontroller.

Both the COP470 and COP472 may be cascaded to drive somewhat larger displays. The COP470 and COP472 are software compatible devices. Code can be written that works with either the COP470 or the COP472 either alone or cascaded. The four extra data bits in the COP472 correspond to brightness control in the COP470.

Both the COP470 and COP472 load data eight bits at a time. The format for the data is as follows:



SH for digit 1 is the first data bit shifted into the device. SA for digit four is the last data bit (i.e., 32nd data bit) shifted into the device. The segments are mapped into a standard numeric seven-segment plus decimal point display. There is, of course, no requirement that the display be configured in this manner.

The fifth and final group of eight bits sent to the device(s) is as follows:

C4	C3	C2	C1	SP4	SP3	SP2	SP1

SP1 is the first data bit sent in this group, C4 is the last bit sent.

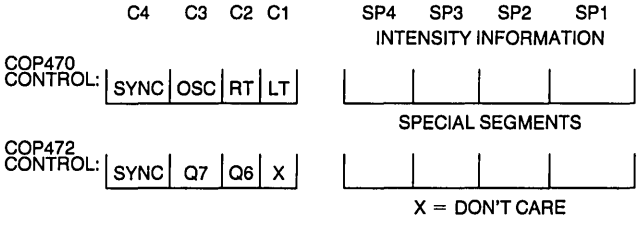
The COP470 and COP472 display drivers may be "cascaded" to provide more digits and "stacked" to provide more segments per digit. Both the COP472 and COP470 are code compatible devices even when they are used in expanded form.

Single COP470, COP472 Control Bits:

The control bits for the COP470 and COP472 are listed below in Table VII. These control bits were positioned to allow for common software operations.

The COP470 also contains four bits of intensity information which is in the same bit locations corresponding to the four special segments of the COP472. In code compatible routines, the four special segments of the LCD display will reflect the intensity information of the COP470. The control bits that enable code compatible operation with four-digit displays are given in Table VII.

TABLE VII. Control Bits



COP472	COP470	CONTROL BIT
SYNC	SYNC	0
Q7	0SC	0
Q6	RT	1
X	LT	1

Eight Digit

COP470 and COP472 devices are cascadable to obtain more digits of display. The control codes for a multiple device display driver configuration are listed in Table VIII.

TABLE VIII. Control Codes

		Control Codes		
COP472	**COP470**	**Initialize (Both Devices)**	**Master (Left Device)**	**Slave (Right Device)**
Sync	Sync	1	0	0
07	Osc	1	0	1
06	RT	1	0	1
X	LF	0	1	0
			X = Don't Care	

The sequence of operations to load a single COP470 or COP472 is as follows:

1. Turn $\overline{CS}$ low.
2. Clock in eight bits of data for digit 1.
3. Clock in eight bits of data for digit 2.
4. Clock in eight bits of data for digit 3.
5. Clock in eight bits of data for digit 4.
6. Clock in eight bits of data for special segments/brightness and the control function.

   0 0 1 1 SP4 SP3 SP2 SP1
7. Turn $\overline{CS}$ high.

$\overline{CS}$ may be turned high after any step. It is not necessary to continuously reload the control bits but they must be loaded at least once. If the special segments or brightness bits are changed, the control bits must be reloaded.

$\overline{CS}$ must toggle between writes. $\overline{CS}$ is the state that resets the internal counters in the device which controls data loading.

Typical code to write a single COP470 or COP472 is shown below. The look-up table is not shown but is obviously required. The routine is written as in-line code. It does the code conversion and writes to the display driver. The original values are destroyed in the operation. DO is arbitrarily chosen as a chip select for the device. Note that chip select is an essential connection for these devices. Chip select must toggle between accesses for proper operation. The data to be displayed is in locations 0,12 through 0,15. The special segments or brightness bits are in location 0,0.

```
DISPLAY:  LBI   0,12     ; point to first display data
          OBD            ; turn CS low (DO) to select drive
LOOP:     CLRA
          LQID           ; look up segment data
          CQMA           ; copy data from Q to M & A
          SC             ; set C to turn on SK
          XAS            ; output lower four bits of data
          NOP            ; delay
          NOP            ; delay
          LD             ; load A with upper four bits
          XAS            ; output four bits of data
          NOP            ; delay
          NOP            ; delay
          RC             ; reset C
          XAS            ; turn off SK clock
          XIS            ; increment B for next data
          JP    LOOP     ; skip this jump after last digit
          SC             ; set C
          LBI   0,0      ; address special segments or brightness
          LD             ; load into A
          XAS            ; output special segments or brightness
          NOP
          CLRA
          AISC  12       ; 12 to A=code for single chip operation
          XAS            ; output control bits
          NOP
          LBI   0,15     ; 15 to B to deselect the device
          RC            reset C
          XAS            ; turn off SK
          OBD            ; turn CS high (DO)
          .
          .
          .
```

This code works with either the COP470 or COP472.

The sequence to drive two COP470s or COP472s in an eight-digit display is outlined below. There is an initialization procedure required in order to set up the two devices properly. The control bits are different during the initialization sequence than they are during subsequent data loads. For the COP472s, this sequence sets up the left chip as the master and the right chip as the slave. For the COP470s, the left chip provides the oscillator for the right chip. The sequence is as follows:

1. Turn $\overline{CS}$ low to both devices.
2. Shift in 32 bits of data—slave's four digits for COP472, right four digits for COP470.
3. Shift in four bits of special segment/brightness data, a zero and three ones.

1	1	1	0	SP4	SSP3	SP2	SP1

This synchronizes and stops both chips. Both chips are expecting an external oscillator.

4. Turn $\overline{CS}$ high to both chips.
5. Turn $\overline{CS}$ low to left device—(master COP472, left COP470).
6. Shift in 32 bits of data for that device.
7. Shift in four bits of special segment/brightness data, a one and three zeroes.

0	0	0	1	SP4	SSP3	SP2	SP1

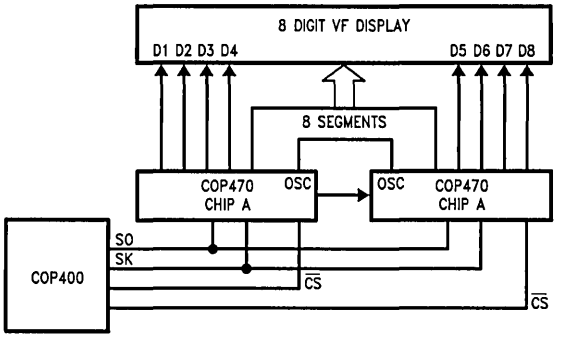
This set this device to internal oscillator and provides an oscillator output to the other device.

8. Turn $\overline{CS}$high.

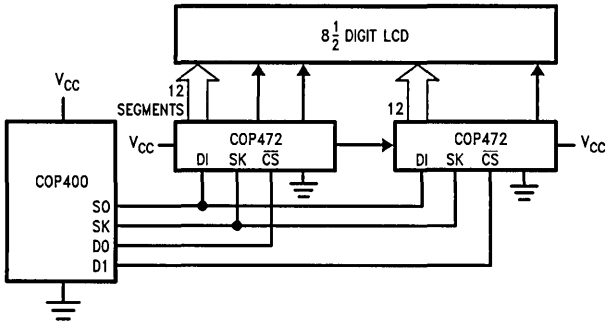
The chips are now synchronized and driving eight digits of display. New data is loaded in the normal manner. Care must be taken to keep the control bits in the proper state. For the master COP472 or left COP470, the control bits specified in Step 7 are the proper state. For the slave COP472 or right COP470, the following must be sent in every case except the initialization sequence:

0	1	1	0	SP4	SSP3	SP2	SP1

Figure 5-28 provides system diagrams for the dual COP470/COP472 systems.



TL/DD/8800-41



TL/DD/8800-42

FIGURE 5-28. Dual COP470/472 Systems

Typical code to write to the devices in this way is shown below. The display data for the slave (right) device is in register 0, digits 12 through 15. The display data for the master (left) device is in register 1, digits 12 through 15. Digit 0,0 contains special segment/brightness data for the slave. Digit 1,0 contains special segment/brightness data for the master. DO is used as the chip select for the master; D1 is the chip select for the slave. The code is again shown as in-line code.

Display Initialization Sequence:

```
INIT:   LBI   0,15
        OBD           ; turn both CSs high
        LEI   8       ; enable SO out of S.R.
        RC
        XAS           ; turn off SK clock
        LBI   3,15    ; use M(3,15) for control bits
        STII  7       ; store 7 to sync both chips
        LBI   0,12    ; set B to turn both CSs low
        JSR   OUT     ; call output subroutine
```

Main Display Sequence:

```
DISPLAY: LBI   3,15
         STII  8        ; set control bits for slave right devices
         LBI   0,13     ; set B to turn slave CS low
         JSR   OUT      ; output data from register 0
         LBI   3,15
         STII  6        ; set control bits for master left device
         LBI   1,14     ; set B to turn master CS low
         JSR   OUT      ; output data from register 1
```

Output Subroutine:

```
OUT:    OBD           ; output B to CSs
        CLRA
        AISC  12      ; 12 to A
        CAB           ; point to display digit (BD=12)
```

```
LOOP:   CLRA
        LQID        ; look up segment data
        CQMA        ; copy data from Q to M & A
        SC
        XAS         ; output lower four bits of data
        NOP         ; delay
        NOP         ; delay
        LD          ; load A with upper four bits
        XAS         ; output four bits of data
        NOP         ; delay
        NOP         ; delay
        RC          ; reset C
        XAS         ; turn off SK
        XIS         ; increment B for next display digit
        JP    LOOP  ; skip this jump after last digit
        SC          ; set C
        NOP
        LD          ; load special segments
        XAS         ; output special segments
        NOP
        LBI   3,15
        LD          ; load A
        XAS         ; output control bits
        NOP
        NOP
        RC
        XAS         ; turn off SK
        OBD         ; turn CSs high (BD=15)
        RET
```

The MM54XX and MM58XXX Series Display Drivers

The MM54XX and MM58XXX series drives are a family of status display drivers for vacuum fluorescent, liquid crystal, and LFD displays. All of these devices require a start bit and 35 data bits. All the devices are MICROWIRE compatible. Table IX indicates the present devices that comprise the MM54XX and MM58XXX series. The code here is applicable to all similar type devices. The MM54XX and MM58XXX devices are static segment drivers and must be loaded with the appropriate segment information.

TABLE IX. MM54XX Series Devices

MM58241 - Static Vacuum Fluorescent
MM58242 - Static Vacuum Fluorescent
MM58248 - Static Vacuum Fluorescent
MM58341 - Static Vacuum Fluorescent
MM58342 - Static Vacuum Fluorescent
MM58348 - Static Vacuum Fluorescent
MM5450 - Static LED
MM5451 - Static LED
MM5452 - Static Liquid Crystal
MM5453 - Static Liquid Crystal
MM5480 - Static LED (Smaller Package)
MM5481 - Static LED (Smaller Package)

Two basic output techniques can be used. The first approach is the same as that illustrated for the COP470 and COP472: turn the clock on and off and convert the number on the fly. This example will use G0 as the data enable control: G0 must go low to enable the device. The routine assumes G0 high, SO low, and SK low on entry. The look-up table is not shown.

```
DISPLAY:  CLRA          ; set up start bit
          AISC  1
          SC
          DGI   14      ; select the device
          XAS           ; turn on clock and send start bit
          RC
          CLRA
          NOP
          XAS           ; turn off the clock
          LBI   0,7     ; point to start of data
```

```
LOOP:   CLRA            ; set up table address
        LQID
        CQMA
        SC              ; send eight data bits
        XAS
        NOP
        NOP
        LD
        XAS
        NOP
        CLRA
        RC
        XAS
        LD
        XIS
        JP    LOOP
        OGI   15        ; deselect the device
        LEI   0         ; turn SO low
        RET
```

The other approach is to load a display bufffer with the segment data and then simply send all the information out in one burst of data. This technique can also be used with the COP470 and COP472. The following routine implements this procedure. Again, the table is not shown, and G0 is the data enable. The display output is the BCD number contained in locations 2,12 through 2,15. Register 0 will be used as the display output register. The segmented data will be placed in digits 0,7 through 0,15. Digit 0,15 will be loaded with 0s to fill out the required 35 data bits. The code is as follows:

```
DISPLAY: LBI   2,12     ; convert data to segment information
         CLRA           ; set up table address
         LQID
         LBI   0,7      ; save segments in register 0
         JSRP  INQ
         LBI   2,13
         LQID
         LBI   0,9
         JSRP  INQ
         LBI   2,14
         LQID
         LBI   0,11
         JSRP  INQ
         LBI   2,15
         LQID
         LBI   0,13
         JSRP  INQ
         STII  0        ; load 0s to 0,15
         LBI   0,7      ; point to first segment data
         SC             ; set C to turn on clock
         AISC  1        ; set up start bit
         LEI   8        ; enable shift register output
         XAS            ; send start bit
         JSR   DATOUT
         RET
```

The following subroutines are used:

```
INQ:    CQMA     DATOUT: LD
        XIS              XAS
        XIS              XIS
        CLRA             JP       DATOUT
        RET              RC               ; turn off clock
                         XAS
                         RET
```

Universal Display Loading Routine

Theory of Operation

The universal display driver loading routine both initializes and sends 32 data bits to the display drivers. In those devices with more than 32 data bits, the extra segments are not used. The routine is compatible with the COP470, COP472, MM54XX and MM58XXX series devices.

Associated with the COP470/COP472, MM54XX and MM58XXX series are two communication protocols. The COP470 and the COP472 accept data in blocks of eight bits and require an initialization procedure. The MM54XX and MM58XXX series requires a start bit and a block of 35 bits before data is latched in the output buffers. There exists a common block of 32 data bits between all these devices (less are bonded out on the MM5480 and MM5481) and this similarity makes it possible to create universal display load routine. The control bits for the COP470 and the COP472 are sent once upon initialization, and the start bit for the MM54XX and MM58XXX series is sent on the tail end of the data load routine every time it is called.

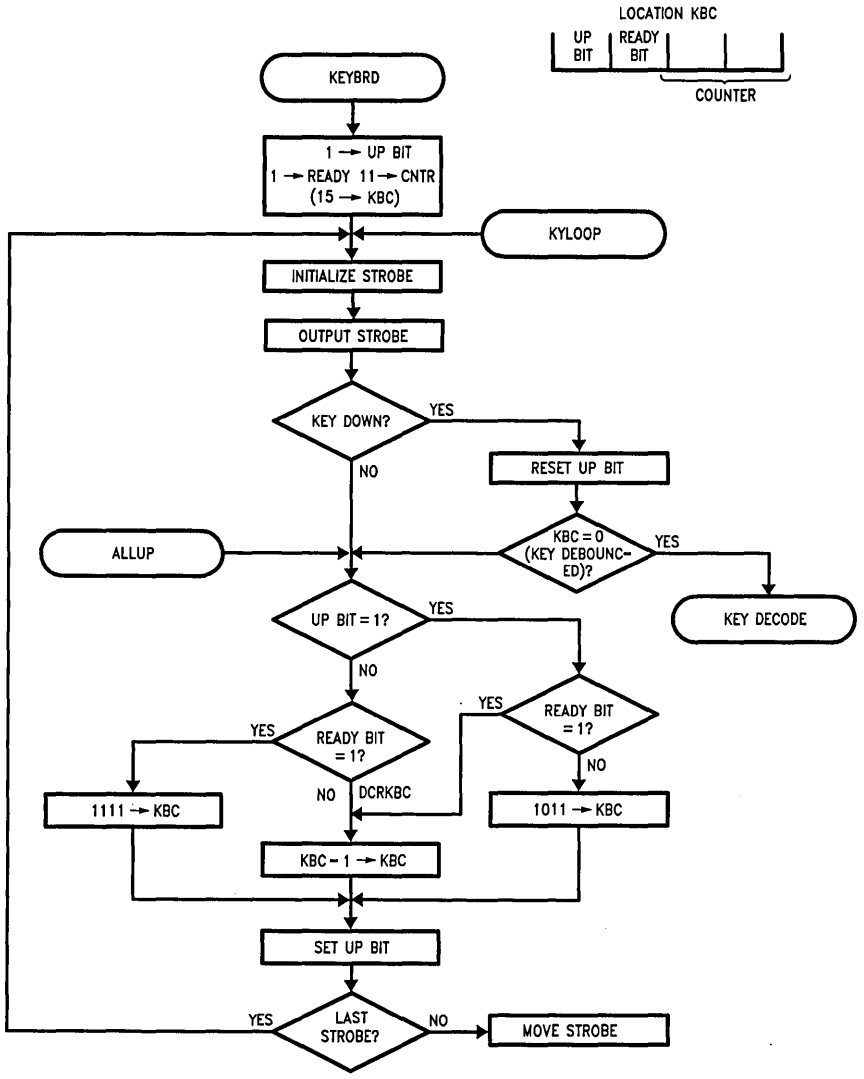
The COP470 and COP472 have a chip select which, upon a high to low transition, clears the input register and the internal counters which route the data and control bits to their ultimate positions. (See COP470, COP472 block diagrams.) Each of these devices accepts a serial data pattern and latches that serial stream in blocks of eight. For example, once initialized, the first digit may be changed, without affecting the other digits, by chip selecting and sending eight data bits. Data streams of less than eight bits, between chip selects or after a block of eight bits has been accepted, will be ignored. The initialization routine for the COP470 and COP472, which sends 44 bits, makes use of this type of operation; the last four bits are ignored.

The MM54XX and MM58XXX series displays, unlike the COP470 and COP472, have a data enable. This input to the device does not reset any counter and functions only as a data enable. This is to say that information contained within the display buffers and the input counter are not affected by the data enable signal. It is for this reason that the start bit for MM54XX and MM58XXX series devices is sent out at the tail end of each data output routine. Initially, the MM54XX and MM58XXX devices must be cleared and this is accomplished by clocking in more than 35 zeroes. In normal operation, the MM54XX and MM58XXX type devices are automatically cleared at power up due to SIO port power up state; SK as clock and SO as a logical zero, lasting much more than 36 cycles. In the universal display routine, the MM54XX and MM58XXX series devices will contain the COP470 and COP472 control codes along with a start bit in the first position. This must be cleared out by sending 35 zeroes and a new start bit. This will clock in 32 zeroes to the COP470 and COP472, and again the last four bits will be ignored in the COPS display drivers.

Now both display device types are initialized and data may be sent out in 36 bit blocks, first 32 data, next three zeroes, and the last bit a start bit. The first 32 segment outputs of the COP472 and MM54XX and MM58XXX series devices will correspond to the COP470s segment outputs.

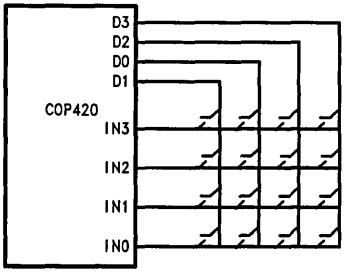
5.7 KEYBOARD SCAN

Reading a keyboard is a common requirement. The following routine is representative of a keyboard scan routine. The four D lines provide the strobes for the keyboard. The IN lines are the keyboard return lines. Thus, this routine is structured to read a 16-key keyboard arranged in a 4 by 4 matrix. A key is detected when one of the IN lines goes low. The strobes, D lines, are normally high and go low to strobe the keyboard. *Figure 5-29* is the flow chart for this routine. *Figure 5-30* is the interconnect. This routine uses two RAM digits: digit 0,15 for a debounce counter and digit 0,14 for temporary storage. The routine debounces the keys up and down.



TL/DD/8800-43

FIGURE 5-29. Keyboard Scan Flow Chart



TL/DD/8800-44

FIGURE 5-30. Interconnect for Key Scan Routine

```
        KBC   = 0,15
        KEYIN = 0,14
KEYBRD: LBI         KBC     ; initialize debounce counter
        STII        15
KYLOOP: LBI         1,14    ; set DO low, see if a key is down
        JSRP        SCAN
        JP          KEY0    ; key is down
        LBI         0,13    ; set Dl low and see if a key is down
        JSRP        SCAN
        JP          KEY1
        LBI         0,11    ; set D2 low and see if a key is down
        JSRP        SCAN
        JP          KEY2
        LBI         0,7     ; set D3 low and see if a key is down
        JSRP        SCAN
        JP          KEY3    ; if the routine falls through to this point
                            ; there is no key down on this scan,
                            ; or key not fully debounced
NOKEY:  LBI         KBC
        CBA                 ; put 15 to A
DBNCE:  SKMBZ       3       ; test up bit = 1
        JP          ALLUP   ; yes
        SKMBZ       2       ; up bit = 0, test ready bit
        JP          STR     ; 15 --> KBC else decrement KBC
DCRKBC: ADD                 ; remember A=15, so decrement KBC
STR:    X                   ; A --> KBC
        SMB         3       ; set up bit
        JP          KYLOOP
ALLUP:  SKMBZ       2       ; if ready bit=1, decrement KBC
        JP          DCRKBC
        STII        11      ; else, load KBC with 11
        JP          KYLOOP
KEY3:   These are the key decode positions; located KEYIN
KEY2:   contains IN line data; entry point defines strobe
KEY1:   line. The key is fully debounced if reach any of
KEY0:   these points.
```

The following is the key scan subroutine:

```
SCAN:   OBD         ; output key strobe
        ININ        ; read the return lines
        LBI   KEYIN
        COMP
        X           ; store key information
        LD          ; test if a key is down
        AISC  15
        RETSK       ; no key, return and skip
        CLRA        ; a key is down
        LBI   KBC
        RMB   3     ; reset key up bit
        SKE         ; if KBC is 0, key is fully debounced
        RETSK       ; not debounced yet
        OBD         ; key fully debounced, turn the strobes high
        LBI   KEYIN ; set up pointing to KEYIN for key decode
        RET
```

This is a simple keyboard routine. It is a variation on the routine provided in Section 5.3 of the *COPS Family User's Guide.* The routine continues to scan until a key is detected and fully debounced.

Appendix A

Data RAM in COP410L/411L/413L and COP410C/411C Devices

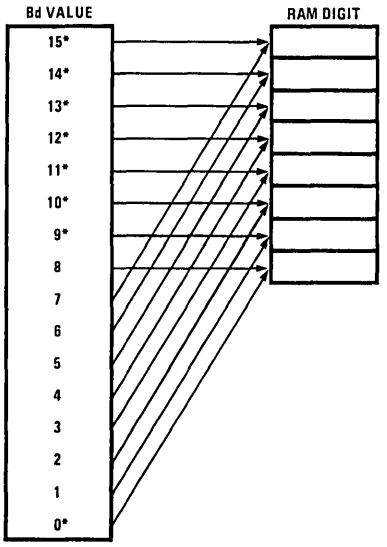
A.1 DATA RAM DESCRIPTION

All COPS microcontrollers except the COP410L, COP411L, COP410C, and COP411C have the data RAM matrix organized as a number of registers by 16 digits. The COP410C series devices mentioned above have the data RAM organized as 4 registers by 8 digits. This is significant because the Bd portion of the RAM address register B is still four bits wide. The D output port is still a four-bit port and it is loaded by Bd as in all COPS microcontrollers.

Physically, only the lower three bits of Bd address the digit portion of RAM. The upper bit is not connected to the RAM in any way. However, the XIS and XDS instructions work on the entire Bd register. The skip conditions on these instructions is the same as always. Bd will increment from 0 to 15. Thus each RAM digit in a COP410 series device is addressed by two values of Bd. Because of this characteristic, the programmer must exercise some care in the implementation of any routine which increments or decrements through the register, e.g., shift routines. The standard digit shift routines provided earlier could actually shift a COP410 register right or left two digits if the programmer started at one end of the register and relied on the XIS or XDS skip to exit the routine. The two shift routines provided below provide one method of circumventing the problem.

```
        LBI    0,9            LBI    0,9
        LD                    LD
        XDS                   XDS
        LD                    LD
LSHIFT: XIS          RSHIFT   XDS
        JP     LSHIFT         JP     RSHIFT
        RET                   RET
```

As written, these routines will shift register 0 left or right one digit. *Figure A-1* below illustrates the RAM mapping in COP410 series devices.



TL/DD/8800–45

FIGURE A-1. RAM Mapping

*Can be directly addressed by LBI instruction.

Appendix B

Devices with Subroutine Stack in RAM

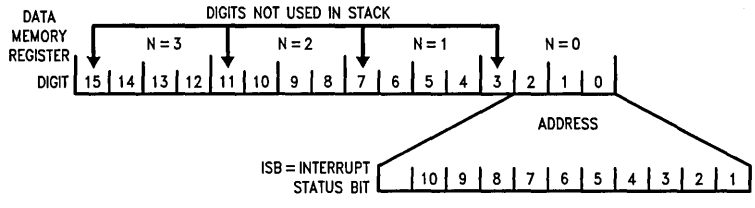
B.1 SUBROUTINE STACK IN RAM DESCRIPTION AND LOCATION

As mentioned earlier, a number of COPS microcontrollers have the subroutine stack in data RAM. In these devices the stack is assigned a specific location and does not, under any circumstances, go outside of the assigned area. It is not possible for the programmer to overflow the stack and destroy some data, although it is quite possible to overflow the stack. The only information lost if the stack overflows is some previous return address. The devices which have the stack in RAM and the location of the stack in the RAM is indicated below.

Device	Location
COP440/441/442 COP404	Stack in register 8

Note that the registers are numbered starting at 0. The register number is the Br address.

Figure B-1 is the strucrture for the stack in RAM. This organization is valid for all the devices with the subroutine stack in data RAM.



TL/DD/8800–46

FIGURE B-1. Stack Structure in RAM

Section 5
HPC Family

Section 5 Contents

The HPC Family

National offers the first 16-bit CMOS High Performance microController (HPC). The HPC family brings innovative solutions to the problem of making a single microcontroller family satisfy the demands of the highest performance in a wide range of end-user products.

And because the controllers are members of the same family, the user learns only **one** architecture, **one** programming language and maintains only **one** development system to produce products in a wide spectrum of applications.

With this 16-bit architecture implemented in microCMOS, the HPC is the highest speed microcontroller available and capable of operating at speeds beyond 17 MHz—with execution times of less than 240 nanoseconds. The HPC tackles products applications previously considered out of the range of any microcontroller. The HPC is designed for high-throughput computation. Its powerful "RISC-like" instruction set includes a 16-bit multiply and divide, single-byte jumps and calls, nine addressing modes, and versatile bit manipulation instructions.

Backing up this number-crunching power is a large set of on-board peripherals. This integration means that systems using the HPC will require fewer components, will be more reliable, and will be easier to test.

Designed with National's latest and scaleable VLSI microCMOS technology, the HPC will be an ever expanding family. With this microCMOS technology and the modularity of the core design allows for expansion in ROM, RAM and I/O. The latter includes on-board EEPROM and A/D as well as logic intensive functions and customer specific gate arrays. And because the HPC is a microCMOS device, it uses very little power, and will operate over a wide range of supply voltages and over a wide range of temperatures. With the low power dissipation of CMOS also comes higher reliability.

CORE: The foundation of all HPC family members is a common CPU or "core". This Von Neumann-architecture core consists of an arithmetic logic unit (ALU), six 16-bit working registers, eight interrupts, three 16-bit timers, control logic, watch-dog circuitry and a MICROWIRE/PLUS™ interface.

The six CPU registers (accumulator, program counter, stack pointer, two indirect address registers and the unique program looping register, K), the internal data paths and the ALU are all 16-bits wide. This allows direct addressing that simplifies interfacing to external hosts, memory, or peripherals. Presently the core allows for 64K direct addressing.

For more responsive and flexible system control every HPC controller has been provided with eight independently-vectored interrupts. There are five external interrupts: a reset interrupt, a non-maskable interrupt and three general purpose maskable interrupts. In addition, there are three internal interrupts: one for the timers and two for on-board peripherals.

An interrupt pending register allows the programmer to determine the priorities of these interrupts in software.

EFFICIENT INSTRUCTION SET: The HPC instruction set has been designed for maximum code efficiency since on chip memory allocated to the program, in single-chip systems, is always at a premium.

The instruction set features a wide variety of single-byte multi-function instructions with nine addressing modes for complete flexibility. Running at 17 MHz, single-byte instructions execute in only 240 ns and 16-bit multiply or divide takes just 7 μs. To accomplish the same task, benchmarks show that programs will require roughly half the memory used by other 16-bit or high-end 8-bit microcontrollers.

MEMORY: All memory, I/O, on-board peripherals and registers are memory-mapped into the controller's present 64k address space. This allows great flexibility in the control of data and makes it possible for program code to be executed from RAM as well as ROM. Program patches or board diagnostics can be quickly added or changed in RAM at any time. To accommodate nesting even for the most complicated subroutine, stack depth and location are defined by the programmer. Bit manipulation instructions are effective anywhere in memory including the registers and I/O. To make interfacing to the outside world even easier, the external data bus can be configured for either 8- or 16-bit operation.

The size of the on-board ROM varies for current family members from 0 to 8k bytes. Memory can be accessed in bytes or 16-bit words. The modularity of the HPC will allow future members of the family to have significantly different memory sizes as the target application requires it.

TIMERS: All members of the HPC family have at least three 16-bit timers as part of the "core", each timer with:

- external event counting
- input capture
- pulse width modulation
- selectable clock rates

MICROWIRE/PLUS Serial, Interface: MICROWIRE/PLUS is an enhancement of National's proven MICROWIRE™ three-wire serial interface and is being included on all new National controllers. This synchronous interface serially handles data by using an 8-bit, parallel-loaded shift register with separate input, output and synchronous clock. It allows the HPC to take advantage of the wide variety MICROWIRE peripherals now offered by National and other manufacturers, such as A/D converters, EEPROMs, display drivers, etc., and can be used for multiprocessing between 4, 8 and 16 Microcontrollers.

LOW-POWER OPERATION: Because the HPC utilized microCMOS technology it consumes very little power. Even at full speed (17 MHz) the HPC draws less than 20 mA. To further reduce power, the device can be operated at lower speeds and/or lower voltage.

For even more stringent power conservation, the HPC has two power-down modes: IDLE and HALT. Resulting power is as low as 125 microwatts.

DEVELOPMENT SUPPORT: National offers complete development system support for the HPC. See section 8 for a complete description of the hardware and software tools.

National Semiconductor Corporation

PRELIMINARY

HPC16040/HPC26040/HPC36040/HPC46040/HPC16030/ HPC36030/HPC46030 High-Performance Microcontrollers

General Description

The HPC16040 is a member of the HPC™ family of High Performance microControllers. Each member of the family has the same identical core CPU with a unique memory and I/O configuration to suit specific applications. Each part is fabricated in National's advanced microCMOS technology. This process combined with an advanced architecture provides fast, flexible I/O control, efficient data manipulation, and high speed computation.

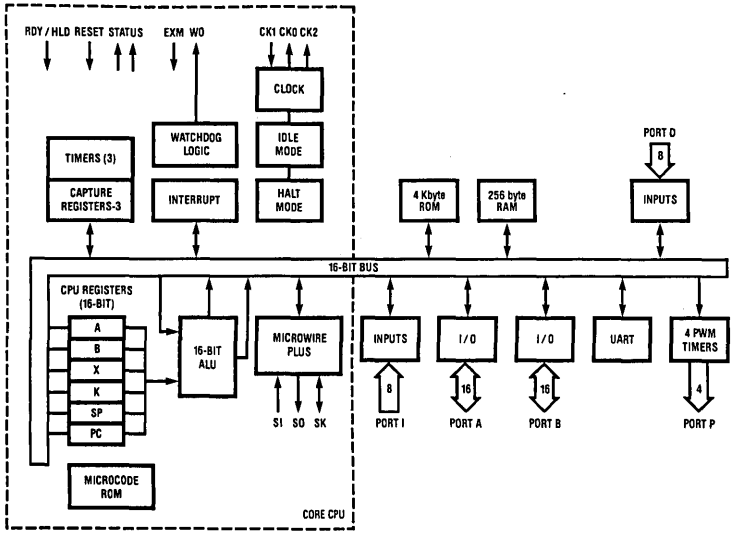
The HPC16040 is a complete microcomputer on a single chip. All system timing, internal logic, ROM, RAM, and I/O are provided on the chip to produce a cost effective solution for high performance applications. On-chip functions such as UART, eight 16-bit timers, vectored interrupts, WATCHDOG™ logic and MICROWIRE/PLUS™ provide a high level of system integration. The ability to address up to 64k bytes of external memory enables the HPC16040 to be used in powerful applications typically performed by microprocessors and expensive peripheral chips.

The microCMOS process results in very low current drain and enables the user to select the optimum speed/power product for his system. The IDLE and HALT modes provide further current savings. The HPC16040 is available in 68-pin PCC, LCC and PGA packages.

Features

- HPC family—core features:
  - — 16-bit architecture, both byte and word
  - — 16-bit data bus, ALU, and registers
  - — 64k bytes of external memory addressing
  - — FAST!—240 ns for register instructions when using 17.0 MHz clock
  - — High code efficiency—most instructions are single byte
  - — 16 x 16 multiply and 32 x 16 divide
  - — Eight vectored interrupt sources
  - — Four 16-bit timer/counters with 3 input capture registers and 4 synchronous outputs
  - — WATCHDOG logic monitors processor
  - — MICROWIRE/PLUS serial I/O interface
  - — CMOS—very low power with two power save modes: IDLE and HALT (2 mA, 250 μA—typ.)
- UART—full duplex, programmable baud rate
- Four additional 16-bit timer/counters with pulse width modulated outputs
- 52 general purpose I/O lines (memory mapped)
- 4k bytes of ROM, 256 bytes of RAM on chip
- ROMless versions available
- Wide voltage supply range: 3V to 5.5V
- Industrial (−40°C to +85°C) and military (−55°C to +125°C) temperature ranges

Block Diagram



TL/DD/8340-1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Total Allowable Source or Sink Current	100 mA
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 10 sec)	300°C
V_{CC} with Respect to GND	−0.5V to 7.0V
All Other Pins	(V_{CC} + 0.5)V to (GND − 0.5)V

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics

V_{CC} = 5.0V ±10% unless otherwise specified, T_A = 0°C to +70°C for HPC46040, −40°C to +85°C for HPC36040, −40°C to +105°C for HPC26040, −55°C to +125°C for HPC16040

Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
I_{CC_1}	Supply Current	V_{CC} = 5.0V, f_{in} = 17.0 MHz*		20		mA
		V_{CC} = 5.0V, f_{in} = 2.0 MHz		2.4		mA
I_{CC_2}	IDLE Mode Current	V_{CC} = 5.0V, f_{in} = 17.0 MHz, T_A = 25°C*		2		mA
		V_{CC} = 5.0V, f_{in} = 2.0 MHz, T_A = 25°C		0.2		mA
I_{CC_3}	HALT Mode Current	V_{CC} = 5.0V, f_{in} = 0 kHz, T_A = 25°C*		250		μA
		V_{CC} = 2.5V, f_{in} = 0 kHz, T_A = 25°C		150		μA
INPUT VOLTAGE LEVELS RESET, NMI, CKI AND WO (SCHMITT TRIGGERED)						
V_{IH_1}	Logic High		0.9 V_{CC}			V
V_{IL_1}	Logic Low				0.1 V_{CC}	V
ALL OTHER INPUTS						
V_{IH_2}	Logic High		0.7 V_{CC}			V
V_{IL_2}	Logic Low				0.2 V_{CC}	V
I_{LI}	Input Leakage Current				±1	μA
C_I	Input Capacitance			10		pF
C_{IO}	I/O Capacitance			20		pF
OUTPUT VOLTAGE LEVELS CMOS OPERATION						
V_{OH_1}	Logic High	I_{OH} = −10 μA	V_{CC} − 0.1			V
V_{OL_1}	Logic Low	I_{OH} = 10 μA			0.1	V
V_{OH_2}	Port A/B Drive, CK2 (A_0–A_{15}, B_{10}, B_{11}, B_{12}, B_{15})	I_{OH} = −7 mA, V_{CC} = 5.0V	2.4			V
V_{OL_2}		I_{OL} = 3 mA			0.4	V
V_{OH_3}	Other Port Pin Drive, WO (open drain) (B_0–B_9, B_{13}, B_{14}, P_0–P_3)	I_{OH} = −1.6 mA, V_{CC} = 5.0V	2.4			V
V_{OL_3}		I_{OL} = 0.5 mA			0.4	V
V_{OH_4}	ST1 and ST2 Drive	I_{OH} = −6 mA, V_{CC} = 5.0V	2.4			V
V_{OL_4}		I_{OL} = 1.6 mA			0.4	V
V_{RAM}	RAM Keep-Alive Voltage			2.5		V
I_{OZ}	TRI-STATE Leakage Current			±5		μA

Note: I_{CC_1}, I_{CC_2}, I_{CC_3} measured with no external drive (I_{OH} and I_{OL} = 0, I_{IH} and I_{IL} = 0).

5

AC Electrical Characteristics

V_{CC} = 5.0V ±10%, f_C = 17.0 MHz, T_A = 0°C to +70°C for HPC46040, −40°C to +85°C for HPC36040

Symbol	Parameter	Min	Typ	Max	Units
f_C = CKI freq.	Operating Frequency	2		17.0	MHz
$t_{C1} = 1/f_C$	Clock Period	59			ns
$t_C = 2/f_C$	Timing Cycle	118	120		ns
$t_{LL} = ½\,t_C - 9$	ALE Pulse Width	50	60		ns
$t_{ST} = ¼\,t_C - 6$	Address Valid to ALE Trailing Edge	23	30		ns
$t_{WAIT} = t_C = WS$	Wait State Period	118	120		ns
$f_{XIN} = \frac{1}{19\,t_{C1}}$	External Timer Input Frequency		877		kHz
$t_{XIN} = 3\,t_{C1}$	Pulse Width for Timer Inputs		180		ns
$f_{XOUT} = \frac{1}{16\,t_{C1}}$	Timer Output Frequency		1.04		MHz
$f_{MW} = \frac{1}{19\,t_{C1}}$	External MICROWIRE/PLUS Clock Input Frequency		877		kHz
$f_U = \frac{1}{19\,t_{C1}}$	External UART Clock Input Frequency		877		kHz
t_{DC1C2}	CK2 Delay from CK1			55	ns

Read Cycle Timing with One Wait State

Symbol	Parameter	Min	Typ	Max	Units
$t_{ARR} = ¼\,t_C - 5$	ALE Trailing Edge to $\overline{RD}$ Falling Edge	24			ns
$t_{RW} = ½\,t_C + WS - 10$	$\overline{RD}$ Pulse Width	167			ns
$t_{DR} = ¾\,t_C - 15$	Data Hold after Rising Edge of $\overline{RD}$	0		75	ns
$t_{ACC} = t_C + WS - 55$	Address Valid to Input Data Valid			181	ns
$t_{RD} = ½\,t_C + WS - 65$	$\overline{RD}$ Falling Edge to Data in Valid			112	ns
$t_{RDA} = t_C - 5$	$\overline{RD}$ Rising Edge to Address Valid	111			ns
$t_{VPR} = ¼\,t_C - 5$	Address Valid from ALE Trailing Edge Prior to $\overline{RD}$	24	35		ns

Write Cycle Timing with One Wait State

Symbol	Parameter	Min	Typ	Max	Units
$t_{ARW} = ½\,t_C - 5$	ALE Trailing Edge to $\overline{WR}$ Falling Edge	54			ns
$t_{WW} = ¾\,t_C + WS - 15$	$\overline{WR}$ Pulse Width	192			ns
$t_{HW} = ¼\,t_C - 5$	Data Hold after Trailing Edge of $\overline{WR}$	24			ns
$t_V = ½\,t_C + WS - 5$	Data Valid before Trailing Edge of $\overline{WR}$	172			ns
$t_{VPW} = ¼\,t_C + 20$	Address Valid from Trailing Edge Prior to $\overline{WR}$	50			ns

Note: Bus Output (Port A) C_L = 100 pF, CK2 Output C_L = 50 pF, other Outputs C_L = 80 pF.

Ready/Hold Timing f_C = 16.78 MHz with One Wait State

Symbol	Parameter	Min	Typ	Max	Units
$t_{DAR} = 1/4\ t_C + WS - 50$	Falling Edge of ALE to Falling Edge of $\overline{RDY}$		100		ns
$t_{RWP} = t_C$	$\overline{RDY}$ Pulse Width		120		ns
$t_{SALE} = 1/4\ t_C + 40$	Falling Edge of $\overline{HLD}$ to Rising Edge of ALE		70		ns
$t_{HWP} = t_C + 10$	$\overline{HLD}$ Pulse Width		130		ns
t_{HAD}	Rising Edge on $\overline{HLD}$ to Rising Edge on $\overline{HLDA}$		120		ns
$t_{HAE} = t_C + 100$	Falling Edge on $\overline{HLD}$ to Falling Edge on $\overline{HLDA}$		220	*	ns
$t_{BF} = t_C + 30$	Bus Float before Falling Edge on $\overline{HLDA}$		150		ns
$t_{BE} = 2\ t_C + 50$	Bus Enable from Rising Edge of $\overline{HLD}$		290		ns

*Note: t_{HAE} may be as long as ($3t_C$ + 4ws + $72t_C$ + 90) depending on which instruction is being executed, the addressing mode and number of wait states.

Status Timing f_C = 16.78 MHz

Symbol	Parameter	Min	Typ	Max	Units
$t_{SRS2} = 40 - (1/4\ t_C + 25)$	Setup Time for ST2 on Rising Edge of ALE		−15		ns
$t_{HRS2} = 3/4\ t_C - 15$	Hold Time for ST2 on Rising Edge of ALE		75		ns
$t_{SFS2} = 40 - (1/4\ t_C + 25)$	Setup Time for ST2 on Falling Edge of ALE		−15		ns
$t_{HFS2} = 3/4\ t_C - 15$	Hold Time for ST2 on Falling Edge of ALE		75		ns
t_{SFS1}	Setup Time for ST1 on Falling Edge of $\overline{RD}$		20		ns
$t_{HRS1} = 1/2\ t_C - 15$	Hold Time for ST1 on Rising Edge of $\overline{RD}$		45		ns

UPI Read/Write Timing

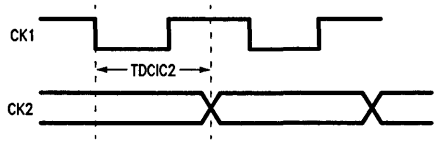
Symbol	Parameter	Min	Typ	Max	Units
t_{UAS}	Address Setup Time to Falling Edge of $\overline{UPIRD}$		5		ns
t_{UAH}	Address Hold Time from Rising Edge of $\overline{UPIRD}$		5		ns
t_{RPW}	$\overline{UPIRD}$ Pulse Width		100		ns
t_{OE}	$\overline{UPIRD}$ Falling Edge to Data Out Valid		60		ns
t_{OD}	End of $\overline{UPIRD}$ to Data Out Valid		35		ns
t_{DRDY}	$\overline{RDRDY}$ Delay from Trailing Edge of $\overline{UPIRD}$		70		ns
t_{WDW}	$\overline{UPIWR}$ Pulse Width		40		ns
t_{UDS}	Data in Valid before Trailing Edge of $\overline{UPIWR}$		10		ns
t_{UDH}	Data in Hold after Trailing Edge of $\overline{UPIWR}$		15		ns
t_A	$\overline{WRRDY}$ Delay from Trailing Edge of $\overline{UPIWR}$		70		ns

Note: Bus Output (Port A) C_L = 100 pF, CK2 Output C_L = 50 F, other Outputs C_L = 80 pF.

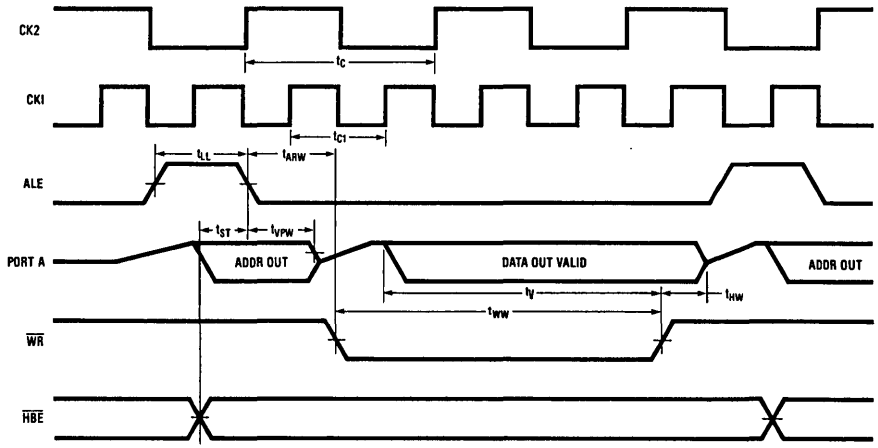
5

Timing Waveforms

CK2 Delay Timing Diagram

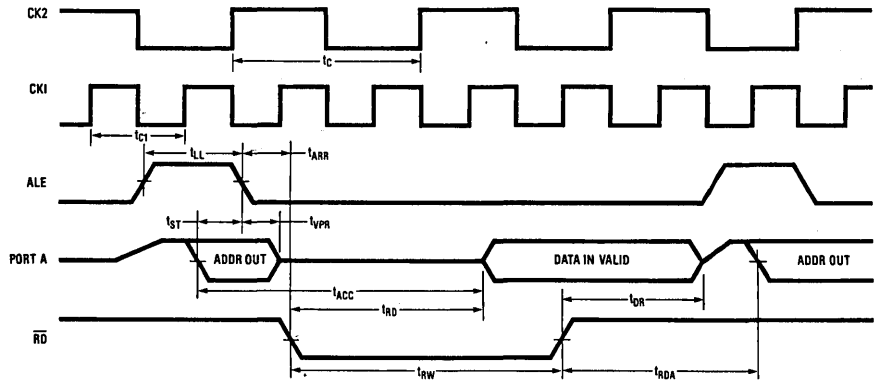


TL/DD/8340-29



TL/DD/8340-2

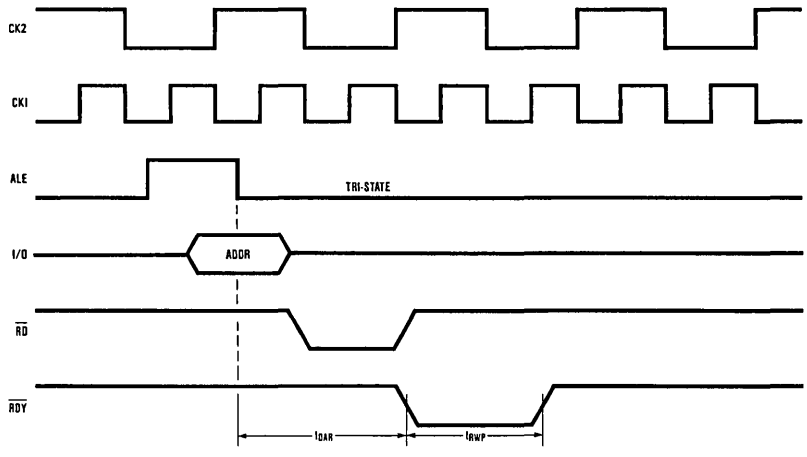
FIGURE 1. Write Cycle



TL/DD/8340-3

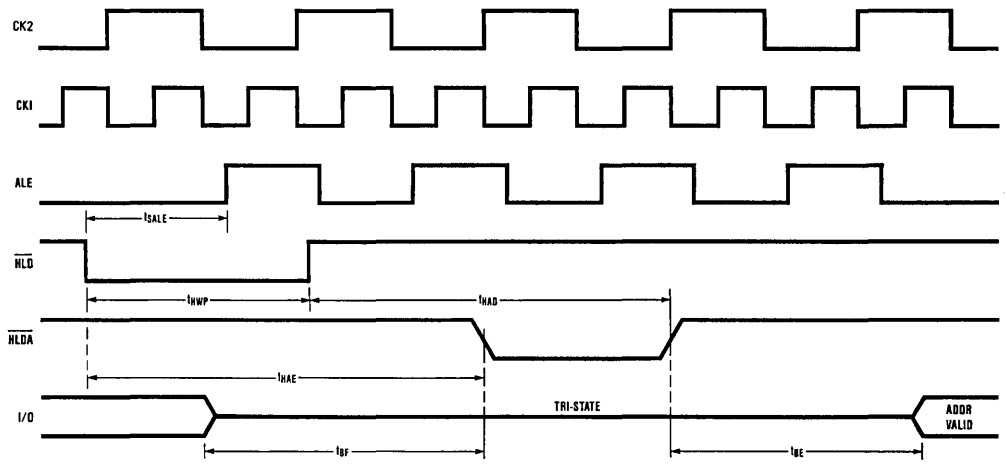
FIGURE 2. Read Cycle

Timing Waveforms (Continued)



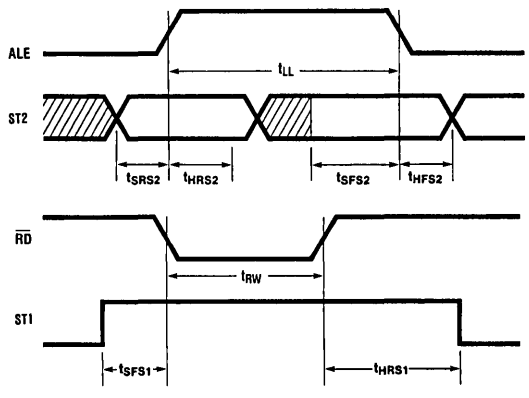
TL/DD/8340-4

FIGURE 3. Ready Mode Timing



TL/DD/8340-5

FIGURE 4. Hold Mode Timing



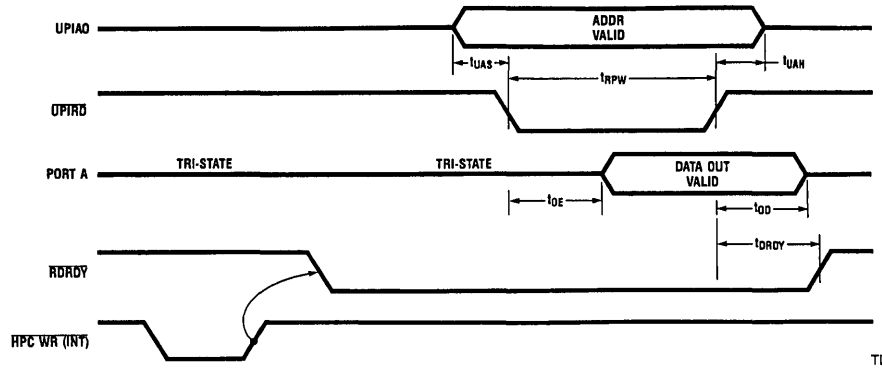
TL/DD/8340-6

TL/DD/8340-7

FIGURE 5. Status Timing

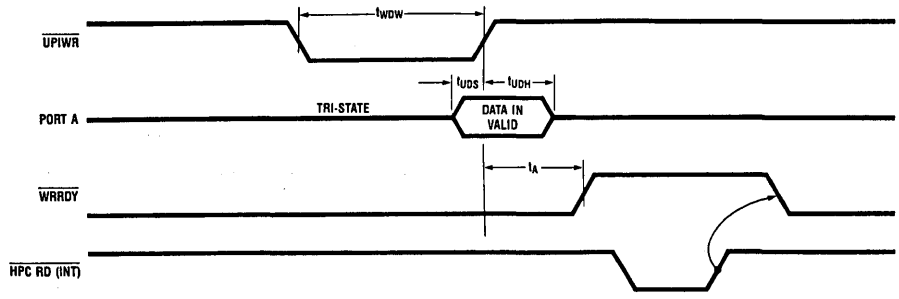
5

Timing Waveforms (Continued)



TL/DD/8340-8

FIGURE 6. UPI Read Timing



TL/DD/8340-9

FIGURE 7. UPI Write Timing

Pin Descriptions

The HPC16040 is available in 68-pin PCC and LCC packages, and a 48-pin ceramic DIP.

I/O PORTS

Port A is a 16-bit bidirectional I/O port with a data direction register to enable each separate pin to be individually defined as an input or output. When accessing external memory, port A is used as the multiplexed address/data bus.

Port B is a 16-bit port with 12 bits of bidirectional I/O similar in structure to Port A. Pins B10, B11, B12 and B15 are general purpose outputs only in this mode. Port B may also be configured via a 16-bit function register BFUN to individually allow each pin to have an alternate function.

B0:	TDX	UART Data Output
B1:		
B2:	CKX	UART Clock (Input or Output)
B3:	T2IO	Timer2 I/O Pin
B4:	T3IO	Timer3 I/O Pin
B5:	SO	MICROWIRE/PLUS Output
B6:	SK	MICROWIRE/PLUS Clock (Input or Output)
B7:	$\overline{\text{HLDA}}$	Hold Acknowledge Output
B8:	TS0	Timer Synchronous Output
B9:	TS1	Timer Synchronous Output
B10:	UA0	Address 0 Input for UPI Mode
B11:	$\overline{\text{WRRDY}}$	Write Ready Output for UPI Mode
B12:		
B13:	TS2	Timer Synchronous Output
B14:	TS3	Timer Synchronous Output
B15:	$\overline{\text{RDRDY}}$	Read Ready Output for UPI Mode

When accessing external memory, four bits of port B are used as follows:

B10:	ALE	Address Latch Enable Output
B11:	$\overline{\text{WR}}$	Write Output
B12:	$\overline{\text{HBE}}$	High Byte Enable Output/Input (sampled at reset)
B15:	$\overline{\text{RD}}$	Read Output

Port I is an 8-bit input port that can be read as general purpose inputs and is also used for the following functions:

I0:		
I1:	NMI	Nonmaskable Interrupt Input
I2:	INT2	Maskable Interrupt/Input Capture/$\overline{\text{URD}}$
I3:	INT3	Maskable Interrupt/Input Capture/$\overline{\text{UWR}}$
I4:	INT4	Maskable Interrupt/Input Capture
I5:	SI	MICROWIRE/PLUS Data Input
I6:	RDX	UART Data Input
I7:		

Port D is an 8-bit input port that can be used as general purpose digital inputs.

Port P is a 4-bit output port that can be used as general purpose data, or selected to be controlled by timers 4

Pin Descriptions (Continued)

through 7 in order to generate frequency, duty cycle and pulse width modulated outputs.

POWER SUPPLY PINS

V_{CC} Positive Power Supply (3V to 5.5V)

GND Ground for On-Chip Logic

DGND Ground for Output Buffers

Note: There are two electrically connected V_{CC} pins on the chip, GND and DGND are electrically isolated. Both V_{CC} pins and both ground pins must be used.

CLOCK PINS

CKI The Chip System Clock Input

CKO The Chip System Clock Output (inversion of CKI)

Pins CKI and CKO are usually connected across an external crystal.

CK2 Clock Output (CKI divided by 2)

OTHER PINS

WO This is an active low open drain output that signals an illegal situation has been detected by the Watch Dog logic.

ST1 Bus Cycle Status Output: indicates first opcode fetch.

ST2 Bus Cycle Status Output: indicates machine states (skip, interrupt and first instruction cycle).

RESET is an active low input that forces the chip to restart and sets the ports in a TRI-STATE® mode.

RDY/HLD has two uses, selected by a software bit. It's either a READY input to extend the bus cycle for slower memories, or a HOLD request input to put the bus in a high impedance state for DMA purposes.

NC (no connection) unused at this time.

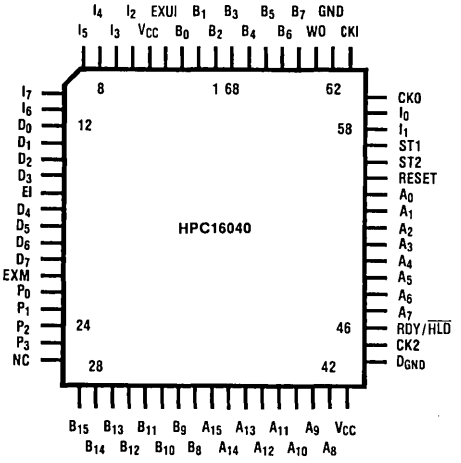
EXM External memory enable (active high) disables internal ROM and maps it to external memory.

EI External interrupt with vector address FFF1:FFF0. (active high)

EXUI External interrupt which is internally OR'ed with the UART interrupt with vector address FFF3:FFF2 (Active Low).

Connection Diagrams

Plastic and Leadless Chip Carriers



TL/DD/8340-10

Top View

Order Number HPC16040E or V
See NS Package Number E68B or V68A

Ports A and B

The highly flexible A and B ports are similarly structured. The Port A (see *Figure 9*), consists of a data register and a direction register. Port B (see *Figure 10*) has an alternate function register in addition to the data and direction registers. All the control registers are read/write registers.

The associated direction registers allow the port pins to be individually programmed as inputs or outputs. Port pins selected as inputs, are placed in a TRI-STATE mode by resetting corresponding bits in the direction register.

A write operation to a port pin configured as an input causes the value to be written into the data register, a read operation returns the value of the pin. Writing to port pins configured as outputs causes the pins to have the same value, reading the pins returns the value of the data register.

Primary and secondary functions are multiplexed onto Port B through the alternate function register (BFUN). The secondary functions are enabled by setting the corresponding bits in the BFUN register.

Operating Modes

To offer the user a variety of I/O and expanded memory options, the HPC16040 has four operating modes. The four modes are Single-Chip, Expanded, Single-Chip ROMless, and Expanded ROMless. The four modes are determined by the state of both the External Memory (EXM) pin and the External Access (EA) bit in the PSW Register. The HPC16040 System bus consists of port A and four bits of port B. Port A is defined as the address/data bus and the four bits of port B are referred to as the control bus.

SINGLE-CHIP MODE

In this mode, the HPC16040 functions as a self-contained microcomputer. It can address internal memory consisting of 256 bytes of RAM and 4 kbytes of ROM. All ports are configured as memory mapped I/O ports. The HPC16040 reads 8 bits or 16 bits of data from the ports, depending on whether a byte or word format instruction is used (see *Figure 11*). The EXM pin and EA bit of the PSW Register are both logic "0" during Single-Chip mode signifying that on-chip ROM is being addressed and the range is limited to 4k (see Table II).

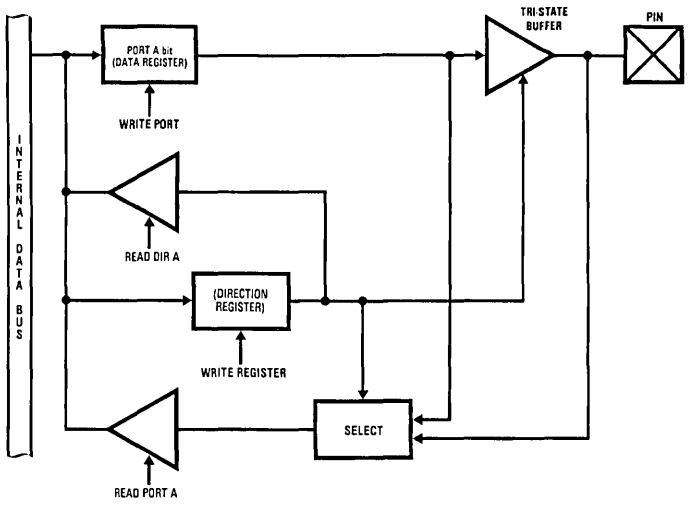
TABLE II. Operating Modes

External Memory Pin (EXM)	External Access Bit (EA)	Operation Mode
0	0	Single Chip
0	1	Expanded
1	0	Single Chip ROMless
1	1	Expanded ROMless

EXPANDED MODE

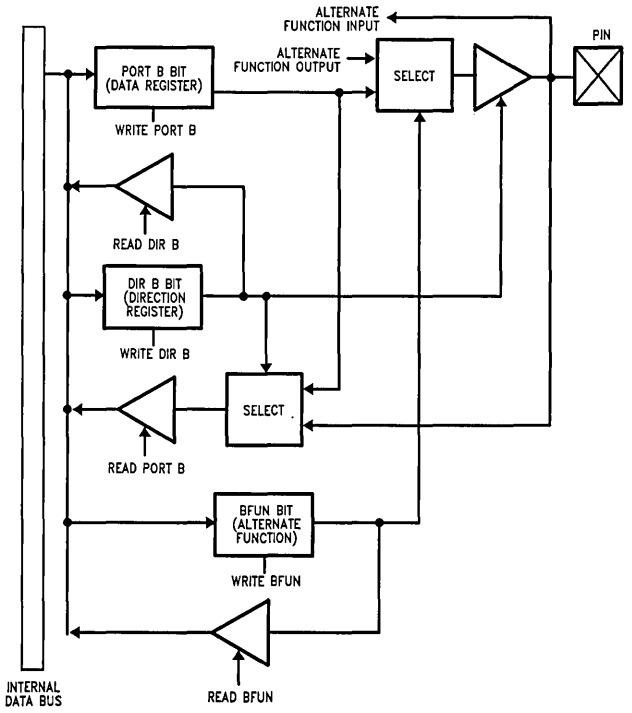
The Expanded mode (see *Figures 12* and *13*) is entered by setting the EA bit in the PSW Register. The HPC16040 can operate within the full 64 kbytes of address space. The 64 kbytes of addressable memory includes all on-chip memory because the EXM pin is grounded during this mode. The external memory may be any combination of RAM and ROM. External memory can be accessed with the data bus defined as either 8 bits wide or 16 bits wide. The System bus may be configured in the 8-bit mode by pulling the $\overline{\text{HBE}}$ pin high at reset. Upon entering the expanded mode, port A

Operating Modes (Continued)



TL/DD/8340–13

FIGURE 9. Port A: I/O Structure

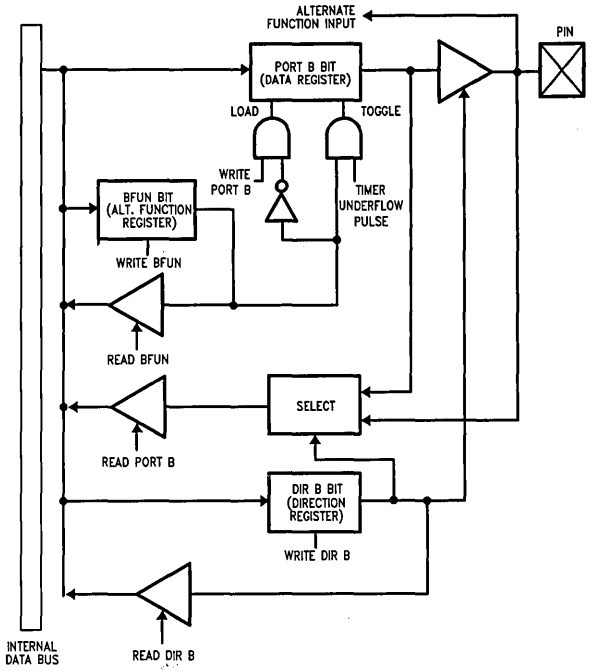


TL/DD/8340–32

FIGURE 10a. Structure of Port B Pins B0, B1, B2, B5, B6 and B7 (Typical Pins)

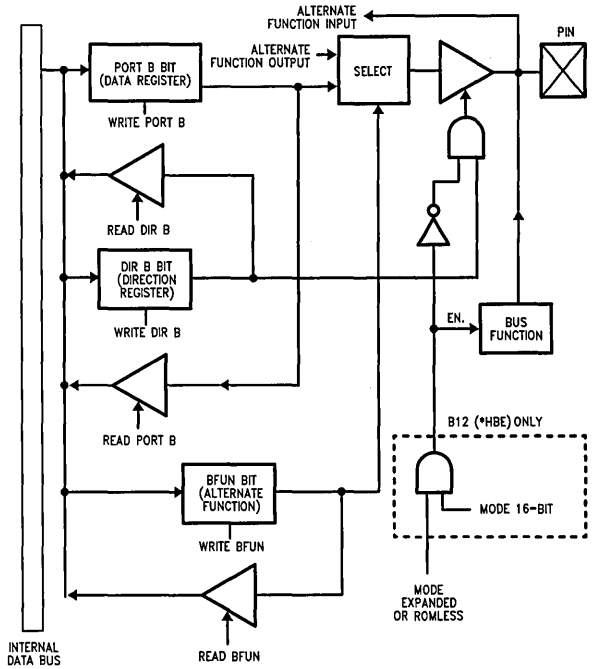
5

Operating Modes (Continued)



TL/DD/8340–33

FIGURE 10b. Structure of Port B Pins B3, B4, B8, B9, B13 and B14 (Timer Synchronous Pins)



TL/DD/8340–34

FIGURE 10c. Structure of Port B Pins B10, B11, B12 and B15 (Pins with Bus Control Roles)

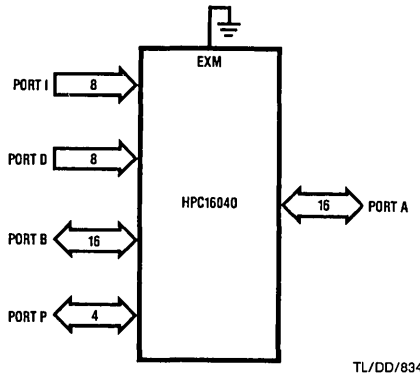
Operating Modes (Continued)

becomes the Address/Data bus. Four bits of port B become ALE, $\overline{WR}$, $\overline{HBE}$ and $\overline{RD}$ signals. The $\overline{RD}$ and $\overline{WR}$ signals are generated only if the selected address is off-chip. The $\overline{HBE}$ is generated only in the 16-bit bus configuration.

ROMless MODES

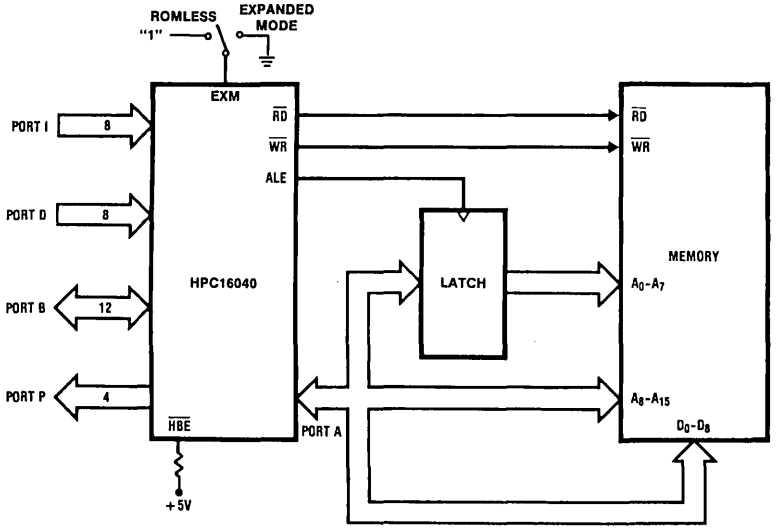
There are two ROMless modes; Single-Chip ROMless and Expanded ROMless. Both ROMless modes are entered by pulling the EXM pin high, (see *Figure 12* and *13*). The EA bit in the PSW Register determines whether the HPC16040 addresses the Single-Chip memory range of 4 kbytes or the Expanded range of 64 kbytes, (see Table II for this information). In both ROMless modes, the HPC16040 continues to use the internal 256 bytes of RAM. The external 4k or 64k of addressed memory may be any combination of RAM and ROM. The address space corresponding to internal ROM is mapped into external memory.

Note: The HPC16040 uses 16-bit words for stack memory. Therefore, when using the 8-bit mode, User's Stack must be in internal RAM.



TL/DD/8340-15

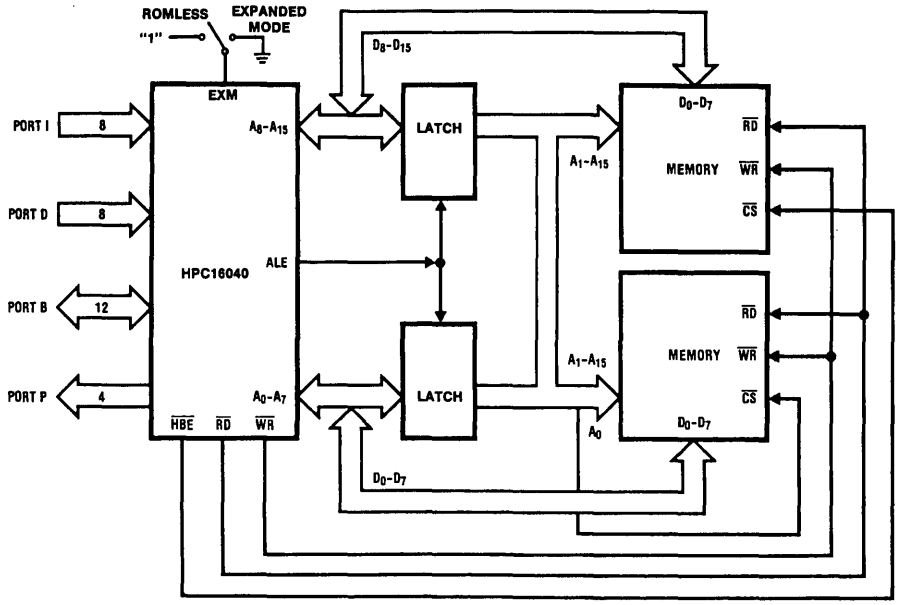
FIGURE 11. Single-Chip Mode



TL/DD/8340-16

FIGURE 12. 8-Bit External Memory

Operating Modes (Continued)



TL/DD/8340–17

FIGURE 13. 16-Bit External Memory

Wait States

The HPC16040 provides four software selectable Wait States that allow access to slower memories. The Wait States are selected by the state of two bits in the PSW register. Additionally, the RDY input may be used to extend the instruction cycle, allowing the user to interface with slow memories and peripherals.

Power Save Modes

Two power saving modes are available on the HPC16040: HALT and IDLE. In the HALT mode, all processor activities are stopped. In the IDLE mode, the on-board oscillator and timer T0 are active but all other processor activities are stopped. In either mode, all on-board RAM, registers and I/O are unaffected.

HALT MODE

The HPC16040 is placed in the HALT mode under software control by setting bits in the PSW. All processor activities, including the clock and timers, are stopped. In the HALT mode, power requirements for the HPC16040 are minimal and the applied voltage (V_{CC}) may be decreased without altering the state of the machine. There are two ways of exiting the HALT mode: via the RESET or the NMI. The RESET input reinitializes the processor. Use of the NMI input will generate a vectored interrupt and resume operation from that point with no initialization. The HALT mode can be enabled or disabled by means of a control register HALT enable. To prevent accidental use of the HALT mode the HALT enable register can be modified only once.

IDLE MODE

The HPC16040 is placed in the IDLE mode through the PSW. In this mode, all processor activity, except the on-board oscillator and Timer T0, is stopped. External interrupt (EXUI) is shared with the UART interrupt. This interrupt is level-low sensitive. To select this interrupt disable the ERI and ETI UART interrupt bits in the ENUI register. To select the UART interrupt leave this pin floating or tie it high. As with the HALT mode, the processor is returned to full operation by the RESET or NMI inputs, but without waiting for oscillator stabilization. A timer T0 overflow will also cause the HPC16040 to resume normal operation.

HPC16040 Interrupts

Complex interrupt handling is easily accomplished by the HPC16040's vectored interrupt scheme. There are eight possible interrupt sources as shown in Table III.

TABLE III. Interrupts

Vector Address	Interrupt Source	Arbitration Ranking
$FFFF:FFFE	RESET	0
$FFFD:FFFC	Nonmaskable external on rising edge of I1 pin	1
$FFFB:FFFA	External interrupt on I2 pin	2
$FFF9:FFF8	External interrupt on I3 pin	3
$FFF7:FFF6	External interrupt on I4 pin	4
$FFF5:FFF4	Overflow on internal timers	5
$FFF3:FFF2	Internal on the UART transmit/receive complete or external on EXUI	6
$FFF1:FFF0	External interrupt on EI pin	7

Interrupt Arbitration

The HPC16040 contains arbitration logic to determine which interrupt will be serviced first if two or more interrupts occur simultaneously. The arbitration ranking is given in Table III. The interrupt on Reset has the highest rank and is serviced first.

Interrupt Processing

Interrupts are serviced after the current instruction is completed except for the RESET, which is serviced immediately.

RESET and EXUI are level-LOW-sensitive interrupts and EI is level-HIGH-sensitive. All other interrupts are edge-sensitive. NMI is positive-edge sensitive. The external interrupts on I2, I3 and I4 can be software selected to be rising or falling edge.

Interrupt Control Registers

The HPC16040 allows the various interrupt sources and conditions to be programmed. This is done through the various control registers. A brief description of the different control registers is given below.

INTERRUPT ENABLE REGISTER (ENIR)

RESET and the External Interrupt on I1 are non-maskable interrupts. The other interrupts can be individually enabled or disabled. Additionally, a Global Interrupt Enable Bit in the ENIR Register allows the Maskable interrupts to be collectively enabled or disabled. Thus, in order for a particular interrupt to be serviced, both the individual enable bit and the Global Interrupt bit (GIE) have to be set.

INTERRUPT PENDING REGISTER (IRPD)

The IRPD register contains a bit allocated for each interrupt vector. The occurrence of specified interrupt trigger conditions causes the appropriate bit to be set. There is no indication of the order in which the interrupts have been received. The bits are set independently of the fact that the interrupts may be disabled. IRPD is a Read/Write register. The bits corresponding to the maskable, external interrupts are normally cleared by the HPC16040 after servicing the interrupts.

For the interrupts from the on-board peripherals, the user has the responsibility of resetting the interrupt pending flags through software.

The NMI bit is read only and I2, I3, and I4 are designed as to only allow a zero to be written to the pending bit (writing a one has no affect). A LOAD IMMEDIATE instruction is to be the only instruction used to clear a bit or bits in the IRPD register. This allows a mask to be used, thus ensuring that the other pending bits are not affected.

INTERRUPT CONDITION REGISTER (IRCD)

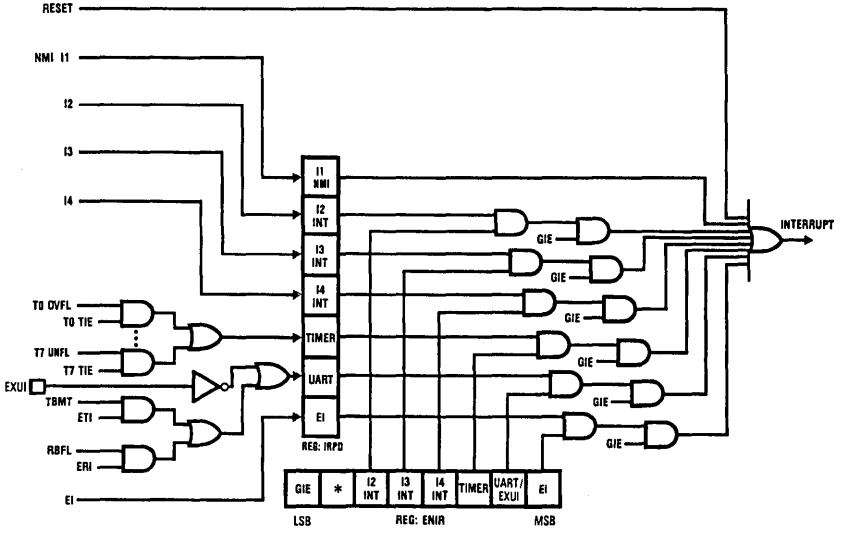
Three bits of the register select the input polarity of the external interrupt on I2, I3, and I4.

Servicing the Interrupts

The Interrupt, once acknowledged, pushes the program counter (PC) onto the stack thus incrementing the stack pointer (SP) twice. The Global Interrupt Enable bit (GIE) is copied into the CGIE bit of the PSW register; it is then reset, thus disabling further interrupts. The program counter is loaded with the contents of the memory at the vector address and the processor resumes operation at this point. At the end of the interrupt service routine, the user does a RETI instruction to pop the stack and re-enable interrupts if the CGIE bit is set, or RET to just pop the stack if the CGIE bit is clear, and then returns to the main program. The GIE bit can be set in the interrupt service routine to nest interrupts if desired. *Figure 14* shows the Interrupt Enable Logic.

Reset

The RESET input initializes the processor and sets ports A, B, and P in the TRI-STATE condition. RESET is an active-low Schmitt trigger input. The processor vectors to FFFF:FFFE and resumes operation at the address contained at that memory location (which must correspond to an on board location).



TL/DD/8340–18

FIGURE 14. Interrupt Enable Logic

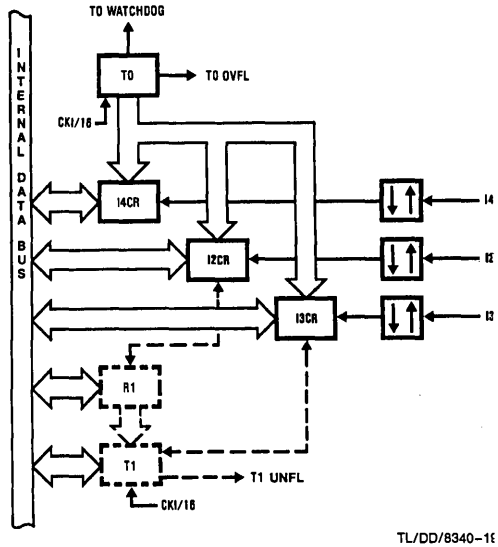
Timer Overview

The HPC16040 contains a powerful set of flexible timers enabling the HPC16040 to perform extensive timer functions; not usually associated with microcontrollers.

The HPC16040 contains eight 16-bit timers. Each timer has an associated 16-bit register. Timer T0 is a free-running timer, counting up at a fixed CKI/16 (Clock Input/16) rate. It is used for Watch Dog logic, high speed event capture, and to exit from the IDLE mode. Consequently, it cannot be stopped or written to under software control. Timer T0 permits precise measurements by means of the capture registers I2CR, I3CR, and I4CR. A control bit in the register TMMODE configures timer T1 and its associated register R1 as capture registers I3CR and I2CR. The capture registers I2CR, I3CR, and I4CR respectively, record the value of timer T0 when specific events occur on the interrupt pins I2, I3, and I4. The control register IRCD programs the capture registers to trigger on either a rising edge or a falling edge of its respective input. The specified edge can also be programmed to generate an interrupt (see *Figure 15*).

The timers T2 and T3 have selectable clock rates. The clock input to these two timers may be selected from the following two sources: an external pin, or derived internally by dividing the clock input. Timer T2 has additional capability of being clocked by the timer T3 underflow. This allows the user to cascade timers T3 and T2 into a 32-bit timer/counter. The control register DIVBY programs the clock input to timers T2 and T3 (see *Figure 16*).

The timers T1 through T7 in conjunction with their registers form Timer-Register pairs. The registers hold the pulse duration values. All the Timer-Register pairs can be read from or written to. Each timer can be started or stopped under software control. Once enabled, the timers count down, and upon underflow, the contents of its associated register are automatically loaded into the timer.



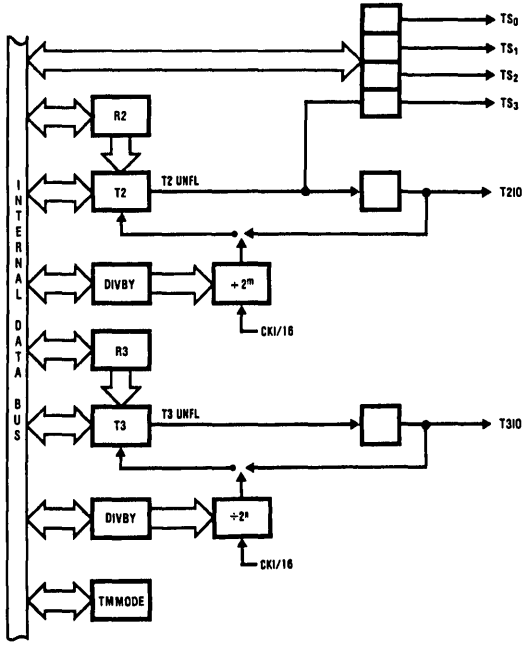
TL/DD/8340-1

FIGURE 15. Timers T0–T1 Block

SYNCHRONOUS OUTPUTS

The flexible timer structure of the HPC16040 simplifies pulse generation and measurement. There are four synchronous timer outputs (TS0 through TS3) that work in conjunction with the timer T2. The synchronous timer outputs can be used either as regular outputs or individually programmed to toggle on timer T2 underflows (see *Figure 16*).

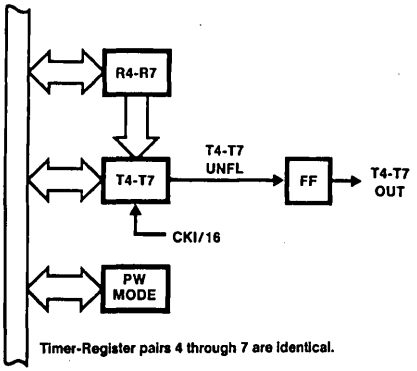
Timer/register pairs 4–7 form four identical units which can generate synchronous outputs on port P (see *Figure 17*).



TL/DD/8340-20

FIGURE 16. Timers T2–T3 Block

Timer Overview (Continued)



TL/DD/8340-21

FIGURE 17. Timers T4–T7 Block

Timer Registers

There are four control registers that program the timers. The divide by (DIVBY) register programs the clock input to timers T2 and T3. The timer mode register (TMMODE) contains control bits to start and stop timers T1 through T3. It also contains bits to latch and enable interrupts from timers T0 through T3. The control register PWMODE similarly programs the pulse width timers T4 through T7 by allowing them to be started, stopped, and to latch and enable interrupts on underflows. The PORTP register contains bits to preset the outputs and enable the synchronous timer output functions.

Timer Applications

The use of Pulse Width Timers for the generation of various waveforms is easily accomplished by the HPC16040.

Frequencies can be generated by using the timer/register pairs. A square wave is generated when the register value is a constant. The duty cycle can be controlled simply by changing the register value.

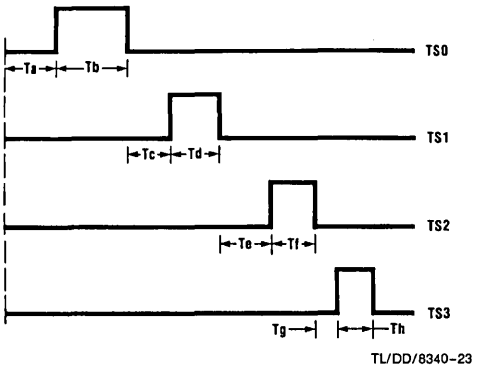
TL/DD/8340-22

FIGURE 18. Square Wave Frequency Generation

Synchronous outputs based on Timer T2 can be generated on the 4 outputs TS0–TS3. Each output can be individually programmed to toggle on T2 underflow. Register R2 contains the time delay between events. *Figure 19* is an example of synchronous pulse train generation.

Watch Dog Logic

The Watch Dog Logic monitors the operations taking place and signals upon the occurrence of any illegal activity. The illegal conditions that trigger the Watch Dog logic are potentially infinite loops and illegal addresses. Should the Watch Dog register not be written to before Timer T0 overflows



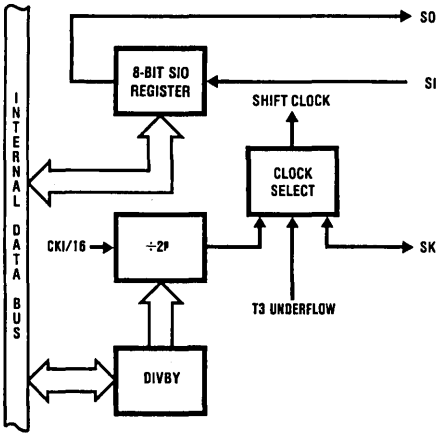
TL/DD/8340-23

FIGURE 19. Synchronous Pulse Generation

twice, or more often than once every 4096 counts, an infinite loop condition is assumed to have occurred. An illegal condition also occurs when the processor generates an off-chip address when in the Single-Chip mode. The illegal condition forces the Watch Out (WO) pin low. The WO pin is an open drain output and can be connected to the RESET or NMI inputs or to the users external logic.

MICROWIRE/PLUS

MICROWIRE/PLUS is used for synchronous serial data communications (see *Figure 20*). MICROWIRE/PLUS has an 8-bit parallel-loaded, serial shift register using SI as the input and SO as the output. SK is the clock for the serial shift register (SIO). The SK clock signal can be provided by an internal or external source. The internal clock rate is programmable by the DIVBY register. A DONE flag indicates when the data shift is completed.



TL/DD/8340-24

FIGURE 20. MICROWIRE/PLUS

The MICROWIRE/PLUS capability enables it to interface with any of National Semiconductor's MICROWIRE peripherals (i.e., A/D converters, display drivers, EEPROMs).

5

MICROWIRE/PLUS Operation

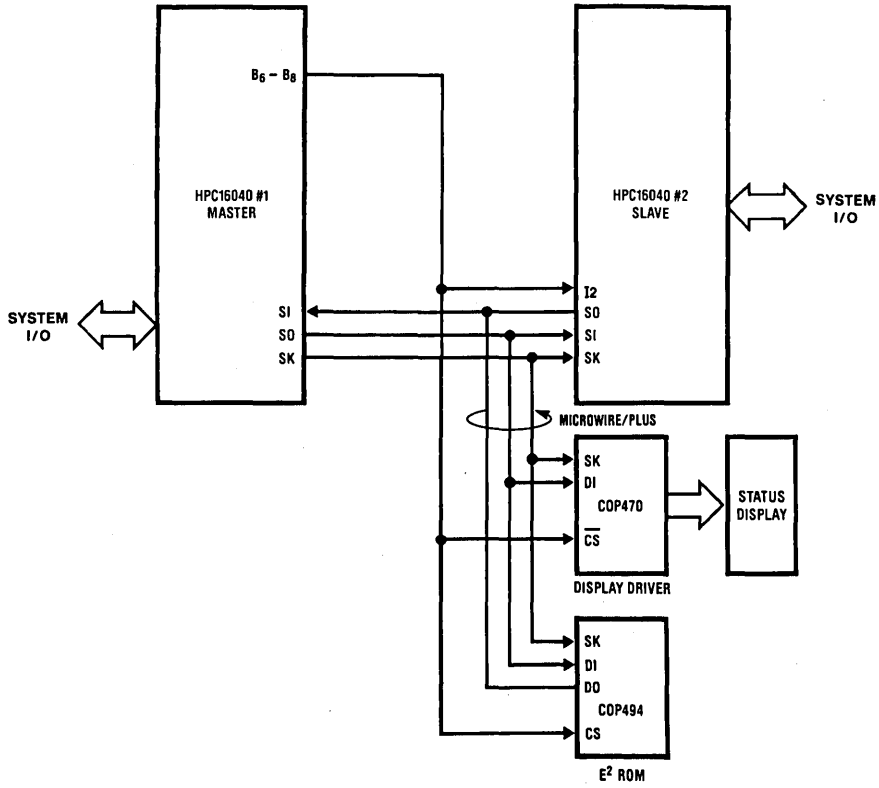
The HPC16040 can enter the MICROWIRE/PLUS mode as the master or a slave. A control bit in the IRCD register determines whether the HPC16040 is the master or slave. The shift clock is generated when the HPC16040 is configured as a master. An externally generated shift clock on the SK pin is used when the HPC16040 is configured as a slave. When the HPC16040 is a master, the DIVBY register programs the frequency of the SK clock. The DIVBY register allows the SK clock frequency to be programmed in 15 selectable steps from 64 Hz to 1 MHz with CKI at 16.0 MHz.

The contents of the SIO register may be accessed through any of the memory access instructions. Data waiting to be transmitted in the SIO register is clocked out on the falling edge of the SK clock. Serial data on the SI pin is clocked in on the rising edge of the SK clock.

MICROWIRE/PLUS Application

Figure 21 illustrates a MICROWIRE/PLUS arrangement for an automotive application. The microcontroller-based system could be used to interface to an instrument cluster and various parts of the automobile. The diagram shows two HPC16040 microcontrollers interconnected to other MICROWIRE peripherals. HPC16040 #1 is set up as the master and initiates all data transfers. HPC16040 #2 is set up as a slave answering to the master.

The master microcontroller interfaces the operator with the system and could also manage the instrument cluster in an automotive application. Information is visually presented to the operator by means of a VF display controlled by the COP470 display driver. The data to be displayed is sent serially to the COP470 over the MICROWIRE/PLUS link. Data such as accumulated mileage could be stored and retrieved from the EEPROM COP494. The slave HPC16040 could be used as a fuel injection processor and generate timing signals required to operate the fuel valves. The master processor could be used to periodically send updated values to the slave via the MICROWIRE/PLUS link. To speed up the response, chip select logic is implemented by connecting an output from the master to the external interrupt input on the slave.



TL/DD/8340–25

FIGURE 21. MICROWIRE/PLUS Application

HPC16040 UART

The HPC16040 contains a software programmable UART. The UART (see *Figure 22*) consists of a transmit shift register, a receiver shift register and five addressable registers, as follows: a transmit buffer register (TBUF), a receiver buffer register (RBUF), a UART control and status register (ENU), a UART receive control and status register (ENUR) and a UART interrupt and clock source register (ENUI). The ENU register contains flags for transmit and receive functions; this register also determines the length of the data frame (8 or 9 bits) and the value of the ninth bit in transmission. The ENUR register flags framing and data overrun errors while the UART is receiving. Other functions of the ENUR register include saving the ninth bit received in the data frame and enabling or disabling the UART's Wake-up Mode of operation. The determination of an internal or external clock source is done by the ENUI register, as well as selecting the number of stop bits and enabling or disabling transmit and receive interrupts.

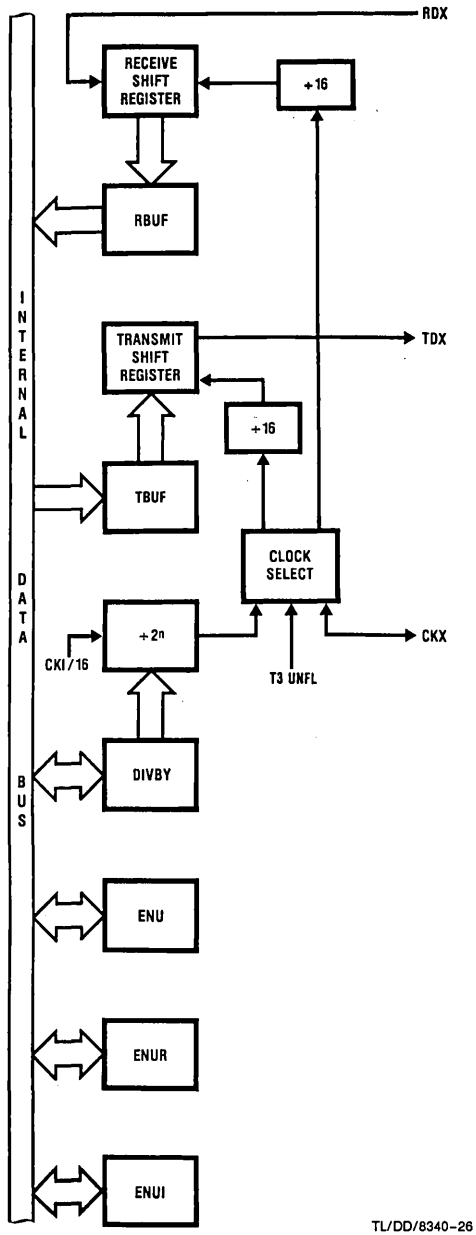
The baud rate clock for the Receiver and Transmitter can be selected for either an internal or external source using two bits in the ENUI register. The internal baud rate is programmed by the DIVBY register. The baud rate may be selected from a range of 8 Hz to 64 kHz in binary steps or T3 underflow. By selecting a 9.83 MHz crystal, all standard baud rates from 75 baud to 38.4 kBaud can be generated. The external baud clock source comes from the CKX pin. The Transmitter and Receiver can be run at different rates by selecting one to operate from the internal clock and the other from an external source.

The HPC16040 UART supports two data formats. The first format for data transmission consists of one start bit, eight data bits and one or two stop bits. The second data format for transmission consists of one start bit, nine data bits, and one or two stop bits. Receiving formats differ from transmission only in that the Receiver always requires only one stop bit in a data frame.

UART Wake-up Mode

The HPC16040 UART features a Wake-up Mode of operation. This mode of operation enables the HPC16040 to be networked with other processors. Typically in such environments, the messages consist of addresses and actual data. Addresses are specified by having the ninth bit in the data frame set to 1. Data in the message is specified by having the ninth bit in the data frame reset to 0.

The UART monitors the communication stream looking for addresses. When the data word with the ninth bit set is received, the UART signals the HPC16040 with an interrupt. The processor then examines the content of the receiver buffer to decide whether it has been addressed and whether to accpet subsequent data.



TL/DD/8340–26

FIGURE 22. UART Block Diagram

Universal Peripheral Interface

The Universal Peripheral Interface (UPI) allows the HPC16040 to be used as an intelligent peripheral to another processor. The UPI could thus be used to tightly link two HPC16040's and set up systems with very high data exchange rates. Another area of application could be where a HPC16040 is programmed as an intelligent peripheral to a host system such as the Series 32000 microprocessor. *FIGURE 23* illustrates how a HPC16040 could be used an an intelligent peripherial for a Series 32000-based application.

The interface consists of a Data Bus (port A), a Read Strobe ($\overline{URD}$), a Write Strobe ($\overline{UWR}$), a Read Ready Line ($\overline{RDRDY}$), a Write Ready Line ($\overline{WRRDY}$) and one Address Input (UA0). The data bus can be either eight or sixteen bits wide.

The $\overline{URD}$ and $\overline{UWR}$ inputs may be used to interrupt the HPC16040. The $\overline{RDRDY}$ and $\overline{WRRDY}$ outputs may be used to interrupt the host processor.

The UPI contains an Input Buffer (IBUF), an Output Buffer (OBUF) and a Control Register (UPIC). In the UPI mode, port A on the HPC16040 is the data bus. UPI can only be used if the HPC16040 is in the Single-Chip mode.

Shared Memory Support

Shared memory access provides a rapid technique to exchange data. It is effective when data is moved from a peripheral to memory or when data is moved between blocks of memory. A related area where shared memory access proves effective is in multiprocessing applications where two CPUs share a common memory block. The HPC16040 supports shared memory access with two pins. The pins are the RDY/$\overline{HLD}$ input pin and the $\overline{HLDA}$ output pin. The user can software select either the Hold or Ready function by the state of a control bit. The $\overline{HLDA}$ output is multiplexed onto port B.

The host uses DMA to interface with the HPC16040. The host initiates a data transfer by activating the $\overline{HLD}$ input of the HPC16040. In response, the HPC16040 places its system bus in a TRI-STATE Mode, freeing it for use by the host. The host waits for the acknowledge signal ($\overline{HLDA}$) from the HPC16040 indicating that the sytem bus is free. On receiving the acknowledge, the host can rapidly transfer data into, or out of, the shared memory by using a conventional DMA controller. Upon completion of the message transfer, the host removes the HOLD request and the HPC16040 resumes normal operations.

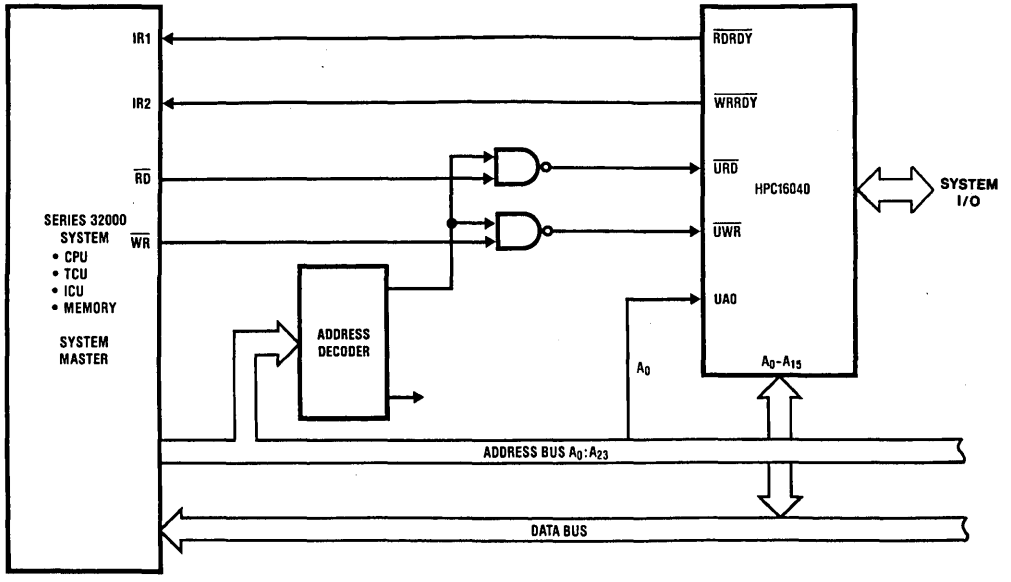
FIGURE 24 illustrates an application of the shared memory interface between the HPC16040 and a Series 32000 system.

Memory

The HPC16040 has been designed to offer flexibility in memory usage. A total address space of 64 kbytes can be addressed with 4096 bytes of ROM and 256 bytes of RAM available on the chip itself. The ROM may contain program instructions, constants or data. The ROM and RAM share the same address space allowing instructions to be executed out of RAM.

Program memory addressing is accomplished by the 16-bit program counter on a byte basis. Memory can be addressed directly by instructions or indirectly through the B, X and SP registers. Memory can be addressed as words or bytes. Words are always addressed on even-byte boundaries. The HPC16040 uses memory-mapped organization to support registers, I/O and on-chip peripheral functions.

The HPC16040 memory address space extends to 64 kbytes and registers and I/O are mapped as shown in Table IV.



TL/DD/8340-27

FIGURE 23. HPC16040 as a Peripheral: (UPI Interface to Series 32000 Application)

Shared Memory Support (Continued)

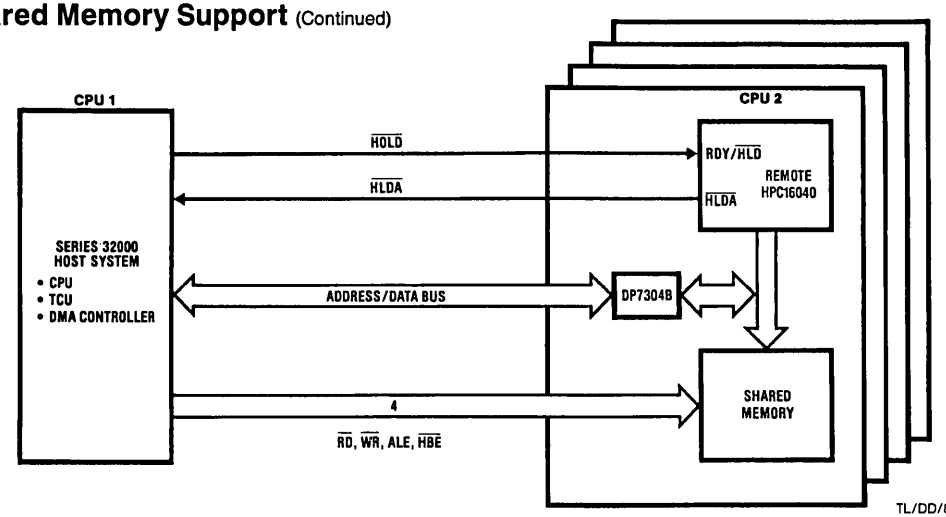


FIGURE 24. Shared Memory Application: HPC16040 Interface to Series 32000 System

TABLE IV. Memory Map

FFFF:FFF0	Interrupt Vectors	USER MEMORY
FFEF:FFD0	JSRP Vectors	
FFCF:FFCE		
: :	On-Chip ROM	
F001:F000		
EFFF:EFFE		
: :	External Expansion	
0201:0200	Memory	
01FF:01FE		
: :	On-Chip RAM	
01C1:01C0		
0195:0194	Watchdog Address	Watchdog Logic
0192	T0CON Register	Timer Block T0:T3
0191:0190	TMMODE Register	
018F:018E	DIVBY Register	
018D:018C	T3 Timer	
018B:018A	R3 Register	
0189:0188	T2 Timer	
0187:0186	R2 Register	
0185:0184	I2CR Register/ R1	
0183:0182	I3CR Register/ T1	
0181:0180	I4CR Register	
0153:0152	Port P Register	Timer Block T4:T7
0151:0150	PWMODE Register	
014F:014E	R7 Register	
014D:014C	T7 Timer	
014B:014A	R6 Register	
0149:0148	T6 Timer	
0147:0146	R5 Register	
0145:0144	T5 Timer	
0143:0142	R4 Register	
0141:0140	T4 Timer	

0128	ENUR Register	UART
0126	TBUF Register	
0124	RBUF Register	
0122	ENUI Register	
0120	ENU Register	
0104	Port D Input Register	
00F5:00F4	BFUN Register	PORTS A & B CONTROL
00F3:00F2	DIR B Register	
00F1:00F0	DIR A Register / IBUF	
00E6	UPIC Register	UPI CONTROL
00E3:00E2	Port B	PORTS A & B
00E1:00E0	Port A / OBUF	
00DE	Microcode ROM Dump	PORT CONTROL & INTERRUPT CONTROL REGISTERS
00DD:00DC	HALT Enable Register	
00D8	Port I Input Register	
00D6	SIO Register	
00D4	IRCD Register	
00D2	IRPD Register	
00D0	ENIR Register	
00CF:00CE	X Register	HPC16040 CORE REGISTERS
00CD:00CC	B Register	
00CB:00CA	K Register	
00C9:00C8	A Register	
00C7:00C6	PC Register	
00C5:00C4	SP Register	
00C3:00C2	(reserved)	
00C0	PSW Register	
00BF:00BE		USER RAM
: :	On-Chip RAM	
0001:0000		

5

HPC16040 CPU

The HPC16040 CPU has a 16-bit ALU and six 16-bit registers

Arithmetic Logic Unit (ALU)

The ALU is 16 bits wide and can do 16-bit add, subtract and shift or logic AND, OR and exclusive OR in one timing cycle. The ALU can also output the carry bit to a 1-bit C register.

Accumulator (A) Register

The 16-bit A register is the source and destination register for most I/O, arithmetic, logic and data memory access operations.

Address (B and X) Registers

The 16-bit B and X registers can be used for indirect addressing. They can automatically count up or down to sequence through data memory.

Boundary (K) Register

The 16-bit K register is used to set limits in repetitive loops of code as register B sequences through data memory.

Stack Pointer (SP) Register

The 16-bit SP register is the pointer that addresses the stack. The SP register is incremented by two for each push or call and decremented by two for each pop or return. The stack can be placed anywhere in user memory and be as deep as the available memory permits.

Program (PC) Register

The 16-bit PC register addresses program memory.

Addressing Modes

ADDRESSING MODES—ACCUMULATOR AS DESTINATION

Register Indirect

This is the "normal" mode of addressing for the HPC16040 (instructions are single-byte). The operand is the memory addressed by the B register (or X register for some instructions).

Direct

The instruction contains an 8-bit or 16-bit address field that directly points to the memory for the operand.

Indirect

The instruction contains an 8-bit address field. The contents of the WORD addressed points to the memory for the operand.

Indexed

The instruction contains an 8-bit address field and an 8- or 16-bit displacement field. The contents of the WORD addressed is added to the displacement to get the address of the operand.

Immediate

The instruction contains an 8-bit or 16-bit immediate field that is used as the operand.

Register Indirect (Auto Increment and Decrement)

The operand is the memory addressed by the X register. This mode automatically increments or decrements the X register (by 1 for bytes and by 2 for words).

Register Indirect (Auto Increment and Decrement) with Conditional Skip

The operand is the memory addressed by the B register. This mode automatically increments or decrements the B register (by 1 for bytes and by 2 for words). The B register is then compared with the K register. A skip condition is generated if B goes past K.

ADDRESSING MODES—DIRECT MEMORY AS DESTINATION

Direct Memory to Direct Memory

The instruction contains two 8- or 16-bit address fields. One field directly points to the source operand and the other field directly points to the destination operand.

Immediate to Direct Memory

The instruction contains an 8- or 16-bit address field and an 8- or 16-bit immediate field. The immediate field is the operand and the direct field is the destination.

Double Register Indirect Using the B and X Registers

Used only with Reset, Set and IF bit instructions; a specific bit within the 64 kbyte address range is addressed using the B and X registers. The address of a byte of memory is formed by adding the contents of the B register to the most significant 13 bits of the X register. The specific bit to be modified or tested within the byte of memory is selected using the least significant 3 bits of register X.

HPC Instruction Set Description

Mnemonic	Description	Action
ARITHMETIC INSTRUCTIONS		
ADD	Add	MA + Meml → MA carry → C
ADC	Add with carry	MA + Meml + C → MA carry → C
DADC	Decimal add with carry	MA + Meml + C → MA (Decimal) carry → C
SUBC	Subtract with carry	MA − Meml + C → MA carry → C
DSUBC	Decimal subtract w/carry	MA − Meml + C → MA (Decimal) carry → C
MULT	Multiply (unsigned)	MA*Meml → MA & X, 0 → K, 0 → C
DIV	Divide (unsigned)	MA/Meml → MA, rem. → X, 0 → K, 0 → C
IFEQ	If equal	Compare MA & Meml, Do next if equal
IFGT	If greater than	Compare MA & Meml, Do next if MA > Meml
AND	Logical and	MA and Meml → MA
OR	Logical or	MA or Meml → MA
XOR	Logical exclusive-or	MA xor Meml → MA
MEMORY MODIFY INSTRUCTIONS		
INC	Increment	Mem + 1 → Mem
DECSZ	Decrement, skip if 0	Mem − 1 → Mem, Skip next if Mem = 0

HPC Instruction Set Description (Continued)

Mnemonic	Description	Action
IT INSTRUCTIONS		
SET	Set bit	1 → Mem.bit (bit is 0 to 7 immediate)
RESET	Reset bit	0 → Mem.bit
IF	If bit	If Mem.bit is true, do next instr.
EMORY TRANSFER INSTRUCTIONS		
LD	Load	Meml → MA
	Load, incr/decr X	Mem(X) → A, X ±1 (or 2) → X
ST	Store to Memory	A → Mem
X	Exchange	A ⟷ Mem
	Exchange, incr/decr X	A ⟷ Mem(X), X ±1 (or 2) → X
PUSH	Push Memory to Stack	W → W(SP), SP+2 → SP
POP	Pop Stack to Memory	SP−2 → SP, W(SP) → W
LDS	Load A, incr/decr B, Skip on condition	Mem(B) → A, B ±1 (or 2) → B, Skip next if B greater/less than K
XS	Exchange, incr/decr B, Skip on condition	Mem(B) ⟷ A,B±1 (or 2) → B, Skip next if B greater/less than K
EGISTER LOAD IMMEDIATE INSTRUCTIONS		
LD A	Load A immediate	imm → A
LD B	Load B immediate	imm → B
LD K	Load K immediate	imm → K
LD X	Load X immediate	imm → X
LD BK	Load B and K immediate	imm → B,imm' → K
CCUMULATOR AND C INSTRUCTIONS		
CLR A	Clear A	0 → A
INC A	Increment A	A + 1 → A
DEC A	Decrement A	A − 1 → A
COMP A	Complement A	1's complement of A → A
SWAP A	Swap nibbles of A	A15:12 ← A11:8 ← A7:4 ⟷ A3:0
RRC A	Rotate A right thru C	C → A15 → ... → A0 → C
RLC A	Rotate A left thru C	C ← A15 ← ... ← A0 ← C
SHR A	Shift A right	0 → A15 → ... → A0 → C
SHL A	Shift A left	C ← A15 ← ... ← A0 ← 0
SET C	Set C	1 → C
RESET C	Reset C	0 → C
IF C	IF C	Do next if C = 1
IFN C	IF not C	Do next if C = 0
RANSFER OF CONTROL INSTRUCTIONS		
JSRP	Jump subroutine from table	PC → W(SP),SP+2 → SP W(table#) → PC
JSR	Jump subroutine relative	PC → W(SP),SP+2 → SP,PC+ # → PC (#is +1025 to −1023)
JSRL	Jump subroutine long	PC → W(SP),SP+2 → SP,PC+ # → PC
JP	Jump relative short	PC+ # → PC(# is +32 to −31)
JMP	Jump relative	PC+ # → PC(#is +257 to −255)
JMPL	Jump relative long	PC+ # → PC
JID	Jump indirect at PC + A	PC+A+1 → PC
JIDW		then Mem(PC)+PC → PC
NOP	No Operation	PC + 1 → PC
RET	Return	SP−2 → SP,W(SP) → PC
RETS	Return then skip next	SP−2 → SP,W(SP) → PC, & skip
RETI	Return from interrupt	SP−2 → SP,W(SP) → PC, interrupt re-enabled

Note: W is 16-bit word of memory
MA is Accumulator A or direct memory (8 or 16-bit)
Mem is 8-bit byte or 16-bit word of memory
Meml is 8- or 16-bit memory or 8 or 16-bit immediate data
imm is 8-bit or 16-bit immediate data

Memory Usage

Number Of Bytes For Each Instruction (number in parenthesis is 16-Bit field)

	Using Accumulator A						To Direct Memory			
	Reg Indir. (B)	Reg Indir. (X)	Direct	Indir	Index	Immed.	Direct *	Direct **	Immed. *	Immed. **
LD	1	1	2(4)	3	4(5)	2(3)	3(5)	5(6)	3(4)	5(6)
X	1	1	2(4)	3	4(5)	—	—	—	—	—
ST	1	1	2(4)	3	4(5)	—	—	—	—	—
ADC	1	2	3(4)	3	4(5)	4(5)	4(5)	5(6)	4(5)	5(6)
SBC	1	2	3(4)	3	4(5)	4(5)	4(5)	5(6)	4(5)	5(6)
DADC	1	2	3(4)	3	4(5)	4(5)	4(5)	5(6)	4(5)	5(6)
DSBC	1	2	3(4)	3	4(5)	4(5)	4(5)	5(6)	4(5)	5(6)
ADD	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
MULT	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
DIV	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
IFEQ	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
IFGT	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
AND	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
OR	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
XOR	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)

*8-bit direct address
**16-bit direct address

Instructions that modify memory directly

	(B)	(X)	Direct	Indir	Index	B&X
SET	1	2	3(4)	3	4(5)	1
RESET	1	2	3(4)	3	4(5)	1
IF	1	2	3(4)	3	4(5)	1
DECSZ	3	2	2(4)	3	4(5)	
INC	3	2	2(4)	3	4(5)	

Immediate Load Instructions

	Immed.
LD B,*	2(3)
LD X,*	2(3)
LD K,*	2(3)
LD BK,*,*	3(5)

Register Indirect Instructions with Auto Increment and Decrement

Register B With Skip		
	(B+)	(B−)
LDS A,*	1	1
XS A,*	1	1

Register X		
	(X+)	(X−)
LD A,*	1	1
X A,*	1	1

Instructions Using A and C

CLR	A	1
INC	A	1
DEC	A	1
COMP	A	1
SWAP	A	1
RRC	A	1
RLC	A	1
SHR	A	1
SHL	A	1
SET	C	1
RESET	C	1
IF	C	1
IFN	C	1

Transfer of Control Instructions

JSRP	1
JSR	2
JSRL	3
JP	1
JMP	2
JMPL	3
JID	1
JIDW	1
NOP	1
RET	1
RETS	1
RETI	1

Stack Reference Instructions

	Direct
PUSH	2
POP	2

Code Efficiency

One of the most important criteria of a single chip microcontroller is code efficiency. The more efficient the code, the more features that can be put on a chip. The memory size on a chip is fixed so if code is not efficient, features may have to be sacrificed or the programmer may have to buy a larger, more expensive version of the chip.

The HPC16040 has been designed to be extremely code-efficient. The HPC16040 looks very good in all the standard coding benchmarks; however, it is not realistic to rely only on benchmarks. Many large jobs have been programmed onto the HPC16040, and the code savings over other popular microcontrollers has been considerable—often the jobs take less than one-half the memory!

Reasons for this saving of code include the following:

SINGLE BYTE INSTRUCTIONS

The majority of instructions on the HPC16040 are single-byte. There are two especially code-saving instructions:

JP is a 1-byte jump. True, it can only jump within a range of plus or minus 32, but many loops and decisions are often within a small range of program memory. Most other micros need 2-byte instructions for any short jumps.

JSRP is a 1-byte call subroutine. The user makes a table of his 16 most frequently called subroutines and these calls will only take one byte. Most other micros require two and even three bytes to call a subroutine. The user does not have to decide which subroutine addresses to put into his table; the assembler can give him this information.

EFFICIENT SUBROUTINE CALLS

The 2-byte JSR instructions can call any subroutine within plus or minus 1k of program memory.

MULTIFUNCTION INSTRUCTIONS FOR DATA MOVEMENT AND PROGRAM LOOPING

The HPC16040 has single-byte instructions that perform multiple tasks. For example, the XS instruction will do the following:

1. Exchange A and memory pointed to by the B register
2. Increment the B register
3. Compare the B register versus the K register
4. Generate a conditional skip if B is greater than K

The value of this multipurpose instruction becomes evident when looping through sequential areas of memory and exiting when the loop is finished.

BIT MANIPULATION INSTRUCTIONS

Any bit of memory, I/O or registers can be set, reset or tested by the single byte bit instructions. The bits can be addressed directly or indirectly. Since all registers and I/O are mapped into the memory, it is very easy to manipulate specific bits to do efficient control.

The one exception to the above is with the IRPD register. A LOAD IMMEDIATE instruction is to be the only instruction used to clear a bit or bits in this register (see Interrupt Pending Register section).

DECIMAL ADD AND SUBTRACT

This instruction is needed to interface with the decimal user world.

It can handle both 16-bit words and 8-bit bytes.

The 16-bit capability saves code since many variables can be stored as one piece of data and the programmer does not have to break his data into two bytes. Many applications store most data in 4-digit variables. The HPC16040 supplies 8-bit byte capability for 2-digit variables and literal variables.

MULTIPLY AND DIVIDE INSTRUCTIONS

The HPC16040 has 16-bit multiply, 16-bit by 16-bit divide, and 32-bit by 16-bit divide instructions. This saves both code and time. Multiply and divide can use immediate data or data from memory. The ability to multiply and divide by immediate data saves code since this function is often needed for scaling, base conversion, computing indexes of arrays, etc.

Development Support

The MOLE (Microcontroller On-Line Emulator) is a low cost development system and emulator for all microcontroller products. These include COPs, TMP, 8050U and the HPC Family of Products. The MOLE consists of a BRAIN Board, Personality Board and optional host software.

The purpose of a MOLE is to provide the user with a tool to write and assemble code, emulate code for the target microcontroller and assist in both the software & hardware debugging of the system.

It is a self-contained computer with its own firmware which provides for all system operation, emulation control, communication, PROM programming and diagnostic operations.

It contains three serial ports because multiple ports are usually needed to optionally connect to a terminal, a host system, a printer or modem, or to connect to other MOLEs in a multi-MOLE environment.

MOLE can be used in either a stand alone mode or in conjunction with selected host systems, i.e., those using CP/M or PC-DOS. Communicating via RS-232 port.

Dial-A-Helper is a service provided by the MOLE applications group. If a user is having difficulty in getting a MOLE to operate in a particular mode or it is acting peculiar, he can contact us via his system and a modem. He can leave messages on our electronic bulletin board which we will respond to, or he can arrange for us to actually take control of his system via modem for debugging purposes.

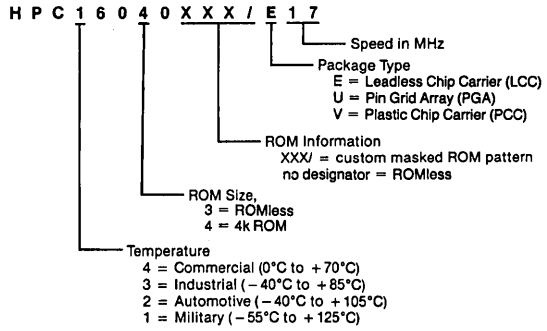
The applications group can then cause his system to execute various commands and try to resolve the customer's problem by actually getting the customer's system to respond. 99% of the time the problem is resolved. This allows us to respond in minutes instead of days when applications help is needed.

The system can also be used to download available applications software.

Part Selection

The HPC family includes devices with many different options and configurations to meet various application needs. The number HPC16040 has been generically used throughout this datasheet to represent the whole family of parts. The following chart explains how to order various options available when ordering HPC family members.

Note: All options may not currently be available.



TL/DD/8340-12

FIGURE 8. HPC Family Part Numbering Scheme

Examples

HPC46030E17 — ROMless, Commercial temp. (0°C to 70°C), LCC

HPC16040XXX/U17— 4k masked ROM, Military temp. (−55°C to +125°C), PGA

HPC26040XXX/V17— 4k masked ROM, Automotive temp. (−40°C to +105°C), PCC

National Semiconductor Corporation

PRELIMINARY

HPC16083/HPC26083/HPC36083/HPC46083/HPC16073/HPC36073/HPC46073 High-Performance Microcontroller with Input Capture Registers

General Description

The HPC16083 is a member of the HPC™ family of High Performance Microcontrollers. Each member of the family has the same identical core CPU with a unique memory and I/O configuration to suit specific applications. Each part is fabricated in National's advanced microCMOS technology. This process combined with an advanced architecture provides fast, flexible I/O control, efficient data manipulation, and high speed computation.

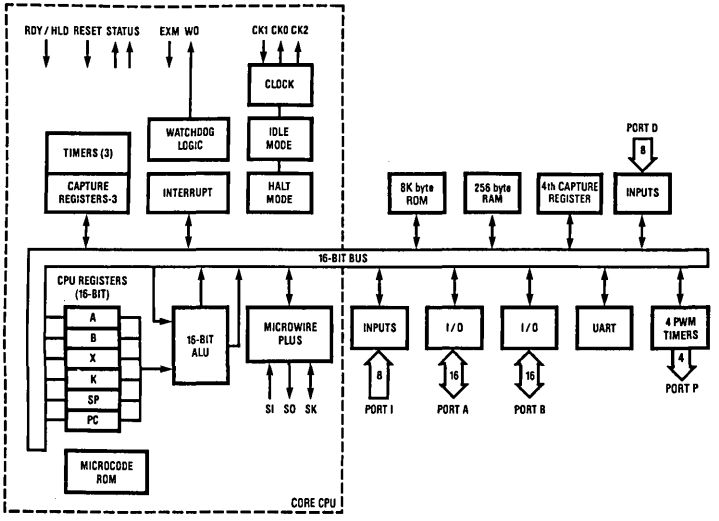
The HPC16083 is a complete microcomputer on a single chip. All system timing, internal logic, ROM, RAM, and I/O are provided on the chip to produce a cost effective solution for high performance applications. On-chip functions such as UART, up to Eight 16-bit timers with up to 4 input capture registers, vectored interrupts, WATCHDOG logic and MICROWIRE/PLUS™ provide a high level of system integration. The ability to address up to 64k bytes of external memory enables the HPC16083 to be used in powerful applications typically performed by microprocessors and expensive peripheral chips.

The microCMOS process results in very low current drain and enables the user to select the optimum speed/power product for his system. The IDLE and HALT modes provide further current savings. The HPC16083 is available in 68-pin PCC, LCC and PGA packages.

Features

- HPC family—core features:
  - — 16-bit architecture, both byte and word
  - — 16-bit data bus, ALU, and registers
  - — 64k bytes of external memory addressing
  - — FAST!—240 ns for register instructions when using 17.0 MHz clock
  - — High code efficiency—most instructions are single byte
  - — 16 x 16 multiply and 32 x 16 divide
  - — Eight vectored interrupt sources
  - — Four 16-bit timer/counters with 4 synchronous outputs and WATCHDOG logic
  - — MICROWIRE/PLUS serial I/O interface
  - — CMOS—very low power with two power save modes: IDLE and HALT (2 mA, 250 μA—typ.)
- UART—full duplex, programmable baud rate
- Four additional 16-bit timer/counters with pulse width modulated outputs
- Four input capture registers
- 52 general purpose I/O lines (memory mapped)
- 8k bytes of ROM, 256 bytes of RAM on chip
- ROMless versions available
- Wide voltage supply range: 3 to 5.5V
- Industrial (−40°C to +85°C) and military (−55°C to +125°C) temperature ranges

Block Diagram



TL/DD/8801-1

National Semiconductor Corporation

PRELIMINARY

HPC16400/HPC36400/HPC46400 High-Performance Microcontrollers with HDLC Controller

General Description

The HPC16400 is a member of the HPC™ family of High Performance microControllers. Each member of the family has the same identical core CPU with a unique memory and I/O configuration to suit specific applications. Each part is fabricated in National's advanced microCMOS technology. This process combined with an advanced architecture provides fast, flexible I/O control, efficient data manipulation, and high speed computation.

The HPC16400 has 4 functional blocks to support a wide range of communication application-2 HDLC channels, 4 channel DMA controller to facilitate data flow for the HDLC channels, programmable serial interface and UART.

The serial interface decoder allows the 2 HDLC channels to be used with devices using interchip serial link for point-to-point & multipoint data exchanges. The decoder generates enable signals for the HDLC channels allowing multiplexed D and B channel data to be accessed.

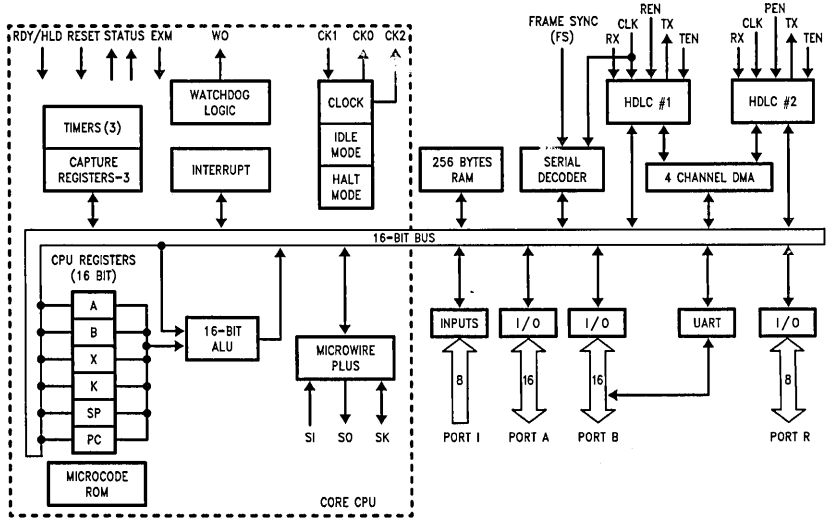
The HDLC channels manage the link by providing sequencing using the HDLC framing along with error control and retransmission based upon a cyclic redundancy check (CRC). Multiple frame addressing, and both bit and byte modes of operation are supported.

The HPC16400 is available in 68-pin PCC, LCC and PGA packages.

Features

- HPC family—core features:
  - 16-bit data bus, ALU, and registers
  - 64 kbytes of external memory addressing
  - FAST!—17.0 MHz system clock
  - High code efficiency
  - 16 x 16 multiply and 32 x 16 divide
  - Eight vectored interrupt sources
  - Four 16-bit timer/counters with WATCHDOG logic
  - MICROWIRE/PLUS serial I/O interface
  - CMOS—low power with two power save modes
- Two full duplex HDLC channels
  - Optimized or X.25 and LAPD applications
  - Programmable frame address recognition
  - Up to 4.1 Mbps aggregate serial data rate
  - Built in diagnostics
- Programmable interchip serial data decoder
- Four channel DMA controller
- UART—full duplex, programmable baud rate
- 544 kbytes of extended addressing
- Easy interface to National's 'U' and 'S' transceiver—TP3400, TP3410 and TP3420
- Wide voltage supply range: 3 to 5.5V
- Industrial (−40°C to +85°C) and military (−55°C to +125°C) temperature ranges

Block Diagram



TL/DD/8802-1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Total Allowable Source or Sink Current	100 mA
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 10 sec)	300°C
V_{CC} with Respect to GND	−0.5V to 7.0V
All Other Pins	(V_{CC} + 0.5)V to (GND − 0.5)V

Note: *Absolute maximum ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications are not ensured when operating the device at absolute maximum ratings.*

DC Electrical Characteristics

V_{CC} = 5.0V ±10% unless otherwise specified, T_A = 0°C to +70°C for HPC46040, −40°C to +85°C for HPC36040, −40°C to +105°C for HPC26040, −55°C to +125°C for HPC16400

Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
I_{CC_1}	Supply Current	V_{CC} = 5.0V, f_{in} = 17.0 MHz*		20		mA
		V_{CC} = 5.0V, f_{in} = 2.0 MHz		2.4		mA
I_{CC_2}	IDLE Mode Current	V_{CC} = 5.0V, f_{in} = 17.0 MHz, T_A = 25°C*		2		mA
		V_{CC} = 5.0V, f_{in} = 2.0 MHz, T_A = 25°C		0.2		mA
I_{CC_3}	HALT Mode Current	V_{CC} = 5.0V, f_{in} = 0 kHz, T_A = 25°C*		25		μA
		V_{CC} = 2.5V, f_{in} = 0 kHz, T_A = 25°C		10		μA
INPUT VOLTAGE LEVELS RESET, NMI, CKI AND WO (SCHMITT TRIGGERED)						
V_{IH_1}	Logic High			0.9 V_{CC}		V
V_{IL_1}	Logic Low			0.1 V_{CC}		V
ALL OTHER INPUTS						
V_{IH_2}	Logic High			0.7 V_{CC}		V
V_{IL_2}	Logic Low			0.2 V_{CC}		V
I_{LI}	Input Leakage Current			±1		μA
C_I	Input Capacitance			10		pF
C_{IO}	I/O Capacitance			20		pF
OUTPUT VOLTAGE LEVELS CMOS OPERATION						
V_{OH_1}	Logic High	I_{OH} = −10 μA		V_{CC} − 0.2		V
V_{OL_1}	Logic Low	I_{OH} = 10 μA		0.2		V
V_{OH_2}	Port A/B Drive (A_0–A_{15}, B_{10}, B_{11}, B_{12}, B_{15})	I_{OH} = −7 mA, V_{CC} = 5.0V		2.4		V
V_{OL_2}		I_{OL} = 3 mA		0.4		V
V_{OH_3}	Other Port Pin Drive (B_0–B_9, B_{13}, B_{14}, P_0–P_3)	I_{OH} = −1.6 mA, V_{CC} = 5.0V		2.4		V
V_{OL_3}		I_{OL} = 0.5 mA		0.4		V
V_{OL_4}	WO (Watchdog Out) Drive	I_{OL} = 0.5 mA, V_{CC} = 5.0V		0.4		V
V_{OH_6}	CK2 Drive	I_{OH} = −12 mA, V_{CC} = 5.0V		2.4		V
V_{OL_6}		I_{OL} = 3.5 mA		0.4		V
V_{OH_7}	ST1 and ST2 Drive	I_{OH} = −6 mA, V_{CC} = 5.0V		2.4		V
V_{OL_7}		I_{OL} = 1.6 mA		0.4		V
V_{RAM}	RAM Keep-Alive Voltage			2.5		V
I_{OZ}	TRI-STATE Leakage Current			±5		μA

Note: I_{CC_1}, I_{CC_2}, I_{CC_3} measured with no external drive (I_{OH} and I_{OL} = 0, I_{IH} and I_{IL} = 0).

AC Electrical Characteristics

V_{CC} = 5.0V ±10%, f_C = 16.78 MHz, T_A = 0°C to +70°C for HPC46040, −40°C to +85°C for HPC36040, −40°C to +105°C for HPC26040, −55°C to +125°C for HPC16400

Symbol	Parameter	Min	Typ	Max	Units
f_C = CKI freq.	Operating Frequency		16.78	17.0	MHz
$t_{C1} = 1/f_C$	Clock Period		60		ns
$t_C = 2/f_C$	Timing Cycle		120		ns
$t_{LL} = 1/2\ t_C$	ALE Pulse Width		60		ns
$t_{ST} = 1/4\ t_C$	Address Valid to ALE Trailing Edge		30		ns
$t_{WAIT} = t_C = WS$	Wait State Period		120		ns
$f_{XIN} = 1/19\ t_{C1}$	External Timer Input Frequency		877		kHz
$t_{XIN} = 3\ t_{C1}$	Pulse Width for Timer Inputs		180		ns
$f_{XOUT} = 1/16\ t_{C1}$	Timer Output Frequency		1.04		MHz
$f_{MW} = 1/19\ t_{C1}$	External MICROWIRE/PLUS Clock Input Frequency		877		kHz
$f_U = 1/19\ t_{C1}$	External UART Clock Input Frequency		877		kHz

Read Cycle Timing

f_C = 16.78 MHz with One Wait State

Symbol	Parameter	Min	Typ	Max	Units
$t_{ARR} = 1/4\ t_C + 5$	ALE Trailing Edge to $\overline{RD}$ Falling Edge		35		ns
$t_{RW} = 1/2\ t_C + WS$	$\overline{RD}$ Pulse Width		180		ns
t_{DR}	Data Valid before Trailing Edge of $\overline{RD}$		15		ns
$t_{ACC} = t_C + WS - 55$	Address Valid to Input Data Valid		185		ns
$t_{RD} = 1/2\ t_C + WS - 65$	$\overline{RD}$ Falling Edge to Data in Valid		115		ns
$t_{RDA} = WS = t_C$	$\overline{RD}$ Falling Edge to Address Valid		120		ns
$t_{VPR} = 1/4\ t_C + 5$	Address Valid from ALE Trailing Edge Prior to $\overline{RD}$		35		ns
$t_{HZ} = 3/4\ t_C - 10$	End of $\overline{RD}$ to Input Data Float		80		ns

Write Cycle Timing

f_C = 16.78 MHz with One Wait State

Symbol	Parameter	Min	Typ	Max	Units
$t_{ARW} = 1/2\ t_C$	ALE Trailing Edge to $\overline{WR}$ Falling Edge		60		ns
$t_{WW} = 3/4\ t_C + WS + 5$	$\overline{WR}$ Pulse Width		215		ns
t_{HW}	Data Hold after Trailing Edge of $\overline{WR}$		20		ns
$t_V = 1/2\ t_C + WS + 5$	Data Valid before Trailing Edge of $\overline{WR}$		185		ns
$t_{VPW} = 1/4\ t_C + 25$	Address Valid from Trailing Edge Prior to $\overline{WR}$		55		ns

Ready/Hold Timing f_C = 16.78 MHz with One Wait State

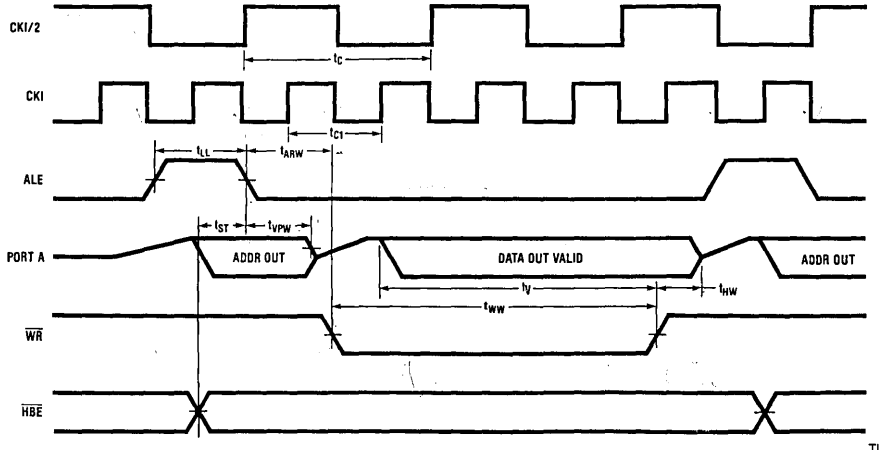
Symbol	Parameter	Min	Typ	Max	Units
$t_{DAR} = \frac{1}{4}t_C + WS - 50$	Falling Edge of ALE to Falling Edge of $\overline{RDY}$		100		ns
$t_{RWP} = t_C$	$\overline{RDY}$ Pulse Width		120		ns
$t_{SALE} = \frac{1}{4}t_C + 40$	Falling Edge of $\overline{HLD}$ to Rising Edge of ALE		70		ns
$t_{HWP} = t_C + 10$	$\overline{HLD}$ Pulse Width		130		ns
t_{HAD}	Rising Edge on $\overline{HLD}$ to Rising Edge on $\overline{HLDA}$		120		ns
$t_{HAE} = t_C + 100$	Falling Edge on $\overline{HLD}$ to Falling Edge on $\overline{HLDA}$		220	*	ns
$t_{BF} = t_C + 30$	Bus Float before Falling Edge on $\overline{HLDA}$		150		ns
$t_{BE} = 2t_C + 50$	Bus Enable from Rising Edge of $\overline{HLD}$		290		ns

*Note: t_{HAE} may be as long as ($3t_C$ + 4ws + $72t_C$ + 90) depending on which instruction is being executed, the addressing mode and number of wait states.

Status Timing f_C = 16.78 MHz

Symbol	Parameter	Min	Typ	Max	Units
$t_{SRS2} = 40 - (\frac{1}{4}t_C + 25)$	Setup Time for ST2 on Rising Edge of ALE		−15		ns
$t_{HRS2} = \frac{3}{4}t_C - 15$	Hold Time for ST2 on Rising Edge of ALE		75		ns
$t_{SFS2} = 40 - (\frac{1}{4}t_C + 25)$	Setup Time for ST2 on Falling Edge of ALE		−15		ns
$t_{HFS2} = \frac{3}{4}t_C - 15$	Hold Time for ST2 on Falling Edge of ALE		75		ns
t_{SFS1}	Setup Time for ST1 on Falling Edge of $\overline{RD}$		20		ns
$t_{HRS1} = \frac{1}{2}t_C - 15$	Hold Time for ST1 on Rising Edge of $\overline{RD}$		45		ns

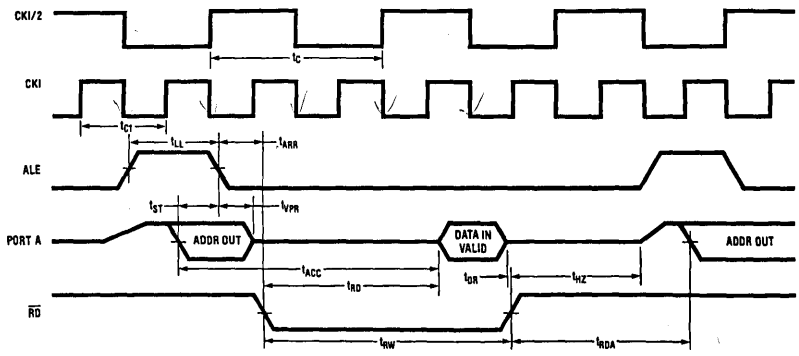
Timing Waveforms



TL/DD/8802-2

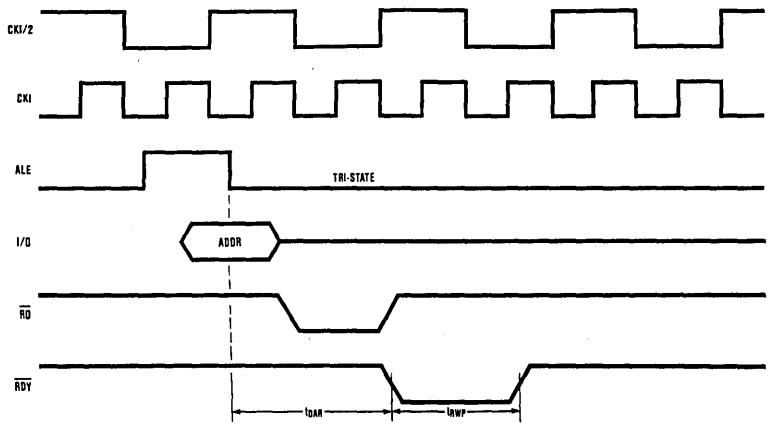
FIGURE 1. Write Cycle

Timing Waveforms (Continued)



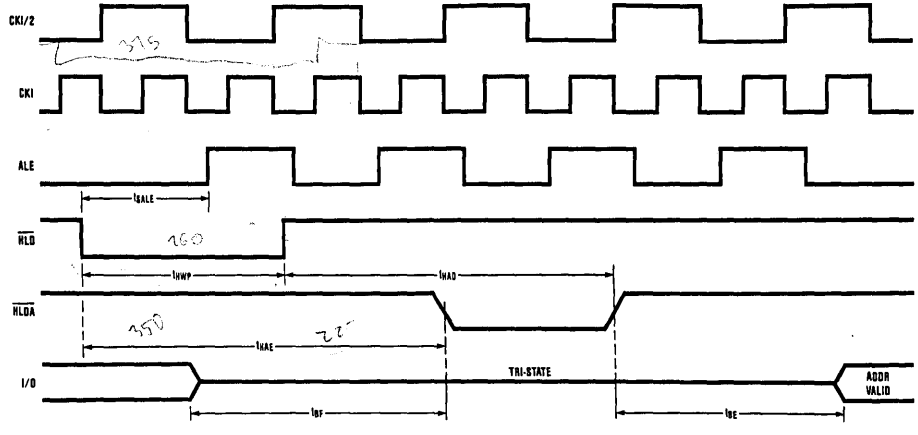
TL/DD/8802-3

FIGURE 2. Read Cycle



TL/DD/8802-4

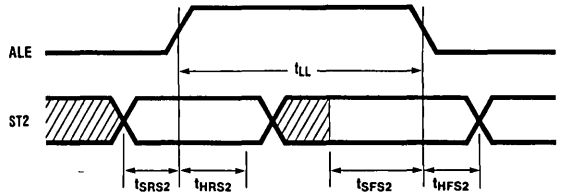
FIGURE 3. Ready Mode Timing



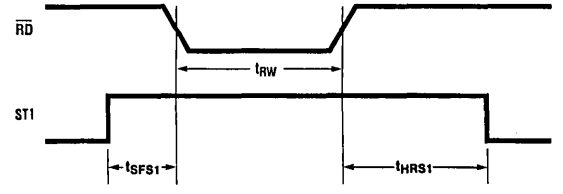
TL/DD/8802-5

FIGURE 4. Hold Mode Timing

Timing Waveforms (Continued)



TL/DD/8802-6



TL/DD/8802-7

FIGURE 5. Status Timing

Pin Descriptions†

I/O PORTS

Port A is a 16-bit multiplexed address/data bus used for accessing external program and data memory. Four associated bus control signals are available on port B. The Address Latch Enable (ALE) signal is used to provide timing to demultiplex the bus. Reading from and writing to external memory are signalled by RD* and WR* respectively. External memory can be addressed as either bytes or words with the decoding controlled by two lines, Bus High Byte enable (HBE*) and Address/Data Line 0 (A0).

Port B is a 16-bit port, with 12 bits of bidirectional I/O similar in structure to port A. Pins B10, B11, B12 and B15 are the control bus signals for the address/data bus. Port B may also be configured via a function register BFUN to individually allow each bidirectional I/O pin to have an alternate function.

B0:	TDX	UART Data Output
B1:		
B2:	CKX	UART Clock (Input or Output)
B3:	T2IO	Timer2 I/O Pin
B4:	T3IO	Timer3 I/O Pin
B5:	SO	MICROWIRE/PLUS Output
B6:	SK	MICROWIRE/PLUS Clock (Input or Output)
B7:	HLDA*	Hold Acknowledge Output
B8:	TS0	Timer Synchronous Output
B9:	TS1	Timer Synchronous Output
B10:	ALE	Address Latch Enable Output for Address/Data Bus
B11:	WR*	Address/Data Bus Write Output
B12:	HBE*	High Byte Enable Output for Address/Data Bus
B13:	TS2	Timer Synchronous Output
B14:	TS3	Timer Synchronous Output
B15:	RD*	Address/Data Bus Read Output

When operating in the extended memory addressing mode, four bits of port B can are used as follows—

B8:	BS0	Memory bank switch output 0 (LSB)
B9:	BS1	Memory bank switch output 1
B13:	BS2	Memory bank switch output 2
B14:	BS3	Memory bank switch output 3 (MSB)

Port I is an 8-bit input port that can be read as general purpose inputs and can also be used for the following functions:

I0:		
I1:	NMI	Nonmaskable Interrupt Input
I2:	INT2	Maskable Interrupt/Input Capture
I3:	INT3	Maskable Interrupt/Input Capture
I5:	SI	MICROWIRE/PLUS Data Input
I6:	RDX	UART Data Input
I7:	FS	IDL Frame Sync Input Signal

Port D is an 8-bit input port that is used for the following functions:

D0:	RX1	HDLC # 1 Receive Data Input
D1:	CLK1	HDLC # 1 Clock Input
D2:	REN1	HDLC # 1 Receiver Enable Input
D3:	TEN1	HDLC # 1 Transmit Enable Input
D4:	RX2	HDLC # 2 Receive Data Input
D5:	CLK2	HDLC # 2 Clock Input
D6:	REN2	HDLC # 2 Receiver Enable Input
D7:	TEN2	HDLC # 2 Transmit Enable Input

Port R is an 8-bit bidirectional I/O port available for general purpose I/O operation. Additional functions are present as indicated.

R0:	TX1	HDLC #1 Transmit Output
R1:	TX2	HDLC #2 Transmit Output
R2:		
R3:		
R4:		
R5:		
R6:		
R7:		

†The formation of the various functions into specified ports has changed. Please contact factory for updated port configurations.

5

Pin Descriptions (Continued)

POWER SUPPLIES

V_{CC}	Positive Power Supply (two pins)
GND	Ground for On-Chip Logic
DGND	Ground for Output Buffers

CLOCK PINS

CKI	The System Clock Input
CKO	The System Clock Output (Inversion of CKI)

Pins CKI and CKO are usually connected across an external crystal.

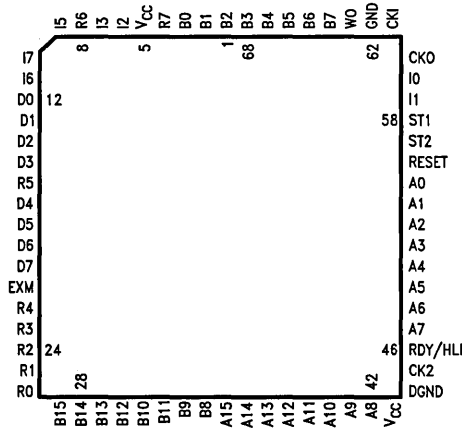
CK2	Clock Output (CKI divided by 2)

OTHER PINS

WO	This is an active low open drain output which signals an illegal situation has been detected by the Watch Dog logic.
ST1	Bus Cycle Status Output indicates first opcode fetch.
ST2	Bus Cycle Status Output indicates machine states (skip and interrupt).
RESET	Active low input that forces the chip to restart and sets the ports in a TRI-STATE mode.
RDY/HLD	Has two uses, selected by a software bit. This pin is either a READY input to extend the bus cycle for slower memories or a HOLD-REQUEST input to put the bus in a high impedance state for external DMA purposes.
EXM	External memory enable which must be tied high for normal operation.

Connection Diagram *

Plastic and Leadless Chip Carriers



Order Number HPC16400E or V
See NS Package Number
E68B or V68A

TL/DD/8802-17

Top View

*The Pin Configuration has changed. Please contact factory for updated pin placement information.

Wait States

The HPC16400 provides four software selectable Wait States that allow access to slower memories. The Wait States are selected by the state of two bits in the PSW register. Additionally, the RDY input may be used to extend the instruction cycle, allowing the user to interface with slow memories and peripherals.

Power Save Modes

Two power saving modes are available on the HPC16400: HALT and IDLE. In the HALT mode, all processor activities are stopped. In the IDLE mode, the on-board oscillator and timer T0 are active but all other processor activities are stopped. In either mode, all on-board RAM, registers and I/O are unaffected.

HALT MODE

The HPC16400 is placed in the HALT mode under software control by setting bits in the PSW. All processor activities, including the clock and timers, are stopped. In the HALT mode, power requirements for the HPC16400 are minimal and the applied voltage (V_{CC}) may be decreased without altering the state of the machine. There are two ways of exiting the HALT mode: via the RESET or the NMI. The RESET input reinitializes the processor. Use of the NMI input will generate a vectored interrupt and resume operation from that point with no initialization. The HALT mode can be enabled or disabled by means of a control register HALT enable. To prevent accidental use of the HALT mode the HALT enable register can be modified only once.

IDLE MODE

The HPC16400 is placed in the IDLE mode through the PSW. In this mode, all processor activity, except the on-board oscillator and Timer T0, is stopped. The HPC16400 resumes normal operation upon timer T0 overflow. As with the HALT mode, the processor is returned to full operation by the RESET or NMI inputs, but without waiting for oscillator stabilization.

HPC16400 Interrupts

Complex interrupt handling is easily accomplished by the HPC16400's vectored interrupt scheme. There are eight possible interrupt sources as shown in Table I.

TABLE I. Interrupts

Vector/ Address	Interrupt Source	Arbitration Ranking
FFFF|FFFE	Reset	0
FFFD|FFFC	Nonmaskable Ext (NMI)	1
FFFB|FFFA	External on I2	2
FFF9|FFF8	External on I3	3
FFF7|FFF6	HDLC/DMA Error	4
FFF5|FFF4	Overflow on Timers	5
FFF3|FFF2	Internal on UART	6
FFF1|FFF0	End of Message (EOM)	7

The 16400 contains arbitration logic to determine which interrupt will be serviced first if two or more interrupts occur simultaneously. Interrupts are serviced after the current instruction is completed except for the RESET which is serviced immediately.

The NMI interrupt will immediately stop DMA activity-byte transfers in progress will finish thereby allowing an orderly transition to the interrupt service vector (see DMA description). The HDLC channels continue to operate, and the user must service data errors that might have occurred during the NMI service routine.

Interrupt Processing

Interrupts are serviced after the current instruction is completed except for the RESET, which is serviced immediately.

RESET is a level-sensitive interrupt. All other interrupts are edge-sensitive. NMI is positive-edge sensitive. The external interrupts on I2, I3 can be software selected to be rising or falling edge.

Interrupt Control Registers

The HPC16400 allows the various interrupt sources and conditions to be programmed. This is done through the various control registers. A brief description of the different control registers is given below.

INTERRUPT ENABLE REGISTER (ENIR)

RESET and the External Interrupt on I1 are non-maskable interrupts. The other interrupts can be individually enabled or disabled. Additionally, a Global Interrupt Enable Bit in the ENIR Register allows the Maskable interrupts to be collectively enabled or disabled. Thus, in order for a particular interrupt to be serviced, both the individual enable bit and the Global Interrupt bit (GIE) have to be set.

INTERRUPT PENDING REGISTER (IRPD)

The IRPD register contains a bit allocated for each interrupt vector. The occurrence of specified interrupt trigger conditions causes the appropriate bit to be set. There is no indication of the order in which the interrupts have been received. The bits are set independently of the fact that the interrupts may be disabled. IRPD is a Read/Write register. The bits corresponding to the maskable, external interrupts are normally cleared by the HPC16400 after servicing the interrupts.

For the interrupts from the on-board peripherals, the user has the responsibility of resetting the interrupt pending flags through software.

INTERRUPT CONDITION REGISTER (IRCD)

Three bits of the register select the input polarity of the external interrupt on I2, I3, and I4.

Servicing the Interrupts

The Interrupt, once acknowledged, pushes the program counter (PC) onto the stack thus incrementing the stack pointer (SP) twice. The Global Interrupt Enable (GIE) bit is reset, thus disabling further interrupts. The program counter is loaded with the contents of the memory at the vector address and the processor resumes operation at this point. At the end of the interrupt service routine, the user does a RETI instruction to pop the stack, set the GIE bit and return to the main program. The GIE bit can be set in the interrupt service routine to nest interrupts if desired. *Figure 6* shows the Interrupt Enable Logic.

Reset

The RESET input initializes the processor and sets ports A, B, and P in the TRI-STATE condition. RESET is an active-low Schmitt trigger input. The processor vectors to FFFF:FFFE and resumes operation at the address contained at that memory location.

Timer Overview

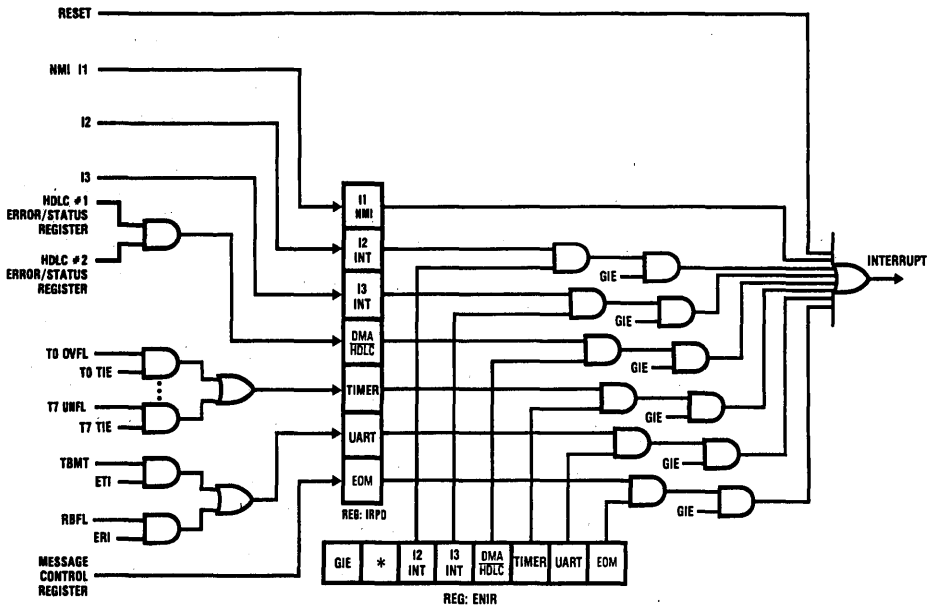
The HPC16400 contains a powerful set of flexible timers enabling the HPC16400 to perform extensive timer functions; not usually associated with microcontrollers.

The HPC16400 contains eight 16-bit timers. Each timer has an associated 16-bit register. Timer T0 is a free-running timer, counting up at a fixed CKI/16 (Clock Input/16) rate. It is used for Watch Dog logic, high speed event capture, and to exit from the IDLE mode. Consequently, it cannot be stopped or written to under software control. Timer T0 permits precise measurements by means of the capture registers I2CR, I3CR, and I4CR. A control bit in the register TMMODE configures timer T1 and its associated register R1 as capture registers I3CR and I2CR. The capture registers I2CR, I3CR, and I4CR respectively, record the value of timer T0 when specific events occur on the interrupt pins I2, I3, and I4. The control register IRCD programs the capture registers to trigger on either a rising edge or a falling edge of its respective input. The specified edge can also be programmed to generate an interrupt (see *Figure 7*).

The timers T2 and T3 have selectable clock rates. The clock input to these two timers may be selected from the following two sources: an external pin, or derived internally by dividing the clock input. Timer T2 has additional capability of being clocked by the timer T3 underflow. This allows the user to cascade timers T3 and T2 into a 32-bit timer/counter. The control register DIVBY programs the clock input to timers T2 and T3 (see *Figure 8*).

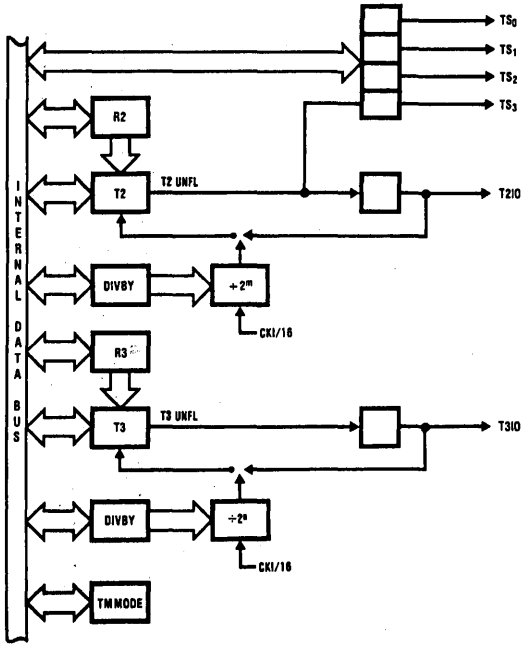
The timers T1 through T7 in conjunction with their registers form Timer-Register pairs. The registers hold the pulse duration values. All the Timer-Register pairs can be read from or written to. Each timer can be started or stopped under software control. Once enabled, the timers count down, and upon underflow, the contents of its associated register are automatically loaded into the timer.

Timer Overview (Continued)



TL/DD/8802-8

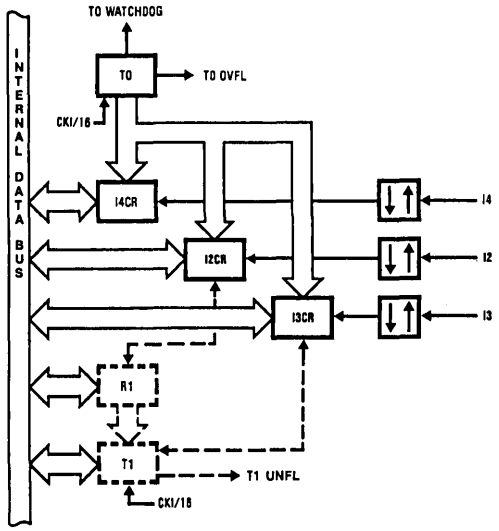
FIGURE 6. Interrupt Enable Logic



TL/DD/8802-10

FIGURE 8. Timers T2–T3 Block

Timer Overview (Continued)



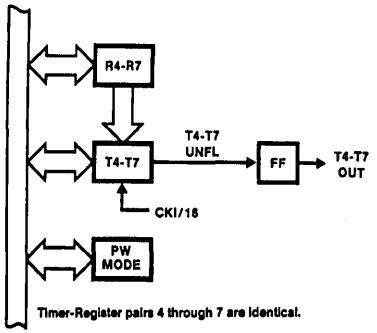
TL/DD/8802-9

FIGURE 7. Timers T0–T1 Block

SYNCHRONOUS OUTPUTS

The flexible timer structure of the HPC16400 simplifies pulse generation and measurement. There are four synchronous timer outputs (TS0 through TS3) that work in conjunction with the timer T2. The synchronous timer outputs can be used either as regular outputs or individually programmed to toggle on timer T2 underflows (see *Figure 8*).

Timer/register pairs 4–7 form four identical units which can generate synchronous outputs on port P (see *Figure 9*).



TL/DD/8802-11

FIGURE 9. Timers T4–T7 Block

Timer Registers

There are four control registers that program the timers. The divide by (DIVBY) register programs the clock input to timers T2 and T3. The timer mode register (TMMODE) contains control bits to start and stop timers T1 through T3. It also contains bits to latch and enable interrupts from timers T0 through T3. The control register PWMODE similarly programs the pulse width timers T4 through T7 by allowing them to be started, stopped, and to latch and enable interrupts on underflows. The PORTP register contains bits to preset the outputs and enable the synchronous timer output functions.

Timer Applications

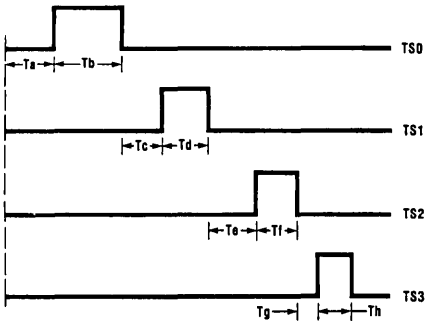
The use of Pulse Width Timers for the generation of various waveforms is easily accomplished by the HPC16400.

Frequencies can be generated by using the timer/register pairs. A square wave is generated when the register value is a constant. The duty cycle can be controlled simply by changing the register value.

TL/DD/8802-12

FIGURE 10. Square Wave Frequency Generation

Synchronous outputs based on Timer T2 can be generated on the 4 outputs TS0–TS3. Each output can be individually programmed to toggle on T2 underflow. Register R2 contains the time delay between events. *Figure 11* is an example of synchronous pulse train generation.



TL/DD/8802-13

FIGURE 11. Synchronous Pulse Generation

Watch Dog Logic

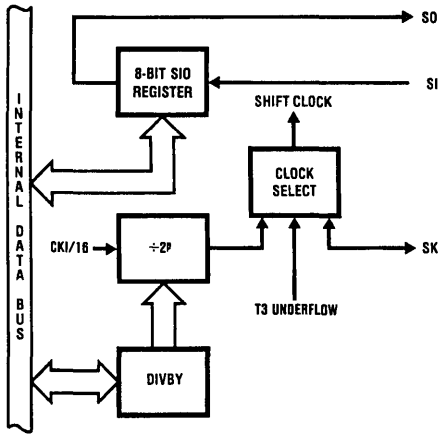
The Watch Dog Logic monitors the operations taking place and signals upon the occurrence of any illegal activity. The illegal conditions that trigger the Watch Dog logic are potentially infinite loops and illegal addresses. Should the Watch Dog register not be written to before Timer T0 overflows twice, or more often than once every 4096 counts, an infinite loop condition is assumed to have occurred. The illegal condition forces the Watch Out (WO) pin low. The WO pin is an open drain output and can be connected to the RESET or NMI inputs or to the users external logic.

MICROWIRE/PLUS

MICROWIRE/PLUS is used for synchronous serial data communications (see *Figure 12*). MICROWIRE/PLUS has an 8-bit parallel-loaded, serial shift register using SI as the input and SO as the output. SK is the clock for the serial shift register (SIO). The SK clock signal can be provided by an internal or external source. The internal clock rate is programmable by the DIVBY register. A DONE flag indicates when the data shift is completed.

The MICROWIRE/PLUS capability enables it to interface with any of National Semiconductor's MICROWIRE peripherals (i.e., A/D converters, display drivers, EEPROMs).

5



TL/DD/8802–14

FIGURE 12. MICROWIRE/PLUS

MICROWIRE/PLUS Operation

The HPC16400 can enter the MICROWIRE/PLUS mode as the master or a slave. A control bit in the IRCD register determines whether the HPC16400 is the master or slave. The shift clock is generated when the HPC16400 is configured as a master. An externally generated shift clock on the SK pin is used when the HPC16400 is configured as a slave. When the HPC16400 is a master, the DIVBY register programs the frequency of the SK clock. The DIVBY register allows the SK clock frequency to be programmed in 15 selectable steps from 64 Hz to 1 MHz with CKI at 17.0 MHz.

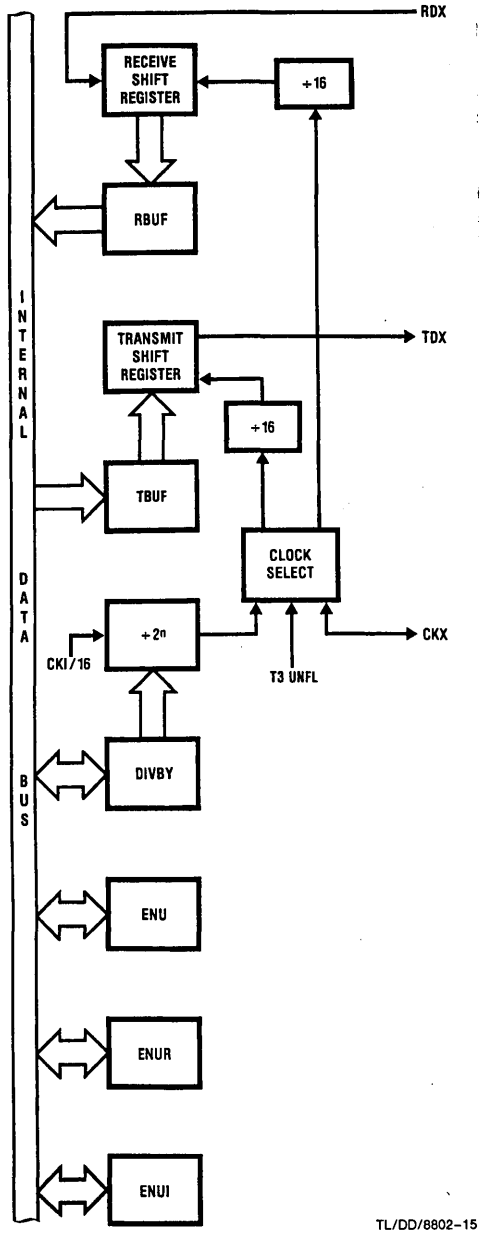
The contents of the SIO register may be accessed through any of the memory access instructions. Data waiting to be transmitted in the SIO register is clocked out on the falling edge of the SK clock. Serial data on the SI pin is clocked in on the rising edge of the SK clock.

HPC16400 UART

The HPC16400 contains a software programmable UART. The UART (see *Figure 13*) consists of a transmit shift register, a receiver shift register and five addressable registers, as follows: a transmit buffer register (TBUF), a receiver buffer register (RBUF), a UART control and status register (ENU), a UART receive control and status register (ENUR) and a UART interrupt and clock source register (ENUI). The ENU register contains flags for transmit and receive functions; this register also determines the length of the data frame (8 or 9 bits) and the value of the ninth bit in transmission. The ENUR register flags framing and data overrun errors while the UART is receiving. Other functions of the ENUR register include saving the ninth bit received in the data frame and enabling or disabling the UART's Wake-up Mode of operation. The determination of an internal or external clock source is done by the ENUI register, as well as selecting the number of stop bits and enabling or disabling transmit and receive interrupts.

The baud rate clock for the Receiver and Transmitter can be selected for either an internal or external source using two bits in the ENUI register. The internal baud rate is programmed by the DIVBY register. The baud rate may be selected from a range of 8 Hz to 128 kHz in binary steps or T3 underflow. By selecting a baud rate crystal, all standard baud rates from 75 baud to 38.4 kbaud can be generated. The external baud clock source comes from the CKX pin. The Transmitter and Receiver can be run at different rates by selecting one to operate from the internal clock and the other from an external source.

The HPC16400 UART supports two data formats. The first format for data transmission consists of one start bit, eight



TL/DD/8802–15

FIGURE 13. UART Block Diagram

HPC16400 UART (Continued)

data bits and one or two stop bits. The second data format or transmission consists of one start bit, nine data bits, and one or two stop bits. Receiving formats differ from transmission only in that the Receiver always requires only one stop bit in a data frame.

UART Wake-up Mode

The HPC16400 UART features a Wake-up Mode of operation. This mode of operation enables the HPC16400 to be networked with other processors. Typically in such environments, the messages consist of addresses and actual data. Addresses are specified by having the ninth bit in the data frame set to 1. Data in the message is specified by having the ninth bit in the data frame reset to 0.

The UART monitors the communication stream looking for addresses. When the data word with the ninth bit set is received, the UART signals the HPC16400 with an interrupt. The processor then examines the content of the receiver buffer to decide whether it has been addressed and whether to accept subsequent data.

Programmable Serial Decoder Interface

The programmable serial decoder interface allows the two HDLC channels to be used with devices employing several popular serial protocols for point-to-point and multipoint data exchanges. These protocols combine the 'B' and 'D' channels onto common pins—received data, transmit data, clock and Sync, which normally occurs at an 8 KHz rate and provides framing for the particular protocol.

The decoder uses the serial link clock and Sync signals to generate internal enables for the 'D' and 'B' channels, thereby allowing the HDLC channels to access the appropriate channel data from the multiplexed link.

HDLC Channel Description

HDLC/DMA Structure

HDLC 1		HDLC 2	
HDLC1 Receive	HDLC1 Transmit	HDLC2 Receive	HDLC2 Transmit
DMAR1	DMAT1	DMAR2	DMAT2

GENERAL INFORMATION

Both HDLC channels on the HPC16400 are identical and operate up to 4.1 Mbps. When used in an ISDN basic access application, HDLC channel #1 has been designated for use with the 16 Kbps D-channel or the B1 channel and HDLC #2 can be used with either of the 64 Kbps B-channels. If the 'D' and 'B' channels are present on a common serial link, the programmable serial decoder interface generates the necessary enable signals needed to access the D and B channel data.

LAPD, the Link Access Protocol for the D channel is derived from the X.25 packet switching LAPB protocol. LAPD specifies the procedure for a terminal to use the D channel for the transfer of call control or user-data information. The procedure is used in both point-to-point and point-to-multipoint configurations. On the 16400, the HDLC controller contains user programmable features that allow for the efficient processing of LAPD Information.

HDLC Channel Pin Description

RX — Receive Serial Data Input. Data clocked in on positive CLK edge.

CLK — HDLC Channel Clock Input Signal.

REN — HDLC Channel Receiver Enable Input.

TX — Transmit Serial Data Output. Data clocked out on negative CLK edge.

TEN — HDLC Channel Transmitter Enable Input.

HDLC Functional Description

TRANSMITTER DESCRIPTION

Data information is transferred from external memory through the DMA controller into the transmit buffer register from where it is loaded into a 8-bit serial shift registers. The CRC is computed and appended to the frame prior to the closing flag being transmitted. Data is output at the TX output pin. If no further transmit commands are given the transmitter sends out continous flags or the idle pattern as selected by the control register.

An interrupt is generated when the transmit shift register is empty or on a transmit error condition. An assoicated transmit status register will contain the status information indicating the specific interrupt source.

TRANSMITTER FEATURES

Interframe fill: the transmitter can send either continuous '1's or repeated flags between the closing flag or one packet and the opening flag of the next. When the CPU commands the transmitter to open a new frame, the interframe fill is terminated immediately.

Abort: the 7 '1's abort sequence will be immediately sent on command from the CPU. If required it may be followed by a new opening flag to resend the aborted packet.

Bit/Byte boundaries: The message length between packet headers may have any number of bits and is not confined to byte boundaries. Three bits in the control register are used to indicate the number of valid bits in the last byte when operating in the bit mode. These bits are loaded by the users software.

RECEIVER DESCRIPTION

Data is input to the receiver on the RX pin. The receive clock can be externally input at the HDLC CLK pin, or it can be internally generated via the programmable timer chain.

Incoming data is routed through one of several paths depending on whether it is the flag, data, or CRC.

Once the receiver is enabled it waits for the opening flag of the incoming frame, then starts the zero bit deletion, addressing handling and CRC checking. All data between the flags is shifted through two 8-bit serial shift registers before being loaded into the buffer register. The user programmable address register values are compared to the incoming data while it resides in the shift registers. If an address match occurs or if operating in the transparent mode, the DMA channel is signaled that attention is required and the byte is transferred by it to external memory. Appropriate interrupts are generated to the CPU on the reception of a complete frame as indicated by a correct CRC, or on the occurance of a frame error.

HDLC Functional Description

(Continued)

There are two sources for the receive channel enable signal. It can be internally generated from the serial interface or it can be externally enabled.

The receive interrupt, in conjunction with status data in the control registers allows interrupts to be generated on the following conditions—CRC error, receive error and receive complete.

RECEIVER FEATURES

Flag sharing: the closing flag of one packet may be shared as the opening flag of the next. Receiver will be able to share a zero between flags—011111101111110 is a valid two flag sequence for receive (not transmit).

Interframe fill: the receiver automatically accepts either repeated flags or all '1's as the interframe fill.

Idle: Reception of successive 1's as the interframe fill sequence to be signaled to the user by setting the Idle bit in the Receive control and status register.

Short Frame Rejection: Reception of less than 4 bytes between flags will generate a frame error, terminating reception of the current frame and setting the Frame Error (FER) status bit in the Receive Control and Status register.

Abort: the 7 '1's abort sequence (received with no zero insertion) will be immediately recognized and will cause the receiver to reinitialize and return to searching the incoming data for an opening flag. Reception of the abort will cause the abort status bit in the Interrupt Error Status register to be set.

Bit/Byte boundaries: The message length between packet headers may have any number of bits and it is not confined to byte boundaries. Three bits in the status register are used to indicate the number of valid bits in the last byte when operating in the bit mode.

Addressing: Two user programmable bytes are available to allow frame address recognition on the two bytes immediately following the opening flag. When the received address matches the programmed value(s), the frame is passed through to the DMA channel. If no match occurs, the received frame address information is disregarded and he receiver returns to searching for the next opening flag and the address recognition process starts anew.

Support is provided to allow recognition of the broadcast address sequence of seven consecutive 1's. Additionally, a transparent mode of operation is available where no address decoding is done.

HDLC INTERRUPT CONDITIONS

The end of message interrupt (EOM) indicates that a complete frame has been received or transmitted by the HDLC controller. Thus, there are four separate sources for this interrupt, two each from each HDLC channel. The Message Control Register contains the pending bits for each source.

The HDLC/DMA error interrupt groups several related error conditions. Error conditions from both transmit/receiver channels can cause this interrupt, and the possible sources each have a status bit in the below register that is set on the occurrence of an error. The bit must then be serviced by the user.

HDLC CHANNEL CLOCK

Each HDLC channel uses the rising edge of the clock to sample the receive data. Outgoing transmit data is shifted out on the falling edge of the external clock. The maximum data rate when using the externally provided clocks is 4.1 Mb/s.

CYCLIC REDUNDACY CHECK

There are two standard CRC codes used in generating the 16-bit Frame Check Sequence (FCS) that is appended to the end of the data frame. Both codes are supported and the user selects the error checking code to be used through software control (Configuration reg). The two error checking polynomials available are:

(1) CRC—16 (x16 + x15 + x2 + 1)

(2) CCITT CRC (x16 + x12 + x5 + 1)

LOOP BACK OPERATIONAL MODE

The user has the ability, by appropriately configuring the control registers, to internally route the transmitter output and the TX pin to the receiver input at the RX pin. The transmit clock would then be internally connected to the receive clock.

DMA Controller*

GENERAL INFORMATION

The HPC16400 uses Direct Memory Access (DMA) logic to facilitate data transfer between the 2 full Duplex HDLC channels and external packet RAM. There are four DMA channels to support the four individual HDLC channels. Control of the DMA channels is accomplished through registers which are configured by the CPU. These control registers define specific operation of each channel and changes are immediately reflected in DMA operation. In addition to individual control registers, a global control bit (MSS in Message Control Register) is available so that the HDLC channels may be globally controlled.

The DMA issues a bus request to the CPU when one or more of the individual HDLC channels request service. Upon receiving a bus acknowledge from the CPU, the DMA completes all requests pending and any requests that may have occurred during DMA operation before returning control to the CPU. If no further DMA transfers are pending, the DMA relinquishes the bus and the CPU can again initiate a bus cycle.

Four memory expansion bits have been added for each of the four channels to support data transfers into the expanded memory bank areas.

The DMA has priority logic for a DMA requesting service. The priorities are:

1st priority .....Receiver channel 1

2nd priority .....Transmit channel 1

3rd priority .....Receive channel 2

4th priority .....Transmit channel 2

RECEIVER DMA OPERATION

A receiver DMA operation is initiated by the Buffer register. Once a byte has been placed in the Buffer register from the HDLC, it generates a request and upon obtaining control of the bus, the DMA places the byte in external memory.

RECEIVER REGISTERS

All the following registers are Read/Write

A. Frame Length Register

This user programmable 16-bit register contains the maximum number of bytes to be placed in a data "block". If

*The specific registers and/or register names may have changed. Please contact the factory for updated information.

DMA Controller (Continued)

this number is exceeded, a Frame Too Long (FTLR1, FTLR2) error is generated. This register is decremented by one each Receiver DMA cycle.

3. CNTRL ADDR 1
   DATA ADDR 1
   CNTRL ADDR 2
   DATA ADDR 2

   The CNTRL ADDR register contains the external memory address where the Frame Header (Control & Address fields) are to be stored and the DATA ADDR register contains an equivalent address for the Information field.

TRANSMITTER DMA OPERATION

A transmitter DMA cycle is initiated by the TX Data Buffer (TDB). The TX Data Buffer generates a request when empty and the DMA responds by placing a Byte in the TDB. The HDLC transmitter can then accept the Byte to send when needed, upon which the TDB will issue another request, resulting in a subsequent DMA cycle.

TRANSMITTER REGISTERS

The following registers are Read/Write:

1. Field Address 1 (FA1)
   # Bytes Field 1 (NBF1)
   Field Address (FA2)
   # Bytes Field 2 (NBF2)

   FA1 and FA2 are starting addresses of blocks of information to transmitter.

   NBF1 and NBF2 are the number of bytes in the block to be transmitted starting the FA1.

The following registers are Read only:

3. Working Field Address (WFA)
   Working Number of Bytes (WNB)

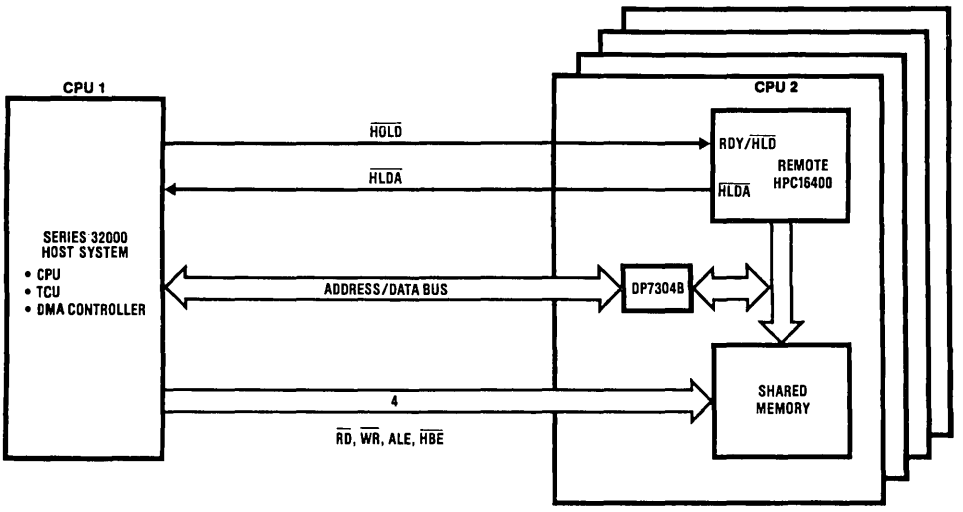
   WFA is the present value of the Field address and will be the next memory location where the next byte will be accessed. WNB is the present value of the number of bytes to be transmitted. The value will be the number of bytes to be fetched from memory before the block information transfer is completed. On each transmit DMA cycle, WFA is incremented and WNB is decremented.

Shared Memory Support

Shared memory access provides a rapid technique to exchange data. It is effective when data is moved from a peripheral to memory or when data is moved between blocks of memory. A related area where shared memory access proves effective in multiprocessing applications where two CPUs share a common memory block. The HPC16400 supports shared memory access with two pins. The pins are the RDY/$\overline{HLD}$ input pin and the $\overline{HLDA}$ output pin. The user can software select either the Hold or Ready function by the state of a control bit. The $\overline{HLDA}$ output is multiplexed onto port B.

The host uses DMA to interface with the HPC16400. The host initiates a data transfer by activating the $\overline{HLD}$ input of the HPC16400. In response, the HPC16400 places its system bus in a TRI-STATE Mode, freeing it for use by the host. The host waits for the acknowledge signal ($\overline{HLDA}$) from the HPC16400 indicating that the sytem bus is free. On receiving the acknowledge, the host can rapidly transfer data into, or out of, the shared memory by using a conventional DMA controller. Upon completion of the message transfer, the host removes the HOLD request and the HPC16400 resumes normal operations.

Figure 14 illustrates an application of the shared memory interface between the HPC16400 and a Series 32000 system.



TL/DD/8802-16

FIGURE 14. Shared Memory Application: HPC16400 Interface to Series 32000 System

Memory

The HPC16400 has been designed to offer flexibility in memory usage. A total address space of 64 kbytes can be addressed with 256 bytes of RAM available on the chip itself.

Program memory addressing is accomplished by the 16-bit program counter on a byte basis. Memory can be addressed directly by instructions or indirectly through the B, X and SP registers. Memory can be addressed as words or bytes. Words are always addressed on even-byte boundaries. The HPC16400 uses memory-mapped organization to support registers, I/O and on-chip peripheral functions.

The HPC16400 memory address space extends to 64 kbytes and registers and I/O are mapped as shown in Table II.

Extended Memory Addressing

If more than 64k of addressing is desired in a HPC16400 system, on board bank select circuitry is available that allows four I/O lines of Port B (B9, B10, B13, B14) to be used in extending the address range. This gives the user a main routine area of 32k and 16 banks of 32k each for subroutine and data, thus getting a total of 544k of memory.

The Extended Memory Addressing mode is entered by setting the EMA control bit in the Message Control Register. If this bit is not set, the port B lines (B9, B10, B13, B14) are available as general purpose I/O or synchronous outputs as selected by the BFUN register.

The main memory area contains the interrupt vectors & service routines, stack memory, and common memory for the bank subroutines to use. The 16 banks of memory can contain program or data memory (note- since the on chip resources are mapped into addresses 0000-01FF, the first 512 bytes of each bank are not usable).

TABLE II. Memory Map*

FFFF:FFF0 FFEF:FFD0	Interrupt Vectors JSRP Vectors	
FFCF:FFCE : : : : 0201:0200	External expansion Memory	USER MEMORY
01FF:01FE : : 01C1:01C0	On Chip RAM	
01BB:01BA 01B9:01B8 01B7:01B6 01B5:01B4 01B3:01B2 01B1:01B0	Configuration Reg Rec Addr Comp Reg 2 Rec Addr Comp Reg 1 Interrupt Error Stat Xmit Cntrl & Status Recv Cntrl & Status	HDLC # 2
01AB:01AA 01A9:01A8 01A7:01A6 01A5:01A4 01A3:01A2 01A1:01A0	Configuration Reg Rec Addr Comp Reg 2 Rec Addr Comp Reg 1 Interrupt Error Stat Xmit Cntrl & Status Recv Cntrl & Status	HDLC # 1
0195:0194	Watch Dog Address	Watch Dog Logic
0193:0192 0191:0190 018F:018E 018D:018C 018B:018A 0189:0188 0187:0186 0185:0184 0183:0182 0181:0180	T0CON Register TMMODE Register DIVBY Register T3 Timer R3 Register T2 Timer R2 Register I2CR Register/ R1 I3CR Register/ T1 I4CR Register	Timer Block T0:T3
017D:017C 017B:017A 0179:0178 0177:0176 0175:0174 0173:0172 0171:0170	Working # Bytes Working Field Addr # Bytes 2 Field Addr 2 # Bytes 1 Field Addr 1 Xmit Cntrl & Status	DMAT # 2 (Xmit)
016B:016A 0169:0168 0167:0166 0165:0164 0163:0162 0161:0160	Frame Length Data Addr 2 Cntrl Addr 2 Data Addr 1 Cntrl Addr 1 Recv Cntrl & Status	DMAR # 2 (Recv)
015D:015C 015B:015A 0159:0158 0157:0156 0155:0154 0153:0152 0151:0150	Working # Bytes Working Field Addr # Bytes 2 Field Addr 2 # Bytes 1 Field Addr 1 Xmit Cntrl & Status	DMAT # 1 (Xmit)
014B:014A 0149:0148 0147:0146 0145:0144 0143:0142 0141:0140	Frame Length Data Addr 2 Cntrl Addr 2 Data Addr 1 Cntrl Addr 1 Recv Cntrl & Status	DMAR # 1 (Recv)
0131:0130	Message Control	
0128 0126 0124 0122 0120	ENUR register TBUF register RBUF register ENIU register ENU register	UART
0107:0106 0105:0104 0103:0102 0101:0100	DIR R register Port R register Serial Decoder Port D register	PORTS
00F5:00F4 00F3:00F2 00F1:00F0	BFUN register DIR B register DIR A register	PORTS A & B CONTROL
00E7:00E6	Reserved	
00E3:00E2 00E1:00E0	Port B Port A	PORTS A & B
00DE 00DD:00DC 00D8 00D6 00D4 00D2 00D0	Microcode ROM dump Halt Enable register Port I input register SIO register IRCD register IRPD register ENIR register	PORT CONTROL & INTERRUPT CONTROL REGISTERS
00CF:00CE 00CD:00CC 00CB:00CA 00C9:00C8 00C7:00C6 00C5:00C4 00C3:00C2 00C1:00C0	X register B register K register A register PC register SP register (reserved) PSW register	HPC10640 CORE REGISTERS
00BF:00BE : : 0001:0000	On Chip RAM	USER RAM

*The Memory Map has changed. Please contact factory for an updated version of the Memory Map.

HPC16400 CPU

The HPC16400 CPU has a 16-bit ALU and six 16-bit registers.

Arithmetic Logic Unit (ALU)

The ALU is 16 bits wide and can do 16-bit add, subtract and shift or logic AND, OR and exclusive OR in one timing cycle. The ALU can also output the carry bit to a 1-bit C register.

Accumulator (A) Register

The 16-bit A register is the source and destination register for most I/O, arithmetic, logic and data memory access operations.

Address (B and X) Registers

The 16-bit B and X registers can be used for indirect addressing. They can automatically count up or down to sequence through data memory.

Boundary (K) Register

The 16-bit K register is used to set limits in repetitive loops of code as register B sequences through data memory.

Stack Pointer (SP) Register

The 16-bit SP register is the stack pointer that addresses the stack. The SP register is incremented by two for each push or call and decremented by two for each pop or return. The stack can be placed anywhere in user memory and be as deep as the available memory permits.

Program (PC) Register

The 16-bit PC register addresses program memory.

Addressing Modes

ADDRESSING MODES—ACCUMULATOR AS DESTINATION

Register Indirect

This is the "normal" mode of addressing for the HPC16400 (instructions are single-byte). The operand is the memory addressed by the B register (or X register for some instructions).

Direct

The instruction contains an 8-bit or 16-bit address field that directly points to the memory for the operand.

Indirect

The instruction contains an 8-bit address field. The contents of the WORD addressed points to the memory for the operand.

Indexed

The instruction contains an 8-bit address field and an 8- or 16-bit displacement field. The contents of the WORD addressed is added to the displacement to get the address of the operand.

Immediate

The instruction contains an 8-bit or 16-bit immediate field that is used as the operand.

Register Indirect (Auto Increment and Decrement)

The operand is the memory addressed by the X register. This mode automatically increments or decrements the X register (by 1 for bytes and by 2 for words).

Register Indirect (Auto Increment and Decrement) with Conditional Skip

The operand is the memory addressed by the B register. This mode automatically increments or decrements the B register (by 1 for bytes and by 2 for words). The B register is then compared with the K register. A skip condition is generated if B goes past K.

ADDRESSING MODES—DIRECT MEMORY AS DESTINATION

Direct Memory to Direct Memory

The instruction contains two 8- or 16-bit address fields. One field directly points to the source operand and the other field directly points to the destination operand.

Immediate to Direct Memory

The instruction contains an 8- or 16-bit address field and an 8- or 16-bit immediate field. The immediate field is the operand and the direct field is the destination.

Double Register Indirect using the B and X Registers

Used only with Reset, Set and IF bit instructions; a specific bit within the 64 kbyte address range is addressed using the B and X registers. The address of a byte of memory is formed by adding the contents of the B register to the most significant 13 bits of the X register. The specific bit to be modified or tested within the byte of memory is selected using the least significant 3 bits of register X.

HPC Instruction Set Description

Mnemonic	Description	Action
ARITHMETIC INSTRUCTIONS		
ADD	Add	MA + Meml → MA carry → C
ADC	Add with carry	MA + Meml + C → MA carry → C
DADC	Decimal add with carry	MA + Meml + C → MA (Decimal) carry → C
SUBC	Subtract with carry	MA − Meml + C → MA carry → C
DSUBC	Decimal subtract w/carry	MA − Meml + C → MA (Decimal) carry → C
MULT	Multiply (unsigned)	MA*Meml → MA & X, 0 → K, 0 → C
DIV	Divide (unsigned)	MA/Meml → MA, rem. → X, 0 → K, 0 → C
IFEQ	If equal	Compare MA & Meml, Do next if equal
IFGT	If greater than	Compare MA & Meml, Do next if MA → Meml
AND	Logical and	MA and Meml → MA
OR	Logical or	MA or Meml → MA
XOR	Logical exclusive-or	MA xor Meml → MA
MEMORY MODIFY INSTRUCTIONS		
INC	Increment	Mem + 1 → Mem
DECSZ	Decrement, skip if 0	Mem −1 → Mem, Skip next if Mem = 0
BIT INSTRUCTIONS		
SET	Set bit	1 → Mem.bit (bit is 0 to 7 immediate)
RESET	Reset bit	0 → Mem.bit
IF	If bit	If Mem.bit is true, do next instr.
MEMORY TRANSFER INSTRUCTIONS		
LD	Load	Meml → MA
	Load, incr/decr X	Mem(X) → A, X ±1 (or 2) → X
ST	Store to Memory	MA → Mem
X	Exchange	A ↔ Mem; Mem ↔ Mem
	Exchange, incr/decr X	A ↔ Mem(X), X ±1 (or 2) → X
PUSH	Push Memory to Stack	W → W(SP), SP + 2 → SP
POP	Pop Stack to Memory	SP − 2 → SP, W(SP) → W
LDS	Load A, incr/decr B, Skip on condition	Mem(B) → A, B ±1 (or 2) → B, Skip next if B greater/less than K
XS	Exchange, incr/decr B, Skip on condition	Mem(B) ↔ A,B ±1 (or 2) → B, Skip next if B greater/less than K
REGISTER LOAD IMMEDIATE INSTRUCTIONS		
LD A	Load A immediate	imm → A
LD B	Load B immediate	imm → B
LD K	Load K immediate	imm → K
LD X	Load X immediate	imm → X
LD BK	Load B and K immediate	imm → B,imm' → K
ACCUMULATOR AND C INSTRUCTIONS		
CLR A	Clear A	0 → A
INC A	Increment A	A + 1 → A
DEC A	Decrement A	A − 1 → A
COMP A	Complement A	1's complement of A → A
SWAP A	Swap nibbles of A	A15:12 ← A11:8 ← A7:4 ↔ A3:0
RRC A	Rotate A right thru C	C → A15 → ... → A0 → C
RLC A	Rotate A left thru C	C ← A15 ← ... ← A0 ← C
SHR A	Shift A right	0 → A15 → ... → A0 → C
SHL A	Shift A left	C ← A15 ← ... ← A0 ← 0
SET C	Set C	1 → C
RESET C	Reset C	0 → C
IF C	IF C	Do next if C = 1
IFN C	IF not C	Do next if C = 0

HPC Instruction Set Description (Continued)

Mnemonic	Description	Action
TRANSFER OF CONTROL INSTRUCTIONS		
JSRP	Jump subroutine from table	PC → W(SP),SP+2 → SP W(table#) → PC
JSR	Jump subroutine relative	PC → W(SP),SP+2 → SP,PC+ # → PC (#is +1024 to −1023)
JSRL	Jump subroutine long	PC → W(SP),SP+2 → SP,PC+ # → PC
JP	Jump relative short	PC+ # → PC(# is +32 to −31)
JMP	Jump relative	PC+ # → PC(#is +256 to −255)
JMPL	Jump relative long	PC+ # → PC
JID	Jump indirect at PC + A	PC+A+1 → PC
JIDW		then Mem(PC)+PC → PC
NOP	No Operation	PC ← PC + 1
RET	Return	SP−2 → SP,W(SP) → PC
RETS	Return then skip next	SP−2 → SP,W(SP) → PC, & skip
RETI	Return from interrupt	SP−2 → SP,W(SP) → PC, interrupt re-enabled

Note: W is 16-bit word of memory
MA is Accumulator A or direct memory (8 or 16-bit)
Mem is 8-bit byte or 16-bit word of memory
MemI is 8- or 16-bit memory or 8 or 16-bit immediate data
imm is 8-bit or 16-bit immediate data

Memory Usage

Number Of Bytes For Each Instruction (number in parenthesis is 16-Bit field)

	Using Accumulator A						To Direct Memory			
	Reg Indir. (B)	Reg Indir. (X)	Direct	Indir	Index	Immed.	Direct *	Direct **	Immed. *	Immed. **
LD	1	1	2(4)	3	4(5)	2(3)	3(5)	5(6)	3(4)	5(6)
X	1	1	2(4)	3	4(5)	—	—	—	—	—
ST	1	1	2(4)	3	4(5)	—	—	—	—	—
ADC	1	2	3(4)	3	4(5)	4(5)	4(5)	5(6)	4(5)	5(6)
SBC	1	2	3(4)	3	4(5)	4(5)	4(5)	5(6)	4(5)	5(6)
DADC	1	2	3(4)	3	4(5)	4(5)	4(5)	5(6)	4(5)	5(6)
DSBC	1	2	3(4)	3	4(5)	4(5)	4(5)	5(6)	4(5)	5(6)
ADD	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
MULT	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
DIV	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
IFEQ	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
IFGT	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
AND	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
OR	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)
XOR	1	2	3(4)	3	4(5)	2(3)	4(5)	5(6)	4(5)	5(6)

*8-bit direct address
**16-bit direct address

Instructions that modify memory directly

	(B)	(X)	Direct	Indir	Index	B&X
SET	1	2	3(4)	3	4(5)	1
RESET	1	2	3(4)	3	4(5)	1
IF	1	2	3(4)	3	4(5)	1
DDSZ	3	3	2(4)	3	4(5)	
INCD	3	3	2(4)	3	4(5)	

Immediate Load Instructions

	Immed.
LD B,*	2(3)
LD X,*	2(3)
LD K,*	2(3)
LD BK,*,*	3(5)

Memory Usage (Continued)

Register Indirect Instructions with Auto Increment and Decrement

Register B With Skip		
	(B+)	(B−)
LDS A,*	1	1
XS A,*	1	1

Register X		
	(X+)	(X−)
LD A,*	1	1
X A,*	1	1

Instructions Using A and C

CLR	A	1
INC	A	1
DEC	A	1
COMP	A	1
SWAP	A	1
RRC	A	1
RLC	A	1
SHR	A	1
SHL	A	1
SET	C	1
RESET	C	1
IF	C	1
IFN	C	1

Transfer of Control Instructions

JSRP	1
JSR	2
JSRL	3
JP	1
JMP	2
JMPL	3
JID	1
JIDW	1
NOP	1
RET	1
RETS	1
RETI	1

Stack Reference Instructions

	Direct
PUSH	2
POP	2

Code Efficiency

One of the most important criteria of a single chip microcontroller is code efficiency. The more efficient the code, the more features that can be put on a chip. The memory size on a chip is fixed so if code is not efficient, features may have to be sacrificed or the programmer may have to buy a larger, more expensive version of the chip.

The HPC16400 has been designed to be extremely code-efficient. The HPC16400 looks very good in all the standard coding benchmarks; however, it is not realistic to rely only on benchmarks. Many large jobs have been programmed onto the HPC16400, and the code savings over other popular microcontrollers has been considerable—often the jobs take less than one-half the memory!

Reasons for this saving of code include the following:

SINGLE BYTE INSTRUCTIONS

The majority of instructions on the HPC16400 are single-byte. There are two especially code-saving instructions:

JP is a 1-byte jump. True, it can only jump within a range of plus or minus 32, but many loops and decisions are often within a small range of program memory. Most other micros need 2-byte instructions for any short jumps.

JSRP is a 1-byte call subroutine. The user makes a table of his 16 most frequently called subroutines and these calls will only take one byte. Most other micros require two and even three bytes to call a subroutine. The user does not have to decide which subroutine addresses to put into his table; the assembler can give him this information.

EFFICIENT SUBROUTINE CALLS

The 2-byte JSR instructions can call any subroutine within plus or minus 1k of program memory.

MULTIFUNCTION INSTRUCTIONS FOR DATA MOVEMENT AND PROGRAM LOOPING

The HPC16400 has single-byte instructions that perform multiple tasks. For example, the XS instruction will do the following:

1. Exchange A and memory pointed to by the B register
2. Increment the B register
3. Compare the B register versus the K register
4. Generate a conditional skip if B is greater than K

The value of this multipurpose instruction becomes evident when looping through sequential areas of memory and exiting when the loop is finished.

BIT MANIPULATION INSTRUCTIONS

Any bit of memory, I/O or registers can be set, reset or tested by the single byte bit instructions. The bits can be addressed directly or indirectly. Since all registers and I/O are mapped into the memory, it is very easy to manipulate specific bits to do efficient control.

DECIMAL ADD AND SUBTRACT

This instruction is needed to interface with the decimal user world.

It can handle both 16-bit words and 8-bit bytes.

The 16-bit capability saves code since many variables can be stored as one piece of data and the programmer does not have to break his data into two bytes. Many applications store most data in 4-digit variables. The HPC16400 supplies 8-bit byte capability for 2-digit variables and literal variables.

MULTIPLY AND DIVIDE INSTRUCTIONS

The HPC16400 has 16-bit multiply and divide instructions. This saves both code and time. Multiply and divide can use immediate data or data from memory. The ability to multiply and divide by immediate data saves code since this function is often needed for scaling, base conversion, computing indexes of arrays, etc.

Development Support

The MOLE (Microcontroller On-Line Emulator) is a low cost development system and emulator for all microcontroller products. These include COPs, TMP, 8050U and the HPC Family of Products. The MOLE consists of a BRAIN Board, Personality Board and optional host software.

The purpose of a MOLE is to provide the user with a tool to write and assemble code, emulate code for the target microcontroller and assist in both the software & hardware debugging of the system.

It is a self-contained computer with its own firmware which provides for all system operation, emulation control, communication, PROM programming and diagnostic operations.

It contains three serial ports because multiple ports are usually needed to optionally connect to a terminal, a host system, a printer or modem, or to connect to other MOLEs in a multi-MOLE environment.

MOLE can be used in either a stand alone mode or in conjunction with selected host systems, i.e., those using CP/M or PC-DOS. Communicating via RS-232 port.

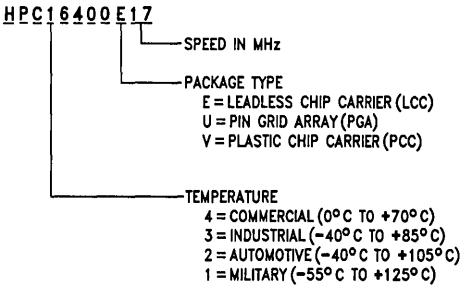
Dial-A-Helper is a service provided by the MOLE applications group. If a user is having difficulty in getting a MOLE to operate in a particular mode or it is acting peculiar, he can contact us via his system and a modem. He can leave messages on our electronic bulletin board which we will respond to, or he can arrange for us to actually take control of his system via modem for debugging purposes.

The applications group can then cause his system to execute various commands and try to resolve the customer's problem by actually getting the customer's system to respond. 99% of the time the problem is resolved. This allows us to respond in minutes instead of days when applications help is needed.

Part Selection

The HPC family includes devices with many different options and configurations to meet various application needs. The number HPC16400 has been generally used throughout this datasheet to represent the whole family of parts. The following chart explains how to order various options available when ordering HPC family members.

Note: All options may not currently be available.



TL/DD/8802–18

FIGURE 15. HPC Family Part Numbering Scheme

Examples

HPC46400V17—Commercial temp (0° to +70°C), PCC

HPC16400E17—Military temp (−55°C to +125°C), LCC

5

National Semiconductor Corporation

PRELIMINARY

HPC16740/HPC36740/HPC46740/ HPC16730/HPC36730/HPC46730 High-Performance Microcontroller with Gate Array

General Description

The HPC16740 is a member of the HPC™ family of High Performance Microcontrollers. Each member of the family has the same identical core CPU with a unique memory and I/O configuration to suit specific applications. Each part is fabricated in National's advanced microCMOS technology. This process combined with an advanced architecture provides fast, flexible I/O control, efficient data manipulation, and high speed computation.

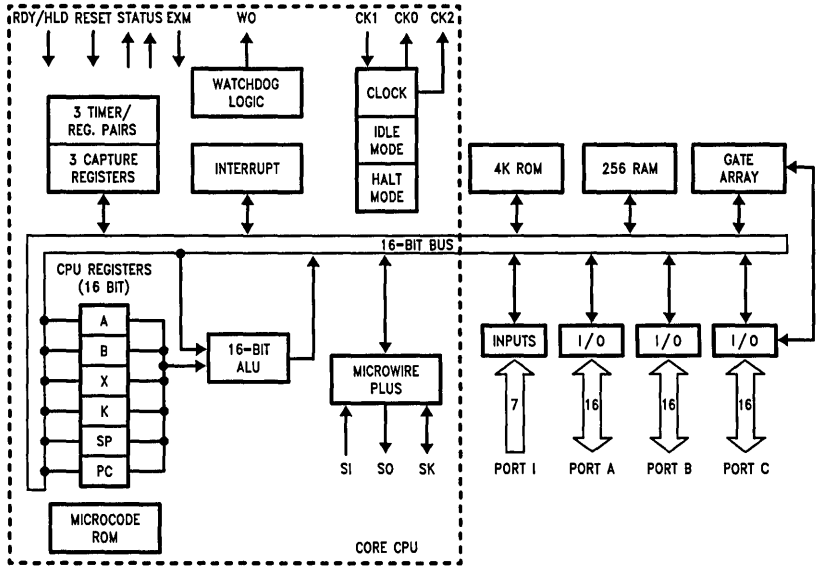
The HPC16740 contains an on-chip gate array giving the user the ability to customize the logic function of his chip. With it the designer can replace external glue logic or implement a complex device like a digital phase lock loop (PLL). The gate array peripheral includes: 1) 280 user programmable gates, 2) 650 gates making 6 8-bit configurable functions consisting of; counters, shift registers, latches or D flip-flops 3) 270 gates for built-in logic to interface to the HPC core and Port C I/O. This offers an on-chip gate array comparable to an external array of about 1200 gates.

The HPC16740 is available in 68-pin PCC, LCC and PGA packages.

Features

- HPC family—core features:
  - — 16-bit architecture, both byte and word
  - — 16-bit data bus, ALU, and registers
  - — 64k bytes of external memory addressing
  - — FASTI—240 ns for register instructions when using 17.0 MHz clock
  - — High code efficiency—Most instructions are single byte
  - — 16 x 16 multiply and 32 x 16 divide
  - — Eight vectored interrupt sources
  - — Four 16-bit timer/counters with 3 input capture registers and 4 synchronous outputs
  - — WATCHDOG logic monitors processor
  - — MICROWIRE/PLUS serial I/O interface
  - — CMOS—very low power with two power save modes: IDLE and HALT (2 mA, 250 μA—typ.)
- 1200 equivalent gates
- 55 general purpose I/O lines (memory mapped)
- 4k bytes of ROM, 256 bytes of RAM on chip
- ROMless versions available
- Wide voltage supply range: 3V to 5.5V
- Industrial (−40°C to +85°C) and military (−55°C to +125°C) temperature ranges

Block Diagram



TL/DD/8803-1

National Semiconductor Corporation

PRELIMINARY

HPC16900/HPC36900/HPC46900 Port Expansion and Recreation Logic

General Description

The HPC16900 is a support chip for the HPC family of High Performance Microcontrollers. It is fabricated using National's advanced microCMOS technology.

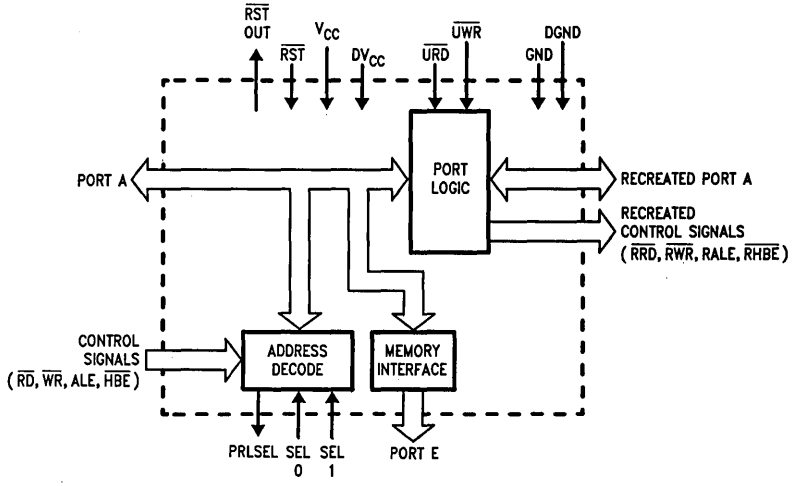
The HPC16900 is a Port Expansion and Recreation Logic (PEARL) chip intended for use as a port expander when the user requires Port A on the HPC microcontroller to serve as an address/data bus. It recreates the entire Port A and 4 bits of Port B. In optional mode, it provides an additional 16-bit I/O port.

The HPC16900 is available in 68-pin PCC, LCC and PGA packages.

Features

- Supports 17.0 MHz HPC operation
- Supports all operating modes of Port A
- Multiplexed Address/Data bus
- Supports UPI (Universal Peripheral Interface)
- Provides up to 36 I/O pins in expanded mode
- Up to 4 PEARL chips may be used in parallel
- Provides direct interface to external memory
- CMOS—very low power consumption
- Wide voltage supply range: 3 to 5.5V
- Industrial (−40°C to +85°C) and military (−55°C to +125°C) temperature ranges

Block Diagram



TL/DD/9122-1

5

Section 6

NS8050 Family

Section 6 Contents

8050

The 8050 Family

These products include the 8048, 8049 and 8050U. The 8050U provides MICROWIRE/PLUS™ for serial communication. Features include:

- Self-contained system timing
- Control logic
- RAM program memory
- RAM data memory
- 27 I/O lines (used to implement dedicated control functions)
- 8048 instruction set
- Memory capacity—1k x 8 ROM and 64 x 8 RAM to 4k x 8 ROM and 256 x 8 RAM
- Low-cost emulators—NS87P50 Piggyback devices

MICROWIRE™ Peripherals

The variety of functions available through serial communication are:

- Memory (CMOS RAMs and EEPROMs)
- Display Drivers (LCD, LED, VF)
- A/D converters
- Timer/Counters
- AM/FM digital phase-locked loop frequency synthesizer

The 8050 Family is the industry standard microcontroller.

National Semiconductor Corporation

PRELIMINARY

INS8048/8049/8048L/8049L/8048I/8049I NS87P50/8050U/8040U/87P50U/8050UI/8050UL Microcomputer/Microprocessor Family

General Description

The members of this family of microcomputers are self-contained, 8-bit parallel, 40-pin dual-in-line devices fabricated using National Semiconductor's scaled N-channel, silicon gate MOS process, XMOS™. The 48-Series devices contain the system timing, control logic, ROM (where applicable) program memory, RAM data memory and 27 I/O lines necessary to implement dedicated control functions. All 48-Series devices are pin-compatible, differing only in the size of on-board ROM (where applicable) and RAM as shown below.

For applications requiring microwire serial communication, the NS8050U provides this feature.

MICROWIRE/PLUS™ consists basically of a three-wire communication port with a clocked 8-bit shift register. The three lines consist of a Serial Output (SO), a Serial Input (SL), and a Serial Clock (SK). The shift register is referred to as the serial input/output register. One 8050U must be designated as the master. The master supplies the clock for the MICROWIRE/PLUS system and initiates all data transfers. All transfers are between the master and one or more slaves. A slave may be any MICROWIRE™ peripheral or another 8050U with MICROWIRE/PLUS. MICROWIRE/PLUS communicates with a variety of MICROWIRE peripherals, such as the COP472 LCD Display Driver, COP494 EEPROM, or other 8050's configured as a peripheral. MICROWIRE/PLUS makes efficient use of the I/O lines. Thus, the MICROWIRE/PLUS expands the capability of the 8050U family.

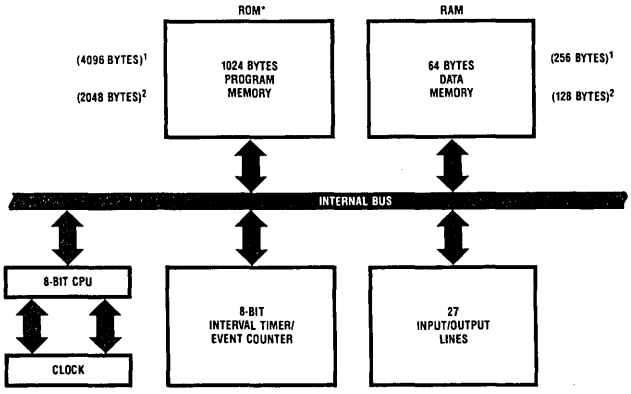
*Supersedes all data sheets and data books.

Features

- The master/slave feature is programmable
- NS87P50 MICROWIRE/PLUS Piggy-back Microcomputer
- NS8040U MICROWIRE/PLUS ROM-less
- NS8050U MICROWIRE/PLUS in 40-pin package
- Transparent enhancement to the standard 8050 Microcomputer
- Ready interface to the MICROWIRE peripheral family
- Testable shifter "Done" flag available
- Selectable shift rate
- Two new MICROWIRE/PLUS control instructions
- Serial data exchange with only three wires
- 8-bit timer/control
- Binary and BCD arithmetic
- Single 5V power supply
- Low standby power
- Low voltage standby
- Expandable memory and I/O
- 1.36 μs cycle, 11 MHz clock

Device	RAM Array	ROM Array
INS8048, 48L, 48I	64 x 8	1k x 8
INS8049, 49L, 48I	128 x 8	2k x 8
NS8050U, 50UL, 50UI	256 x 8	4k x8
INS8035, 35L, 35I	64 x 8	N/A
INS8039, 39L, 39I	128 x 8	N/A
NS8040U, 40UL, 40UI	256 x 8	N/A
NS87P50U	64/128/256 x 8	1k/2k/4k x 8

Block Diagram



TL/C/5488–1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

	87P50 Series 8048 Series 8048L Series	8048I Series
Temperature Under Bias	−20°C to +85°C	−50°C to +100°C
Storage Temperature	−65°C to +150°C	−65°C to +150°C
All Input or Output Voltages with Respect to V_{SS}	−0.5V to +7V	−0.5V to +7V
Power Dissipation	1.5W	1.5W

Note: Absolute Maximum Ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under DC Electrical Characteristics.

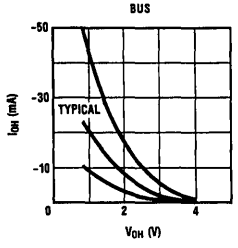
Features	INS8048 Series	INS8048L Series	INS8048I Series	NS87P50 Series
■ −40 to +85°C Operation			X	
■ 8-Bit CPU, RAM, ROM, I/O in a Single Package	X	X	X	
■ 2.5 μs Cycle, 6 MHz Clock; 1.36 μs Cycle, 11 MHz Clock	X	X	X	X
■ Very Low Power, High Speed Operation		X		
■ On-Chip Oscillator Circuit and Clock (or External Source)	X	X	X	X
■ 27 I/O Lines	X	X	X	X
■ Expandable Memory and I/O	X	X	X	X
■ 8-Bit Timer/Counter	X	X	X	X
■ Single-Level Interrupt	X	X	X	X
■ 96 Instructions (Most Single-Byte)	X	X	X	X
■ Binary and BCD Arithmetic	X	X	X	X
■ Single +5V Power Supply	X	X	X	X
■ Low Standby Power Mode	X	X	X	X
■ Low Voltage Standby	2.2V Min	2.2V Min	2.2V Min	2.6V Min

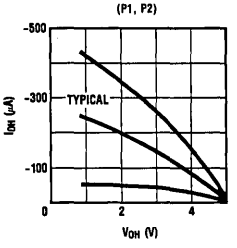
DC Electrical Characteristics

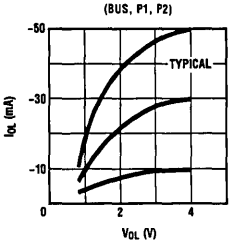
T_A = 0°C to +70°C, V_{CC} = +5V ±10%, V_{SS} = 0V, unless otherwise specified. (NS8048, INS8048L)
T_A = 40°C to +85°C, V_{CC} = +5V ±10%, V_{SS} = 0V, unless otherwise specified. (NS8048I) (Note 2)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
V_{IL}	Input Low Voltage (Except XTAL1, XTAL2, $\overline{RESET}$)		−0.5		0.8	V
V_{IL1}	Input Low Voltage (XTAL1, XTAL2, $\overline{RESET}$)		−0.5		0.6	V
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$)		2.0		V_{CC}	V
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$)		3.8		V_{CC}	V
V_{OL}	Output Low Voltage (Bus)	I_{OL} = 2 mA			0.45	V
V_{OL1}	Output Low Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	I_{OL} = 1.8 mA			0.45	V
V_{OL2}	Output Low Voltage ($\overline{PROG}$)	I_{OL} = 1.0 mA			0.45	V
V_{OL3}	Output Low Voltage (Ports and Others)	I_{OL} = 1.6 mA			0.45	V
V_{OH}	Output High Voltage (Bus)	I_{OH} = −400 μA	2.4			V
V_{OH1}	Output High Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	I_{OH} = −100 μA	2.4			V
V_{OH2}	Output High Voltage (Ports and Others)	I_{OH} = −40 μA	2.4			V
I_{LI}	Input Leakage Current (T1, $\overline{INT}$, EA)	$V_{SS} \le V_{IN} \le V_{CC}$			±10	μA
I_{LI1}	Input Leakage Current Ports	V_{SS} + 0.45 ≤ $V_{IN} \le V_{CC}$			−500	μA
I_{LI2}	Input Leakage Current ($\overline{SS}$, $\overline{RESET}$)	V_{SS} + 0.45 ≤ $V_{IN} \le V_{CC}$			−300	μA
I_{LO}	Output Leakage Current (Bus, T_0) High Impedance State	V_{SS} + 0.45 ≤ $V_{IN} \le V_{CC}$			±10	μA
I_{DD} 64	Standby Current (Note 1)	8048, 8048I, 8048L			5/4	mA
I_{DD} 128	Standby Current (Note 1)	8049, 8049I, 8049L			7/5	mA
I_{DD} 256	Standby Current (Note 1)	8050, 8050I, 8050L			10/10	mA
I_{DD} + I_{CC}	Total Supply (Note 1)	8048, 8048I, 8048L			65/55	mA
I_{DD} + I_{CC}	Total Supply (Note 1)	8049, 8049I, 8049L			70/60	mA
I_{DD} + I_{CC}	Total Supply (Note 1)	8050, 8050I, 8050L			80/65	mA
V_{DD}	Standby Power Supply (8048, 8048I, 8048L Series)		2.2		V_{CC}	V

Note 1: For industrial device T_A = −40°C to +85°C.
Note 2: Combination of low power and industrial temperature devices not accepted.







TL/C/5488–2

AC Characteristics $T_A = 0°C$ to $+70°C$ ($-40°C$ to $+85°C$ for 8048I), $V_{CC} = V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$

Symbol	Parameter	f(t_{CY}) (Note 3)	11 MHz		Units
			Min	Max	
t_{LL}	ALE Pulse Width (Note 1)	7/30 t_{CY} − 170	150		ns
t_{AL}	Address Setup to ALE (Note 1)	2/15 t_{CY} − 110	70		ns
t_{LA}	Address Hold from ALE (Note 1)	1/15 t_{CY} − 40	50		ns
t_{CC1}	Control Pulse Width ($\overline{RD}$, $\overline{WR}$) (Note 1)	1/2 t_{CY} − 200	480		ns
t_{CC2}	Control Pulse Width ($\overline{PSEN}$) (Note 1)	2/5 t_{CY} − 200	350		ns
t_{DW}	Data Setup Before $\overline{WR}$ (Note 1)	13/30 t_{CY} − 200	390		ns
t_{WD}	Data Hold After $\overline{WR}$ (Notes 1, 2)	1/15 t_{CY} − 50	40		ns
t_{DR}	Data Hold ($\overline{RD}$, $\overline{PSEN}$) (Notes 1, 4)	1/10 t_{CY} − 30	0	110	ns
t_{RD1}	$\overline{RD}$ to Data in (Note 1)	2/5 t_{CY} − 170		370	ns
t_{RD2}	$\overline{PSEN}$ to Data in (Note 1)	3/10 t_{CY} − 170		240	ns
t_{AW}	Address Setup to $\overline{WR}$ (Note 1)	1/3 t_{CY} − 150	300		ns
t_{AD1}	Address Setup to Data ($\overline{RD}$) (Note 1)	21/30 t_{CY} − 220		730	ns
t_{AD2}	Address Setup to Data ($\overline{PSEN}$) (Note 1)	1/2 t_{CY} − 200		480	ns
t_{AFC1}	Address Float to $\overline{RD}$, $\overline{WR}$ (Notes 1, 2)	2/15 t_{CY} − 40	140		ns
t_{AFC2}	Address Float to $\overline{PSEN}$ (Notes 1, 2)	1/30 t_{CY} − 40	10		ns
t_{LAFC1}	ALE to Control ($\overline{RD}$, $\overline{WR}$) (Note 1)	1/5 t_{CY} − 75	200		ns
t_{LAFC2}	ALE to Control ($\overline{PSEN}$) (Note 1)	1/10 t_{CY} − 75	60		ns
t_{CA1}	Control to ALE ($\overline{RD}$, $\overline{WR}$, $\overline{PROG}$) (Note 1)	1/15 t_{CY} − 40	50		ns
t_{CA2}	Control to ALE ($\overline{PSEN}$) (Note 1)	4/15 t_{CY} − 40	320		ns
t_{CP}	Port Control Setup to $\overline{PROG}$ (Note 1)	1/10 t_{CY} −80	50		ns
t_{PC}	Port Control Hold from $\overline{PROG}$ (Note 1)	4/15 t_{CY} − 260	100		ns
t_{PR}	$\overline{PROG}$ to P2 Input Valid (Note 1)	17/30 t_{CY} − 140		630	ns
t_{PF}	Input Data Hold from $\overline{PROG}$ (Notes 1, 4)	1/10 t_{CY}	0	140	ns
t_{DP}	Output Data Setup (Note 1)	2/5 t_{CY} − 290	260		ns
t_{PD}	Output Data Hold (Note 1)	1/10 t_{CY} − 90	40		ns
t_{PP}	$\overline{PROG}$ Pulse Width (Note 1)	7/10 t_{CY} − 250	700		ns
t_{PL}	Port 2 I/O Setup to ALE (Note 1)	4/15 t_{CY} − 200	160		ns
t_{LP}	Port 2 I/O Hold to ALE (Note 1)	1/10 t_{CY} − 120	15		ns
t_{PV}	Port Output From ALE (Note 1)	3/10 t_{CY} + 100		510	ns
t_{CY}	Cycle Time (Note 3)		1.36	15	μs
t_{OPRR}	T_0 Rep Rate	3/15 t_{CY}	270		ns

Note 1: Control outputs $C_L = 80$ pF, Bus outputs $C_L = 150$ pF.

Note 2: Bus High Impedance Load = 20 pF.

Note 3: $t_{CY} = 15/f$ (assumes 50% duty cycle).

Note 4: Maximum spec listed is for user information only to prevent system bus contention.

Note 5: $V_{IH} = 3.8V$, $V_{IL} = 0.45V$.

AC Electrical Characteristics for NS8050U/NS8040U

Symbol	Parameter	$f(t_{CY})$ $t_{CY} = 1.36\ \mu s$	11 MHz		Units
			Min	Typ	
tc ÷ 1	Divide by One SK	t_{CY}	1360	1360	ns
tc ÷ 4	Divide by Four SK	$4\ t_{CY}$	5440	5440	ns
t_{setup}	SI Setup to SK ↑		200	140	ns
t_{hold}	SI Hold from SK ↑		100	50	ns
$t_{valid \div 1}$	SO Valid Setup to Next SK ↑	$\frac{1}{2}\ t_{CY} - 180$	500	650	ns
$t_{valid \div 4}$	SO Valid Setup to Next SK ↑	$\frac{1}{2}\ t_{CY} - 180$	1860	2000	ns

Note: Absolute Maximum Ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under DC Electrical Characteristics. DC Electrical Characteristics are specified in INS8048-Series Data Sheet.

Input and Output for AC Tests

TL/C/5488–3

Note: AC testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0". Output timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0" (except X1, X2, and $\overline{\text{RESET}}$) (Note 5).

The INS8048/49/50-Series

The devices are designed to be efficient controllers. They have extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory is derived from an instruction set comprised predominantly of single bytes. The remaining instructions are two bytes in length. Additional external memory may be added up to a maximum of 4k bytes of program memory and 256 bytes of data memory without paging.

The INS8048L/49L/50L-Series

The XMOS 8048L Family now makes it possible for the designers to significantly reduce their power supply requirements. This will allow new designs where low power operation is a requirement for portability or battery back-up. The 8048 Low Power Family will also mean longer product life and reliability because of the reduction in chip temperature within the package.

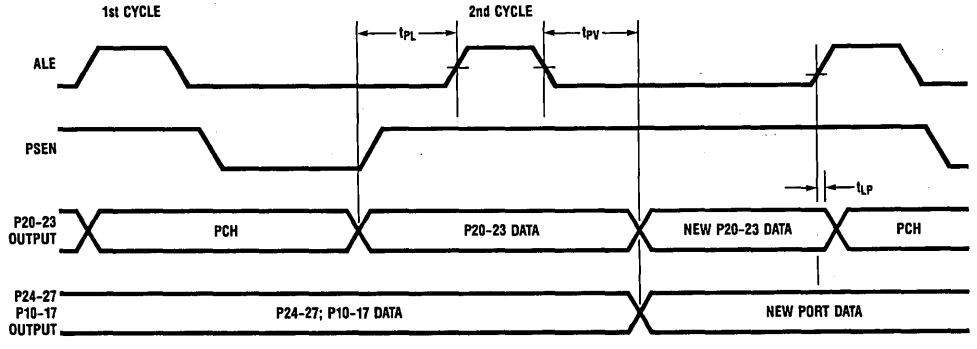
Due to the capabilities of the XMOS process and improvements in circuit design, the 8048L devices consume approximately 25% les than normal N-channel MOS devices available to the marketplace.

The INS8048I/49I/50I-Series

These devices are designed to be efficient controllers in extended temperature range environments (−40°C to +85°C). Similar in every way to the standard devices, the industrial parts are differentiated only by extensive testing and screening to guarantee full functionality in the extended industrial temperature range.

Timing Waveforms

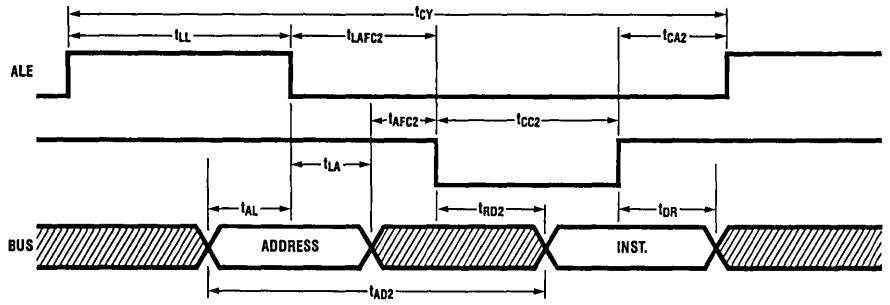
I/O Port Timing



TL/C/5488–4

Timing Waveforms (Continued)

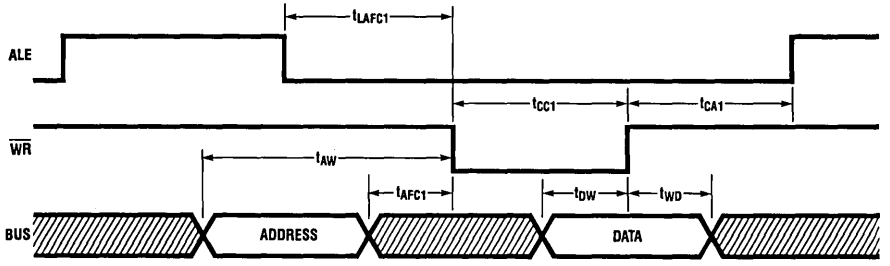
Instruction Fetch from External Program Memory



TL/C/5488–5

Note: Diagonal lines indicate interval of high impedance.

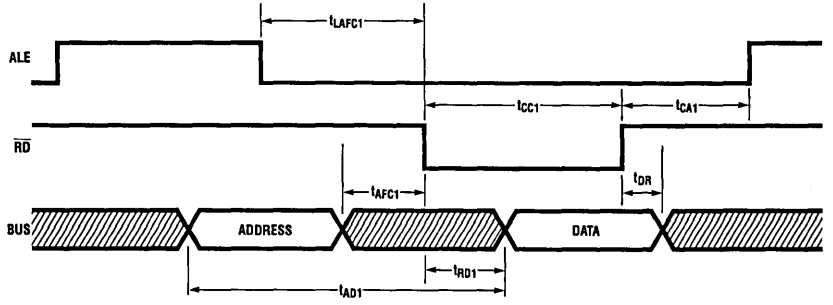
Write to External Data Memory



TL/C/5488–6

Note: Diagonal lines indicate interval of high impedance.

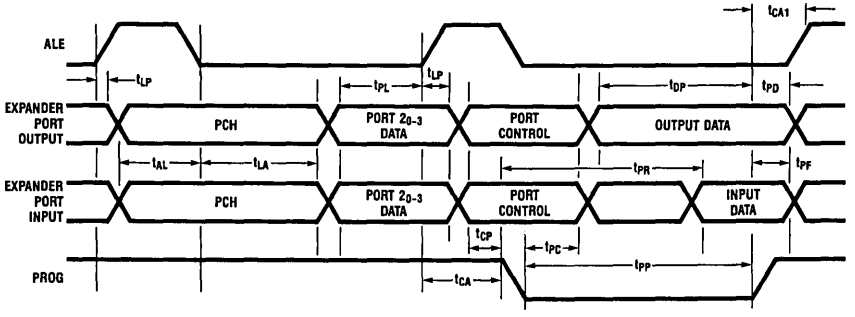
Read from External Data Memory



TL/C/5488–7

Note: Diagonal lines indicate interval of high impedance.

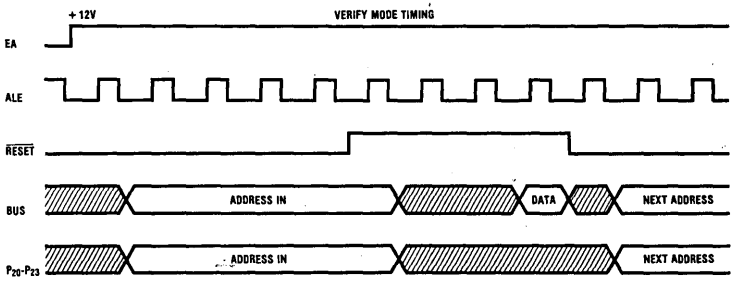
Port 2 Timing



TL/C/5488–8

Timing Waveforms (Continued)

ROM Verify Mode Timing

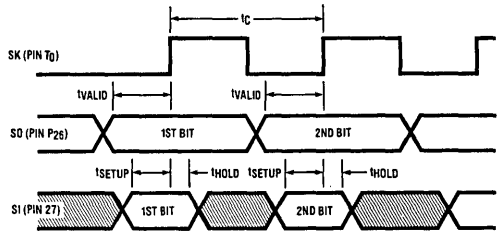


TL/C/5488–9

Recommended Test Methods:

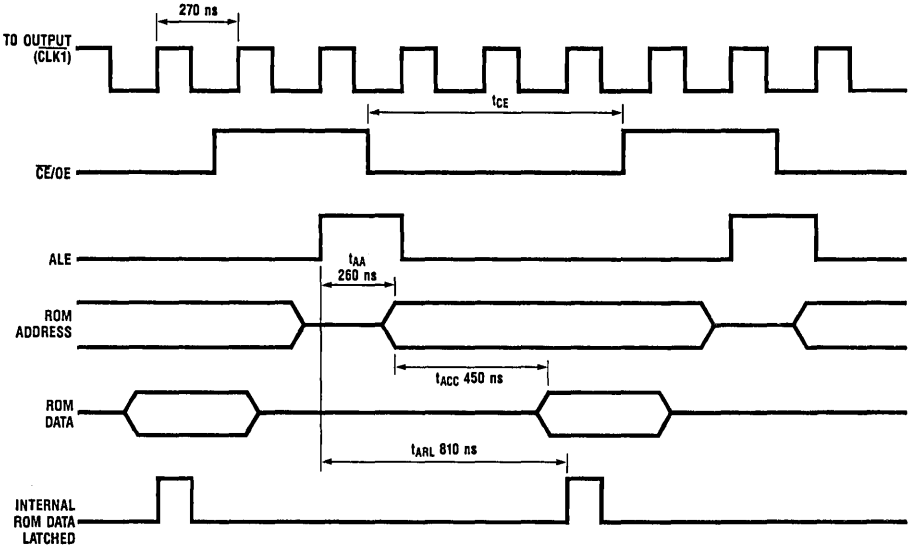
1. $\overline{\text{RESET}}$ must be low before EA goes high.
2. Sync to falling edge of ALE. Falling edge of ALE is machine cycle 4.
3. Force address in with $\overline{\text{RESET}}$ low for 21 machine cycles starting machine cycle 5.
4. Force address in with $\overline{\text{RESET}}$ high for 4 machine cycles.
5. Force $\overline{\text{RESET}}$ high for 12 machine cycles.
6. Force $\overline{\text{RESET}}$ high for 4 machine cycles. Data is valid on bus.
7. Force $\overline{\text{RESET}}$ low for 4 machine cycles.
8. Repeat steps 3 through 7 for other addresses.

NS8050U MICROWIRE/PLUS Timing



TL/C/5488–10

NS87P50U Piggy-Back Microcomputer EPROM Timing



TL/C/5488–11

Frequency = 11 MHz

AC Electrical Characteristics

T_A = 0°C to +70°C, V_{CC} = +5V ±10%, V_{SS} = 0V, V_{DD} = 3.6V to 5.5V, unless otherwise specified.

Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
t_{ACC}	ROM Address Setup to ROM Data (6 MHz Access Time)	6 MHz			850	ns
t_{ACC}	ROM Address Setup to ROM Data (11 MHz Access Time)	11 MHz			450	ns
t_{CE}	Chip Enable and Output Enable Active	11 MHz			870	ns
t_{AA}	ALE to ROM Address Setup	11 MHz			260	ns
t_{ARL}	ALE to ROM Data Latch	11 MHz			810	ns

Other AC Electrical Characteristics same as 48-Series.

DC Electrical Characteristics

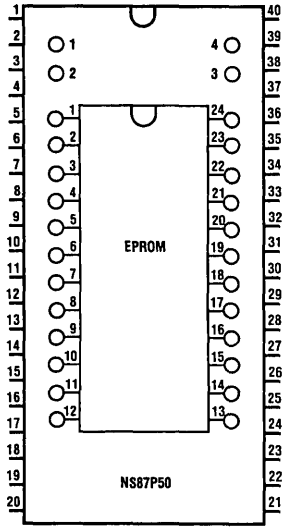
T_A = 0°C to +70°C, V_{CC} = V_{DD} = +5V ±10%, V_{SS} = 0V, unless otherwise specified.

Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
I_{CC1}	Current Available to EPROM	V_{CC} = 5V			150	mA
I_{PP}	V_{PP} Current	V_{PP} = 5V			5	mA
I_{DD}	256 Words on Standby Current				20	mA
I_{DD} + I_{CC}	Total Supply Current (without EPROM)	T_A = 25°C All Outputs Open		60	100	mA
V_{DD}	Standby Power Supply		2.6		V_{CC}	V

Other DC Electrical Characteristics same as 48-Series.

Connection Diagrams

Pin Top View



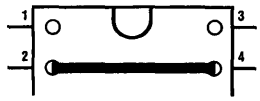
TL/C/5488–12

EPROM Pin Descriptions

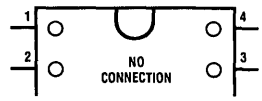
A_0–A_9	Addresses
$\overline{CE}$/PGM	Chip Enable Program
$\overline{OE}$	Output Enable
O_0–O_7	Outputs
A_R	Select Reference Input Level—V_{SS}
V_{PP}	+5V
V_{CC}	+5V
GND	Power Ground
A_R	Ground

INS8048 Emulation

INS8049 Emulation

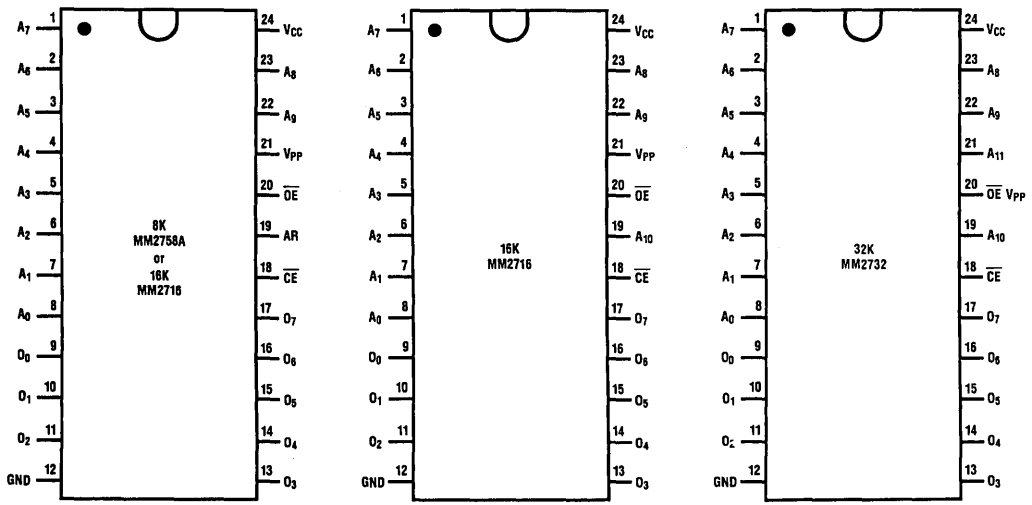


NS8050 Emulation



TL/C/5488–13

EPROM



TL/C/5488–14

Functional Pin Descriptions

INPUT SIGNALS

Reset ($\overline{\text{RESET}}$): An active low (0) input that initializes the processor and is used to verify program memory.

Single Step ($\overline{\text{SS}}$): Active low (0) input which, in conjunction with ALE, can single step the processor through each instruction.

External Access (EA): An active high (1) input that forces all program memory fetches to reference external program memory.

Testable Input 0 (T0): Testable input pin using conditional branch functions JT0 (T0 = 1) or JNT0 (T0 = 0). T0 can be designated as the clock output using instruction ENT0 CLK. For NS8050U, T0 is also used as the SK clock output for MASTER and SK clock input for SLAVE in the MICROWIRE/PLUS mode. Done FLIP FLOP is tested by testing T0.

Testable Input 1 (T1): Testable input pin using conditional branch functions JT1 (T1 = 1) or JNT1 (T1 = 0). T1 can be designated as the Timer/Counter input from an external source using instruction STRT CNT.

Interrupt ($\overline{\text{INT}}$): An active low input that initiates an interrupt when interrupt is enabled. Interrupt is disabled after a RESET. Also can be tested with instruction JNI (INT = 0).

OUTPUT SIGNALS

Read Strobe ($\overline{\text{RD}}$): An active low output strobe activated during a Bus read. Can be used to enable data onto the Bus from an external device. Used as a Read Strobe to External Data Memory.

Write Strobe ($\overline{\text{WR}}$): An active low output strobe activated during a Bus write. Used as a Write Strobe to External Data Memory.

Program Store Enable ($\overline{\text{PSEN}}$): An active low output that occurs only during an external program memory fetch.

Address Latch Enable (ALE): An active high output that occurs once during each cycle and is useful as a clock output. The negative going edge of ALE strobes the address into External Data or Program Memory.

Program ($\overline{\text{PROG}}$): This output (active out) provides the output strobe for INS8243 I/O Expander.

INPUT/OUTPUT SIGNALS

Crystal Input (XTAL1, XTAL2): These two pins connect the crystal for internal oscillator operation. XTAL1 is the timing input for external source.

Port 1 (P10–P17): 8-bit quasi-bidirectional port.

Port 2 (P20–P27): 8-bit quasi-bidirectional port. During an external program memory fetch, the 4 high-order program counter bits occur at P20–P23. They also serve as a 4-bit I/O Expander Bus when the INS8243 I/O Expander is used.

Bus (DB_0–DB_7): True bidirectional port, either statically latched or synchronous. Can be written to using $\overline{\text{WR}}$ Strobe, or Read from using $\overline{\text{RD}}$ Strobe. During an External Program Memory fetch, the 8 lower-order program counter bits are present at this port. The addressed instruction appears on this bus when PSEN is low. During an external RAM data transaction, this port presents address and data under control of ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$.

V_{SS}: Processor Ground potential.

V_{DD}: V_{DD} functions as the Low Power Standby Voltage. Can be tied to V_{CC} if power-down operation is not required.

V_{CC} (Pin 40): Primary power source for 48-Series devices.

NS8050U MICROWIRE Mode Pin Descriptions

Pin 1, Serial Clock (SK): Input or output clocking signal to the MICROWIRE/PLUS serial circuitry.

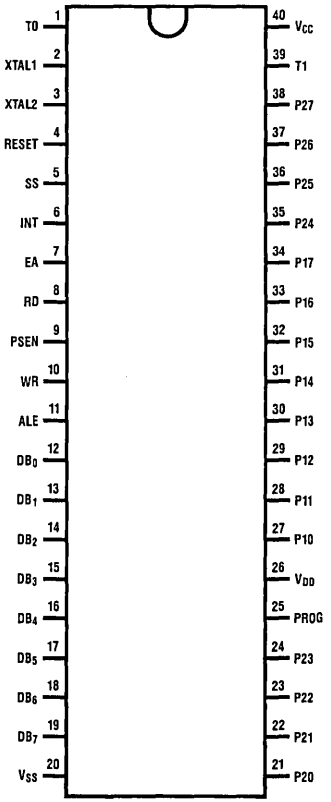
Pin 38, Serial Input (SI): Serial input to the 8-bit MICROWIRE/PLUS shift register.

Pin 37, Serial Output (SO): Serial output from the 8-bit MICROWIRE/PLUS shift register.

Pin 3, XTAL2: High-impedance input to oscillator circuit.

Pin 2, XTAL1: Low-impedance output from oscillator circuit.

Connection Diagram



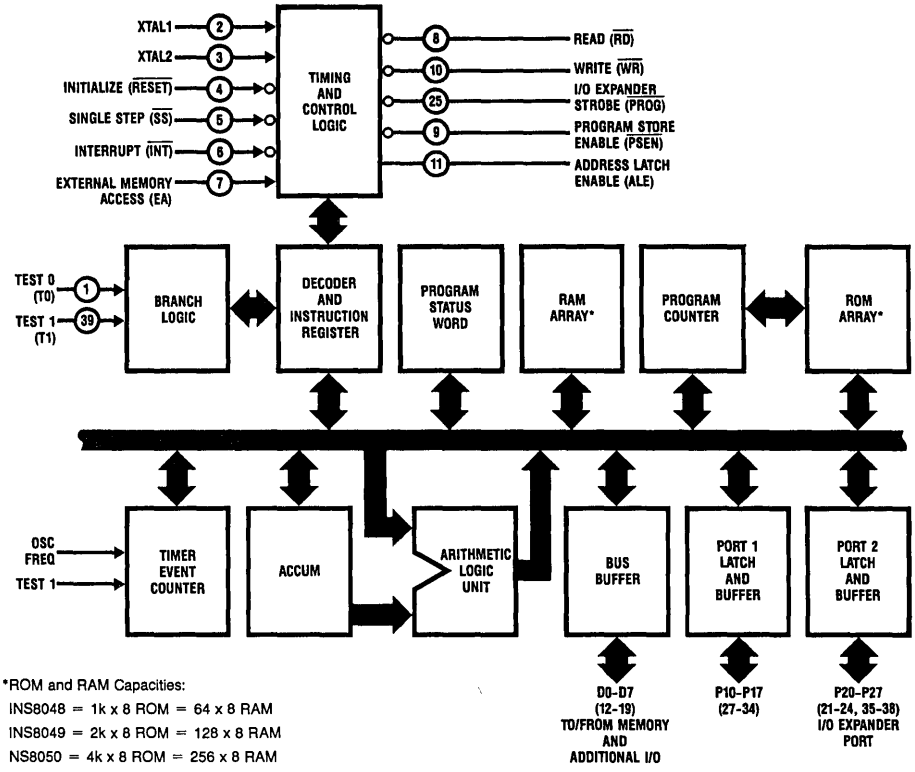
TL/C/5488–15

Order Number INS8048, INS8048J, INS8048L, INS8049, INS8049I, INS8049L, NS8040U, NS8050I, NS8050L, NS8050U, NS87P50, NS87P50I and NS87P50U
See NS Package Number D40G or N40A

Functional Description

The following paragraphs contain the functional description of the major elements of the 48-Series microcomputer/microprocessor. *(Figure 1)* is a block diagram of the 48-Series devices. The data paths are illustrated in simplified form to show how the various logic elements communicate with each other to implement the instruction set common to all devices.

48-Series Block Diagram



*ROM and RAM Capacities:
INS8048 = 1k x 8 ROM = 64 x 8 RAM
INS8049 = 2k x 8 ROM = 128 x 8 RAM
NS8050 = 4k x 8 ROM = 256 x 8 RAM

TL/C/5488–16

FIGURE 1. 48-Series Detailed Block Diagram

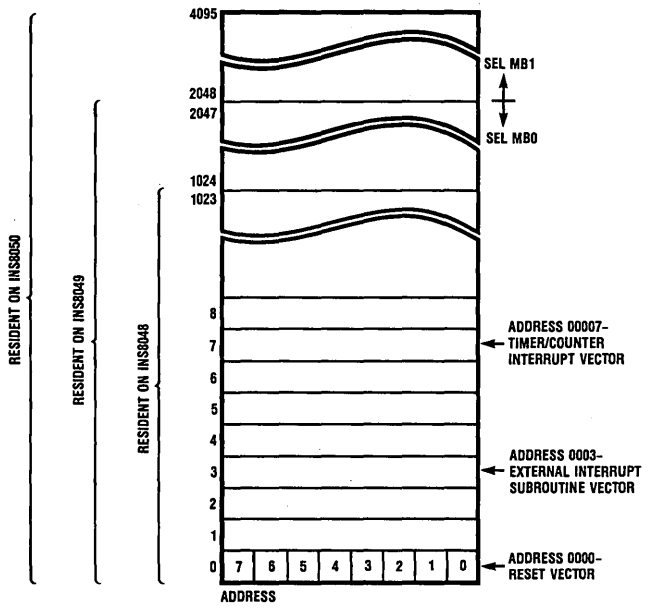
Functional Description (Continued)

PROGRAM MEMORY

The Program Memory (ROM) contained on the INS8048/49/50 devices is comprised of 1024, 2048 or 4096 8-bit bytes, respectively. As is seen by examining the 48-Series instruction set, these bytes may be program instructions, program data or ROM addressing data. The ROM for the above devices must be mask programmed at the National Semiconductor factory. The ROMless microcomputers, NS8035, INS8039 and INS8040, use external program memory. This makes program development straightforward using standard UV erasable PROMs to emulate a possible future single chip (using the on-board ROM) system. ROM addressing, up to a maximum of 4k, is accomplished by a 12-bit Program Counter (PC). The INS8048 and INS8049 will automatically address external memory when the boundary of their internal memories, 1k and 2k respectively, are exceeded. The binary value of the address selects one of the 8-bit bytes contained in ROM. A new address is loaded into the PC register during each instruction cycle. Unless the instruction is a transfer of control instruction, the PC register is loaded with the next sequential binary count value.

With reference to the Program Memory Map (see *Figure 2*) there are three ROM addresses which provide for the control of the microcomputer.

1. Memory Location 0000—Asserting the RESET (negative true) input to the microcomputer forces the first instruction to be fetched from address 0000.
2. Memory Location 0003—Asserting the interrupt (negative true) input to the microcomputer (when interrupt is enabled) forces a jump to subroutine.
3. Memory Location 0007—A Timer/Counter interrupt that results from Timer/Counter overflow (when enabled) forcing a jump to subroutine.



TL/C/5488–17

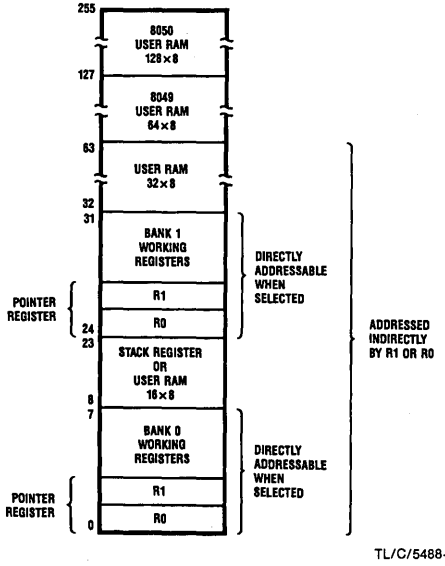
FIGURE 2. INS8048/49/50 Resident ROM Program Memory Map

Functional Decscription (Continued)

DATA MEMORY (RAM)

The resident RAM data memory is arranged as 64 (INS8035/8048), 128 (INS8039/8049), or 256 (NS8040/8050) bytes. RAM addressing is implemented indirectly via either of two 8-bit RAM pointer registers R0 and R1. These pointer registers are essentially the first two locations in the RAM (see *Figure 3*), addressing 00 and 01. RAM addressing may also be performed directly by 11 direct register instructions. The register area of the RAM array is made up of eight working registers that occupy either the first bank (0), locations 0 to 7, or the second bank (1), locations 24 to 31. The second bank of working registers is selected by using the Register Bank Switch instruction (SEL RB). If this bank is not used for working registers, it can be used as user RAM.

There is an 8-level stack after Bank 0 that occupies address locations 8 to 23. These RAM locations are addressed indirectly through R0, R1 or the 3-bit Stack Pointer (SP). The stack pointer keeps track of the return address and pushes each return address down into the stack. There are 8 levels of subroutine nesting possible in the stack because each address occupies 10 bits or more using two bytes in RAM. When the level of subroutine nesting is less than 8, the stacks not used may be utilized as user RAM locations.



TL/C/5488–18

FIGURE 3. 8048-Series Resident RAM Data Memory Map

INPUT/OUTPUT

The 48-series devices have 27 lines of input/output organized as three 8-bit ports plus three test inputs. The three ports may be used as inputs, outputs or bidirectional ports. Ports 1 and 2 differ from Port 3 (Bus Port) in that they are quasi-bidirectional ports. Ports 1 and 2 can be used as input and output while being statically latched. If more I/O lines are required, Port 2 can also serve as a 4-bit I/O bus expander when used in conjunction with the INS8243 I/O Expander.

The bus port is a true bidirectional port and is either statically latched or synchronous. It can be written to using $\overline{WR}$ strobe or read from using $\overline{RD}$ strobe. During an external program memory fetch, the 8 lower-order program counter bits are present at this port. The addressed instruction appears on this bus when $\overline{PSEN}$ is low. During an external RAM data transaction, this port presents address and data under control of ALE, $\overline{RD}$, and $\overline{WR}$.

POWER-DOWN MODE

During the power-down mode, V_{DD}, which normally maintains the RAM cells, is the only pin that receives power. V_{CC}, which serves the CPU and ports, is dropped from a voltage of nominal 5 to 0 after the CPU is reset, so that the RAM cells are unaltered by the loss of power. When power is restored, the processor goes through the normal power-on procedure.

MICROWIRE FUNCTIONS

New functions applicable to microwire mode are as follows:

The SK output frequency is selected by writing into the P27 latch as follows:

0 = Instruction Cycle Divided by 1

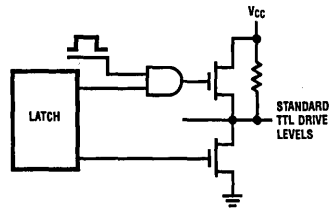
1 = Instruction Cycle Divided by 4

The SO output is enabled or disabled by writing into the P26 latch as follows:

0 = Enabled

1 = Disabled

The DONE flip-flop is tested with the JT0, JNT0 instructions.



TL/C/5488–19

FIGURE 4. Ports 1 and 2 Input/Output Circuit

Instruction Set

Table I details the 96 instructions common to both the microcomputers and the microprocessors. The table provides the mnemonic, function, and description, instruction code, number of cycles and, where applicable, flag settings.

TABLE I. Instruction Set

Mnemonic	Function	Description	Cycles	Bytes	Flags			
					C	AC	F0	F1
CONTROL								
EN I		Enable the External Interrupt Input.	1	1				
DIS I		Disable the External Interrupt Input.	1	1				
ENT0 CLK		Enable T0 as the Clock Output.	1	1				
SEL MB0	(DBF) ← 0	Select Bank 0 (locations 0–2047) of Program Memory.	1	1				
SEL MB1	(DBF) ← 1	Select Bank 1 (locations 2048–4095) of Program Memory.	1	1				
SEL RB0	(BS) ← 0	Select Bank 0 (locations 0–7) of Data Memory.	1	1				
SEL RB1	(BS) ← 1	Select Bank 1 (locations 24–31) of Data Memory.	1	1				
DATA MOVES								
MOV A, #data	(A) ← data	Move Immediate the specified data into the Accumulator.	2	2				
MOV A, Rr	(A) ← (Rr); r = 0–7	Move the contents of the designated register into the Accumulator.	1	1				
MOV A, @ Rr	(A) ← ((Rr)); r = 0–1	Move Indirect the contents of data memory location into the Accumulator.	1	1				
MOV A, PSW	(A) ← (PSW)	Move contents of the Program Status Word into the Accumulator.	1	1				
MOV Rr, #data	(Rr) ← data; r = 0–7	Move Immediate the specified data into the designated register.	2	2				
MOV Rr, A	(Rr) ← (A); r = 0–7	Move Accumulator contents into the designated register.	1	1				
MOV @ Rr, A	((Rr)) ← (A); r = 0–1	Move Indirect Accumulator contents into data memory location.	1	1				
MOV @ Rr, #data	((Rr)) ← data; r = 0–7	Move Immediate the specified data into data memory.	2	2				
MOV PSW, A	(PSW) ← (A)	Move contents of Accumulator into the Program Status Word.	1	1	•	•	•	
MOVP A, @ A	(PC 0–7) ← (A) (A) ← ((PC))	Move the content of program memory location in the current page addressed by the content of Accumulator into the Accumulator.	2	1				
MOVP3 A, @ A	(PC 0–7) ← (A) (PC 8–10) ← 011 (A) ← ((PC))	Move the content of program memory location in page 3 addressed by the content of Accumulator into the Accumulator.	2	1				
MOVX A, @ R	(A) ← ((Rr)); r = 0–1	Move Indirect the contents of external data memory into the Accumulator.	2	1				
MOVX @ R, A	((Rr)) ← (A); r = 0–1	Move Indirect the contents of the Accumulator into external data memory.	2	1				
XCH A, Rr	(A) ⟷ (Rr); r = 0–7	Exchange the Accumulator and designated register's contents.	1	1				
XCH A, @ Rr	(A) ⟷ ((Rr)); r = 0–1	Exchange Indirect contents of Accumulator and location in data memory.	1	1				
XCHD A, @ Rr	(A0–A3) ← (((Rr)) 0–3); R = 0–1	Exchange Indirect 4-bit contents of Accumulator and data memory.	1	1				

TABLE I. Instruction Set (Continued)

Mnemonic	Function	Description	Cycles	Bytes	Flags			
					C	AC	F0	F1
TIMER COUNTER								
EN TCNTI		Enable Internal Interrupt Flag for Timer/ Counter output.	1	1				
DIS TCNTI		Disable Internal Interrupt Flag for Timer/ Counter output.	1	1				
MOV A, T	(A) ← (T)	Move contents of Timer/Counter into Accumulator.	1	1				
MOV T, A	(T) ← (A)	Move contents of Accumulator into Timer/ Counter.	1	1				
STOP TCNT		Stop Count for Event Counter.	1	1				
STRT CNT		Start Count for Event Counter.	1	1				
STRT T		Start Count for Timer.	1	1				
ACCUMULATOR								
ADD A, #data	(A) ← (A) + data	Add Immediate the specified Data to the Accumulator.	2	2	•	•		
ADD A, Rr	(A) ← (A) + (Rr) for r = 0–7	Add contents of designated register to the Accumulator.	1	1	•	•		
ADD A, @ Rr	(A) ← (A) + ((Rr)) for r = 0–1	Add Indirect the contents of data memory location of the Accumulator.	1	1	•	•		
ADDC A, #data	(A) ← (A)(C) + data	Add Immediate with carry the specified data to the Accumulator.	2	2	•	•		
ADDC A, Rr	(A) ← (A) + (C) + (Rr) for r = 0–7	Add with carry the contents of the designated register to the Accumulator.	1	1	•	•		
ADDC A, @ Rr	(A) ← (A) + (C) + ((Rr)) for r = 0–1	Add Indirect with carry the contents of data memory location to the Accumulator.	1	1	•	•		
ANL A, #data	(A) ← (A) AND data	Logical AND specified Immediate Data with Accumulator	2	2				
ANL A, Rr	(A) ← (A) AND (Rr) for r = 0–7	Logical AND contents of designated register with Accumulator	1	1				
ANL A, @ Rr	(A) ← (A) AND ((Rr)) for r = 0–1	Logical AND Indirect the contents of data memory with Accumulator.	1	1				
CPL A	(A) ← NOT (A)	Complement the contents of the Accumulator.	1	1				
CLR A	A ← 0	CLEAR the contents of the Accumulator.	1	1				
DA A		DECIMAL ADJUST the contents of the Accumulator.	1	1	•			
DEC A	(A) ← (A) − 1	DECREMENT by 1 the Accumulator's contents.	1	1				
INC A	(A) ← (A) + 1	Increment by 1 the Accumulator's contents.	1	1				
ORL A, #data	(A) ← (A) OR data	Logical OR specified immediate data with Accumulator.	2	2				
ORL A, Rr	(A) ← (A) OR (Rr) for r = 0–7	Logical OR contents of designated register with Accumulator.	1	1				
ORL A, @ Rr	(A) ← (A) OR ((Rr)) for r = 0–1	Logical OR Indirect the contents of data memory location with Accumulator.	1	1				

TABLE I. Instruction Set (Continued)

Mnemonic	Function	Description	Cycles	Bytes	Flags			
					C	AC	F0	F1
CCUMULATOR (Continued)								
LA	(An + 1) ← (An) for n = 0–6 (A0) ← (A7)	Rotate Accumulator left by 1-bit without carry.	1	1				
LC A	(An + 1) ← (An); n = 0–6 (A0) ← (C) (C) ← (A7)	Rotate Accumulator left by 1-bit through carry.	1	1	•			
R A	(An) ← (An + 1); n = 0–6 (A7) ← (A0)	Rotate Accumulator right by 1-bit without carry.	1	1				
RC A	(An) ← (An + 1); n + 0–6 (A7) ← (C) (C) ← (A0)	Rotate Accumulator right by 1-bit through carry.	1	1	•			
WAP	(A4–A7) ⟷ (A0–A3)	Swap the 2 4-bit nibbles in the Accumulator.	1	1				
RL A, #data	(A) ← (A) XOR data	Logical XOR Immediate specified data with Accumulator.	2	2				
RL A, Rr	(A) ← (A) XOR (Rr) for r = 0–7	Logical XOR contents of designated register with Accumulator.	1	1				
RL A, @ Rr	(A) ← (A) XOR ((Rr)) for r = 0–1	Logical XOR Indirect the contents of data memory location with Accumulator.	1	1				
RANCH								
JNZ Rr, addr	(Rr) ← (Rr) −; r = 0–7 if (Rr) ≠ 0; (PC 0–7) ← addr	Decrement the specified register and test contents.	2	2				
Bb addr	(PC 0–7) ← addr if Bb = 1 (PC) ← (PC) + 2 If BB = 0	Jump to specified address if Accumulator bit is set.	2	2				
C addr	(PC 0–7) ← addr if C = 1 (PC) ← (PC) + 2 If C = 0	Jump to specified address if carry flag is set.	2	2				
FO addr	(PC 0–7) ← addr if F0 = 1 (PC) ← (PC) + 2 If F0 = 0	Jump to specified address if Flag F0 is set.	2	2				
FI addr	(PC 0–7) ← addr if F1 = 1 (PC) ← (PC) + 2 If F1 = 0	Jump to specified address if Flag F1 is set.	2	2				
IMP addr	(PC 8–10) ← 8–10 (PC 0–7) ← addr 0–7 (PC 11) ← DBF	Direct Jump to specified address with the 2k address block.	2	2				
IMPP @ A	(PC 0–7) ← ((A))	Jump Indirect to specified address pointed to by the Accumulator in current page.	2	1				
JNC addr	(PC 0–7) ← addr if C = 0 (PC) ← (PC) + 2 if C = 1	Jump to specified address if carry flag is low.	2	2				
JNI addr	(PC 0–7) ← addr if I = 0 (PC) ← (PC) + 2 if I = 1	Jump to specified address if interrupt is low.	2	2				
JNT0 addr	(PC 0–7) ← addr if T0 = 0 (PC) ← (PC) + 2 if T0 = 1	Jump to specified address if Test 0 is low.	2	2				
JNT1 addr	(PC 0–7) ← addr if T1 = 0 (PC) ← (PC) + 2 if T1 = 1	Jump to specified address if Test 1 is low.	2	2				

TABLE I. Instruction Set (Continued)

Mnemonic	Function	Description	Cycles	Bytes	Flags			
					C	AC	F0	I
BRANCH (Continued)								
JNZ addr	(PC 0–7) ← addr if A ≠ 0 (PC) ← (PC) + 2 if A = 0	Jump to specified address if Accumulator is non-zero.	2	2				
JFT addr	(PC 0–7) ← addr if TF = 1 (PC) ← (PC) + 2 if TF = 0	Jump to specified address if Timer Flag is set to 1.	2	2				
JT0 addr	(PC 0–7) ← addr if T0 = 1 (PC) ← (PC) + 2 if T0 = 0	Jump to specified address if Test 0 is a 1.	2	2				
JT1 addr	(PC 0–7) ← if T1 = 1 (PC) ← (PC) + 2 if T1 = 0	Jump to specified address if Test 1 is a 1.	2	2				
JZ addr	(PC 0–7) ← addr if A = 0 (PC) ← (PC) + 1 if A = 1	Jump to specified address if Accumulator is 0.	2	2				
INPUT/OUTPUT								
ANL BUS, #data	(BUS) ← (BUS) AND data	Logical AND Immediate specified data with contents of BUS.	2	2				
ANL Pp, #data	(Pp) ← (Pp) AND data; p = 1–2	Logical AND Immediate specified data with designated port (1 or 2).	2	2				
ANLD Pp, A	(Pp) ← (Pp) AND (A0–A3); p = 4–7	Logical AND contents of Accumulator with designated port (4–7).	2	1				
IN A, Pp	(A) ← (Pp); p = 1–2	Input data from designated port (1–2) into Accumulator.	2	1				
INS A, BUS	(A) ← (BUS)	Input strobed BUS data into Accumulator.	2	1				
MOVD A, Pp	(A0–A3) ← (Pp); p = 4–7 (A4–A7) ← 0	Move contents of designated port (4–7) into Accumulator.	2	1				
MOVD Pp, A	(Pp) ← (A0–A3); p = 4–7	Move contents of Accumulator to designated port (4–7).	2	1				
ORL BUS, #data	(BUS) ← (BUS) OR data	Logical OR Immediate specified data with contents of BUS.	2	2				
ORLD Pp, A	(Pp) ← (Pp) OR (A0–A3); p = 4–7	Logical OR contents of Accumulator with designated port (4–7).	2	1				
ORL Pp, #data	(Pp) ← (Pp) OR data; p = 1–2	Logical OR Immediate specified data with designated port (1–2).	2	2				
OUTL BUS, A	(BUS) ← (A)	Output contents of Accumulator onto BUS.	2	1				
OUTL Pp, A	(Pp) ← (A); p = 1–2	Output contents of Accumulator to designated port (1–2).	2	1				
REGISTERS								
DEC Rr	(Rr) ← (Rr) − 1; r = 0–7	Decrement by 1 contents of designated register.	1	1				
INC Rr	(Rr) ← (Rr) + 1; r = 0–7	Increment by 1 contents of designated register.	1	1				
INC @ Rr	((Rr)) ← ((Rr)) + 1; r = 0–1	Increment Indirect by 1 the contents of data memory location.	1	1				

TABLE I. Instruction Set (Continued)

Mnemonic	Function	Description	Cycles	Bytes	Flags			
					C	AC	F0	F1
SUBROUTINE								
CALL addr	((SP)) ← (PC) ((SP)) ← (PSW 4–7) (SP) ← (SP) + 1 (PC 8–10) ← addr 8–10 (PC 0–7) ← addr 0–7 (PC 11) ← DBF	Call designated Subroutine.	2	2				
RET	(SP) ← (SP) − 1 (PC) ← ((SP))	Return from Subroutine without restoring Program Status Word.	2	1				
RETR	(SP) ← (SP) − 1 (PC) ← ((SP)) (PSW 4–7) ← ((SP))	Return from Subroutine restoring Program Status Word.	2	1	•	•		
FLAGS								
CPL C	(C) ← NOT (C)	Complement Content of carry bit.	1	1	•			
CPL F0	(F0) ← NOT (F0)	Complement Content of Flag F0.	1	1			•	
CPL F1	(F1) ← NOT (F1)	Complement Content of Flag F1.	1	1				•
CLR C	(C) ← 0	Clear content of carry bit to 0.	1	1	•			
CLR F0	(F0) ← 0	Clear content of Flag 0 to 0.	1	1			•	
CLR F1	(F1) ← 0	Clear content of Flag 1 to 0.	1	1				•
MISCELLANEOUS								
NOP		No operation	1	1				
MICROWIRE INSTRUCTIONS								
XCHM	A ⟷ SIO	Reset DONE flip-flop, Clock counter. SK designated as an output. MICROWIRE/PLUS mode selected.						
XCHS	A ⟷ SIO	Reset DONE flip-flop, Clock counter. SK designated as an input. MICROWIRE/PLUS mode selected.						

Symbol Definitions

Symbol	Description
A	Accumulator
AC	Auxiliary Carry Flag
addr	Program Memory Address (12 bits)
b	Bit Designator (b = 0–7
BS	Bank Switch
Bus	Bus Port
C	Carry Flag
CLK	Clock Signal
CNT	Event Counter
D	Nibble Designator (4 bits)
data	Number of Expression (8 bits)
DBF	Memory Bank Flip-Flop
F_0, F_1	Flags 0, 1
I	Interrupt
P	"In-Page" Operation Designator

Symbol	Description
p	Port Designator (p = 1, 2 or 4–7)
PSW	Program Status Word
r	Register Designator (r = 0, 1 or 0–7)
SP	Stack Pointer
T	Timer
TF	Timer Flag
T0, T1	Testable Flags 0, 1
X	External RAM
#	Prefix for Immediate Data
@	Prefix for Indirect Address
$	Program Counter's Current Value
(x)	Contents of Register
((xx))	Contents of Memory Location Addressed by the Contents of Register
	Replaced by

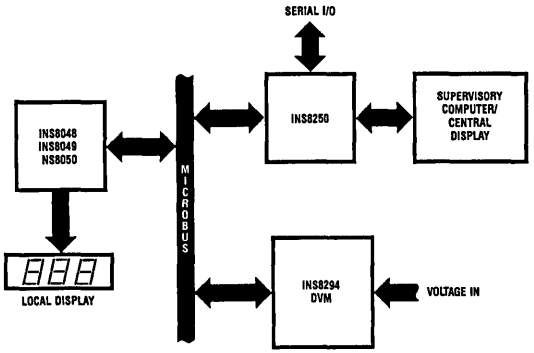
Typical Applications

Figure 5 shows a typical remote data acquisition system with an INS8250 Programmable Asynchronous Communication System, which can receive commands or update information from a supervisory computer. The figure also shows an INS8294 CMOS DVM that receives data at V_{IN} and displays the data on the 7-segment local display unit. Data is transferred from the INS8294 to the INS8094 via National's MICROBUS™.

Figure 6 offers an example system that utilizes the unique capacity of these parts. With the addition of an ADC0837 serial A/D converter and a DM74LS138 3-to-8 decoder, the basic INS8048 and MM5445/MM5446/MM5447/MM5448, MM5450/MM5451/MM5452/MM5453 system can display inputs from 8 separate I/O devices using only one of the INS8048's I/O ports. The other port and the bus can handle any number of tasks. Shown here is the use of Port 2 to control a keyboard, used to enter limits on the inputs. The bus is used to control relay drivers which will regulate the I/O devices when the limits (previously set on the keyboard) are met.

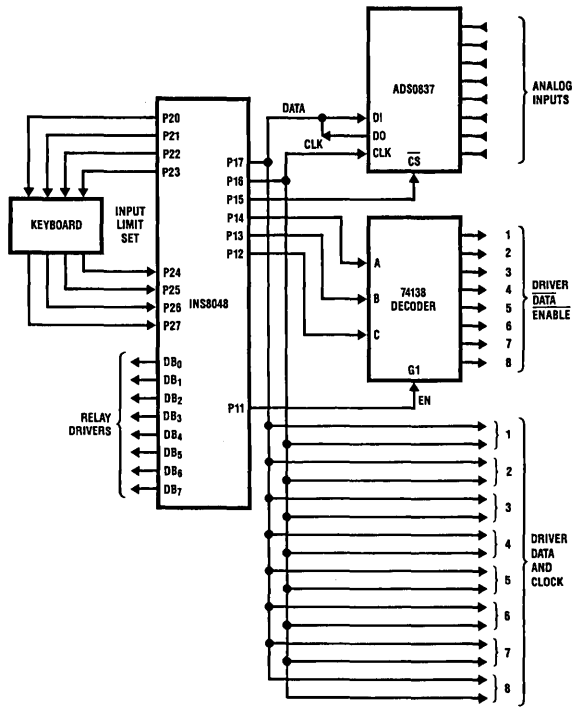
Figure 7 shows a typical way to use the 48-Series microcontrollers in a stand-alone system.

Crystal used is parallel resonant, AT cut and 1 MHz to 6 MHz. All outputs are standard TTL compatible at 5V.



TL/C/5488–20

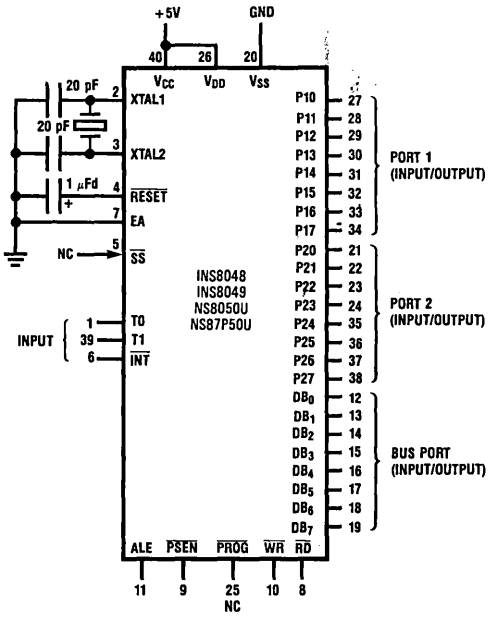
FIGURE 5. Data Acquisition System



TL/C/5488–21

FIGURE 6. Typical Application

Typical Applications (Continued)

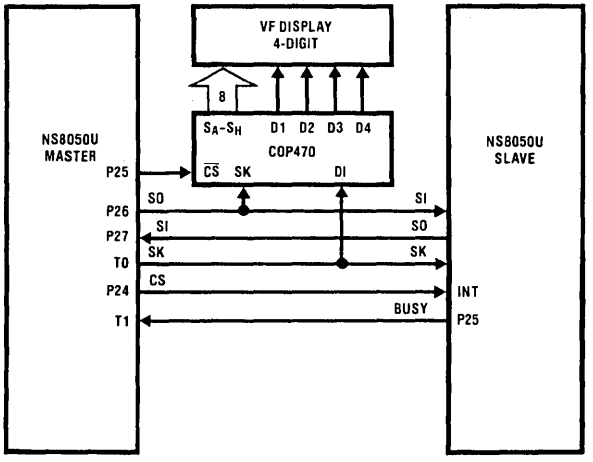


TL/C/5488-22

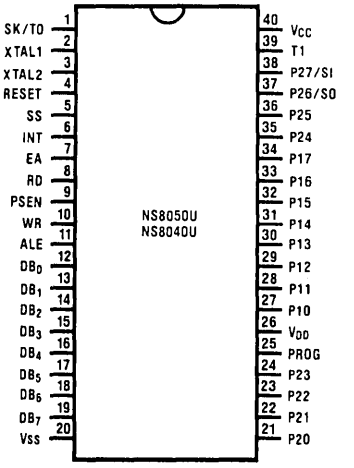
FIGURE 7. Stand-Alone 48-Series

NS8050U Applications

Block and Connection Diagrams



TL/C/5488-23



TL/C/5488-24

Top View

NS8050U Applications (Continued)

MICROWIRE/PLUS consists of a three wire communications port with a clocked 8-bit shift register. The three lines consist of Serial Output (SO), Serial Input (SI) and Serial Clock (SK) and the Shift register is referred to as the Serial Input/Output Register. One 8050U is designated the master and it supplies the clock for the MICROWIRE/PLUS system and initiates data transfer. All transfers are between the master and one or more slaves. A slave can be any MICROWIRE peripheral or another 8050U with MICROWIRE/PLUS.

MICROWIRE Peripherals

DATA ACQUISITION

COP43X	8-bit A/D converters

FREQUENCY GENERATORS AND COUNTERS

COP452	Frequency generator and counter

DISPLAY DRIVERS

COP470	Four-digit vacuum fluorescent display driver
COP472	Triplex liquid crystal display controller
MM5445, 5446, 5447, 5448	Vacuum fluorescent display driver
MM5450, 5451	LED display driver
MM5452, 5453	Liquid crystal display driver
MM5480, 5481	LED display driver
MM5484, 5485	16- and 11-segment LED display driver
MM58201	Multiplexed LCD driver
MM58248	High-voltage display driver

MEMORIES

COP498	Low-power CMOS RAM and timer
COP499	Low-power CMOS RAM
COP494	256-bit, 5V only EEPROM
COP495	1024-bit, 5V only EEPROM

RADIO TUNING

DS890X	AM/FM digital phase-locked loop synthesizer

Peripheral device information may be found in the COPS Microcontrollers Databook.

National Semiconductor Corporation

PRELIMINARY

INS8243 Input/Output Expander

General Description

The INS8243 is an input/output device specifically designed to furnish input/output expansion capabilities for the INS8048, INS8049 and NS8050 single chip microcomputer family. The INS8243 is fabricated using XMOS™ (high density N-channel silicon gate) technology, operates from a single 5V supply and is TTL compatible. It is housed in a 24-pin, dual-in-line package and provides high drive current capabilities at low cost.

The INS8243 expander consists of five, 4-bit bidirectional ports. One port provides the interface with the INS8048/49/50 microcomputer. The remaining four ports provide the input/output expansion.

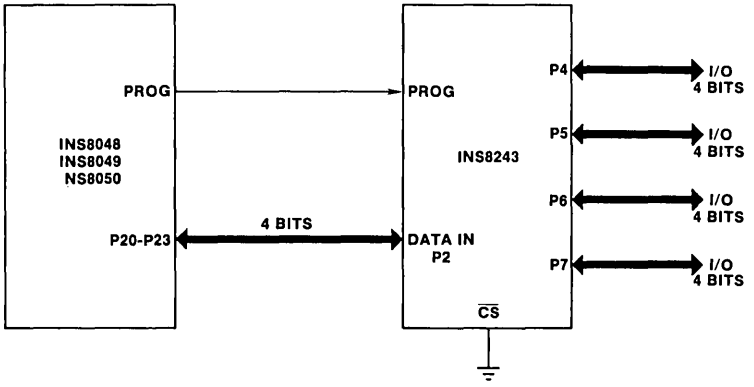
The INS8243 I/O ports function as a direct extension for the resident I/O port of the INS8048/49/50 microcomputer series and are accessed by the MOV, ANL or ORL instructions of the INS8048/49/50.

Features

- XMOS technology
- Single 5V supply
- Low cost I/O expansion
- Easy interface with INS8048/49/50 microcomputers
- High fanout capability
- 24-Pin DIP
- Direct extension of INS8048/49/50 I/O ports

Basic System Configuration

Single Expander Interface



TL/C/7998-1

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Ambient Temperature Under Bias	0°C to +70°C
Storage Temperature Range	−65°C to +150°C
Voltage on Any Pin with Respect to GND	−0.5V to +7.0V

Note: *Absolute maximum ratings indicate limits beyon which permanent damage may occur. Continuous operatio at these limits is not intended; operation should be limited t those conditions specified under DC Electrical Characteris tics.*

DC Electrical Characteristics T_A = 0°C to +70°C, V_{CC} = +5V ±10%, GND = 0V

Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
V_{IL}	Input Low Voltage		−0.5		0.8	V
V_{IH}	Input High Voltage		2.0		V_{CC}+0.5	V
V_{OL1}	Output Low Voltage Ports 4–7	I_{OL} = 5 mA			0.45	V
V_{OL2}	Output Low Voltage Port 7	I_{OL} = 20 mA			1	V
V_{OH1}	Output High Voltage Ports 4–7	I_{OH} = 240 μA	2.4			V
I_{IL1}	Input Leakage Ports 4–7	V_{IN} = V_{CC} to 0V	−10		20	μA
I_{IL2}	Input Leakage Port 2, $\overline{CS}$, PROG	V_{IN} = V_{CC} to 0V	−10		10	μA
V_{OL3}	Output Low Voltage Port 2	I_{OL} = 0.6 mA			0.45	V
I_{CC}	V_{CC} Supply Current			10	20	mA
V_{OH2}	Output Voltage Port 2	I_{OH} = 100 μA	2.4			V
I_{OL}	Sum of all I_{OL} from 16 Output Ports	At 5 mA per pin			100	mA

AC Electrical Characteristics T_A = 0°C to +70°C, V_{CC} = +5V ±10%, GND = 0V

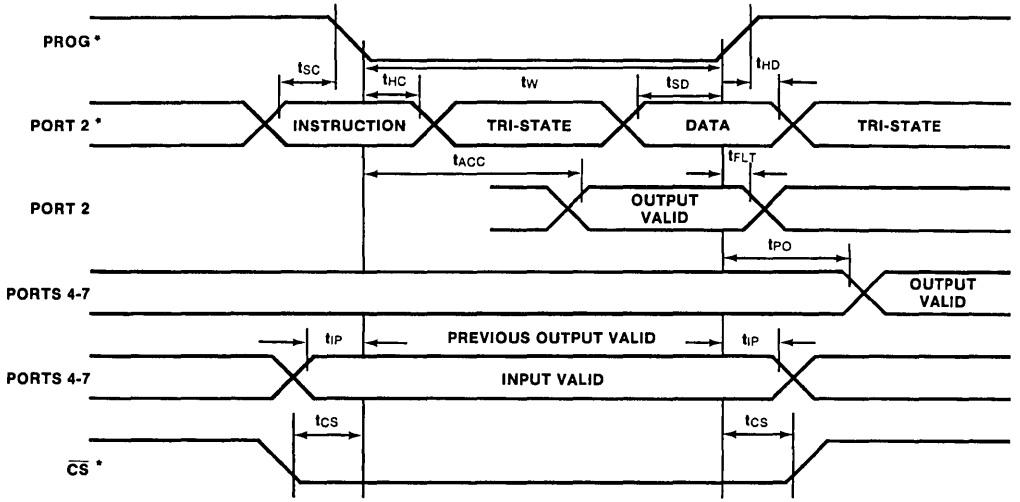
Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
t_{SC}	Code Valid Before PROG	80 pF Load	100			ns
t_{HC}	Code Valid After PROG	20 pF Load	60			ns
t_{SD}	Data Valid Before PROG	80 pF Load	200			ns
t_{HD}	Data Valid After PROG	20 pF Load	20			ns
t_{FLT}	TRI-STATE After PROG	20 pF Load	0		150	ns
t_W	PROG Negative Pulse Width		700			ns
t_{CS}	$\overline{CS}$ Valid Before/After PROG		50			ns
t_{PO}	Ports 4–7 Valid After PROG	100 pF Load			700	ns
t_{LP1}	Ports 4–7 Valid Before/After PROG		100			ns
t_{ACC}	Port 2 Valid After PROG	80 pF Load			650	ns

Timing Waveforms (*These signals are generated by the INS8048/49/50.)

Expander Input Timing

TL/C/7998-2

Input/Output Waveforms



TL/C/7998-3

Functional Pin Descriptions

The following describes the function of the INS8243 input/output pins. Some of these descriptions reference internal circuits.

CONTROL SIGNALS

Chip Select ($\overline{CS}$): When $\overline{CS}$ is low (negative true), the 4-bit input to port 2 is enabled. A high on $\overline{CS}$ inhibits any input to port 2 and no change to internal status and output can occur.

Strobe Input (PROG): The low to high transition on the PROG indicates data is available at port 2; a high to low transition on PROG signifies command and address information is at port 2.

INPUT/OUTPUT SIGNALS

Port 2: Port 2 is a 4-bit bidirectional port that provides the interface between the INS8048/49/50 and the input/output ports 4–7. Communication between the INS8048/49/50 and the INS8243 is accomplished with 4-bit nibbles. A 4-bit nibble consists of two bits that comprise a functional command and two bits that indicate the address of a specific input/output port (P4–P7), or it consists of four data bits. The high to low transition of the PROG input indicates address and command bits are present at port P2. The low to high transition of the PROG input indicates that data bits are present at port P2. If the operation is a read function, data from a selected port (P4–P7) is read to the microcomputer via port 2, prior to the low to high transition of the PROG input. The tables below show the binary inputs for port and command selection.

Port Selection Table

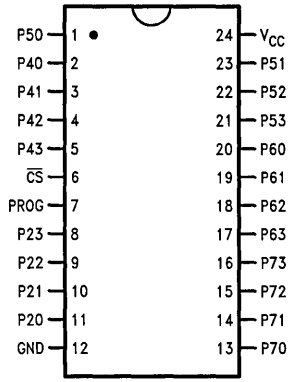
Input/Output Address Code		Selected Input/Output Port
P21	P20	
0	0	4
0	1	5
1	0	6
1	1	7

Functional Command Selection Table

Command Code		Function
P23	P22	
0	0	Read
0	1	Write
1	0	ORLD
1	1	ANLD

Ports 4–7 (PI): These are four, 4-bit bidirectional input/output ports. Each port is addressable and may be programmed to perform a read (input data) or write (output data) via a low impedance latched output. Data presented to port 2 during a write operation may be output directly to the addressed output port or logically ANDed or ORed with data existing in the selected port. For a read operation, data at the addressed port (P4–P7) is transferred to port 2.

Connection Diagram

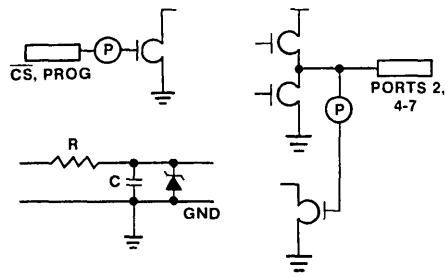


TL/C/7998–4

Order Number INS8243J
See NS Package J24A
Order Number INS8243N
See NS Package N24A

PROTECTION CIRCUITS

Figure 1 illustrates the protection circuits for the input and output pins.



P = Protection Device
τ = 10 ns = RC

TL/C/7998–5

FIGURE 1. Protection Circuits for I/O Pins

Functional Description

The INS8243 *(Figure 2)* consists of four, 4-bit I/O ports that function as the extension of the INS8048/49/50 on-chip input/output port. The four I/O ports (P4 through P7) are addressable and the following programmed INS8048/49/50 instructions are used to access these ports. The instructions move data to/from the INS8048/49/50 accumulator via the INS8243. Timing for the transfer of data is provided by the INS8048/49/50 PROG output.

a. MOVD Pi,A—Shift accumulator data to the addressed port.

b. MOVD A,PI—Shift addressed port data to the accumulator.

c. ANLD Pi,A—ANDing accumulator data to addressed port.

d. ORLD Pi,A—ORing accumulator data to addressed port.

Port 2 of the INS8243 provides the communication interface between the expander and the INS8048/49/50 processor. Each communications exchange is comprised of two, 4-bit nibbles, one nibble consisting of command and address information, the second is a 4-bit data nibble. Timing for the I/O expander is provided by the processor on the PROG input pin.

POWER-ON INITIALIZING MODE

Application of DC power to the chip forces I/O port 2 to the input mode and I/O ports 4 through 7 to the TRI-STATE® output mode (high impedance state). The power-on sequence is initiated when V_{CC} falls below 1V. The input level on the PROG may be high or low when DC is first applied. The initial high to low transition of the PROG input forces the chip to exit the power-on mode.

READ MODE

The INS8243 I/O expander has one read mode that is initialized by the following instruction:

- MOVDA,Pi—instruction from INS8048/49/50 takes data from the addressed I/O port (P4–P7) and moves the data into the accumulator.

The command code and port address code are latched from the input port 2 on the high to low transition of the PROG input pin. When the read command and the port address are decoded, the data from the addressed port is presented back to the INS8048/49/50 on port P2. Termination of the read command occurs on the low to high transition of the PROG input. The port (4, 5, 6 or 7) that was addressed switches to the high impedance state and port 2 reverts to the input (read) mode. A port will normally be in either write (output) mode or read (input) mode. To allow for the settling of the external driver on the port, the first read following a write should be discounted when modes are changed during operation. All succeeding reads are valid.

WRITE MODES

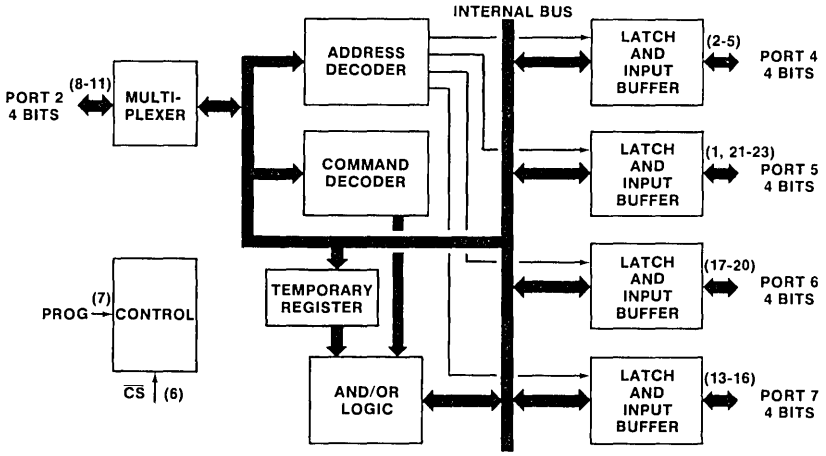
The INS8243 has three write modes that are initialized by the following instructions:

a. MOVD Pi,A—Instruction from the INS8048/49/50 writes new data directly into the addressed port. Existing data is lost.

b. ORLD Pi,A—Instruction from the INS8048/49/50 takes port P2 data, logically ORing it with the existing data in the addressed port and writes the resultant data into the port.

c. ANLD Pi,A—Instruction from the INS8048/49/50 takes port P2 data, logically ANDing it with the existing data in the addressed port and writes the ANDed data into the port.

Command and port address codes are latched from the port 2 input on the high to low transition of the PROG input. Data on port 2 is deposited in the logic circuits of the addressed port. When the logic manipulation has been performed, data is latched and output. Old data stays latched until valid new outputs are written.



Note: Applicable pinout numbers are included within parentheses.

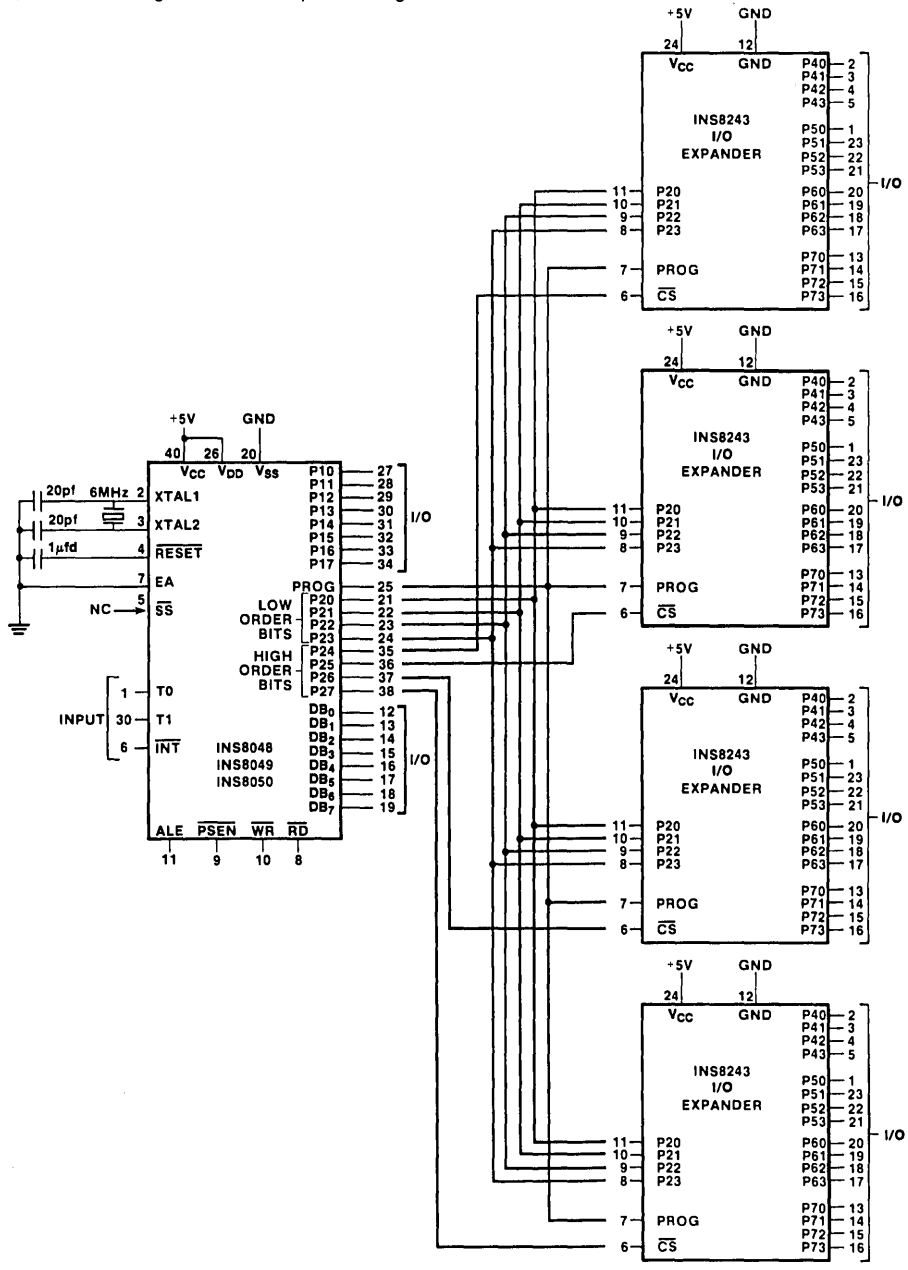
TL/C/7998–6

FIGURE 2. INS8243 I/O Expander Block Diagram

Functional Description (Continued)

MULTIPLE INS8243 I/O EXPANDER USE

To expand the resident I/O port capabilities of the INS8048/49/50 microcomputer series sixteen times, four INS8243 expanders may be connected to the microcomputer, as shown in *Figure 3*. By using the high order bit outputs P24 through P27 as the chip select ($\overline{CS}$) input to the I/O expanders, no additional logic circuits are required. A negative true input (low) on the $\overline{CS}$ line will enable the port 2 input for the selected I/O expander. A single INS8048/49/50 microcomputer may use many (up to twenty) I/O expanders without additional logic circuits on the same bus, limited only by the availability of chip select lines and loading restrictions.



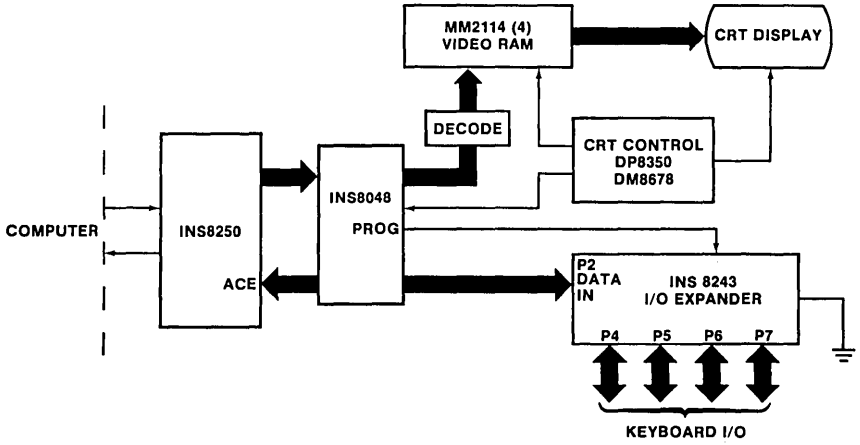
TL/C/7998–7

FIGURE 3. Typical Connections Using INS8243 I/O Expander

Functional Description (Continued)

Larger numbers of INS8243 expanders would require chip select decoder chips to conserve miocrocomputer I/O pins.

Operation of the INS8243 expander selected in the same as was explained in the functional description. *Figure 4* is a typical system application.



TL/C/7998–8

FIGURE 4. Typical System Application

Using E²PROM's with ROMless Single Chip Microcontroller

National Semiconductor Corp.
Application Brief
Joel Fishman

When developing programs for single chip microcontrollers, current thinking suggests that engineers use μvEPROMs as program memory. This technology offers the advantage of non-volatility yet allows the designer to change the program when necessary. This technology has manifested itself in ROMless versions of the COPS™ 4-bit and 8048 8-bit families with parts such as the 8035 and 87P50 piggyback version.

The major disadvantage of this technology is that the entire chip must be erased and reprogrammed regardless of the size of the change. The chip erase cycle takes 20 minutes, typically, and as such, lengthens the software development cycle. Although a number of μvEPROMs' may be held as spares for reprogramming this is not the most efficient method available.

Emerging Electrically Erasable PROM (E²PROM) technology solves this problem. The entire chip may be programmable using a PROM programmer such as the one you'd use for the μvEPROMs. In addition, thanks in part to the 5V only operation, on-board address and data latches, self-timed writing, and single byte programming the E²PROM may be modified in the system with minimal hardware overhead. This application note shows how to design the hardware to interface a ROMless version of the 8048 family with the NMC9816A and NMC9817A, National's 16k (2k x 8), 5V only E²PROM.

When making program changes manually, it may be more efficient to just make patches than to reprogram large sections of memory. After the program is running, a final step would be to reassemble. This manual technique is also suitable for changing minor errors in instruction coding.

Description

Figure 1 shows the block diagram required to implement a ROMless 8-bit microcontroller with 4k bytes of E²PROM.

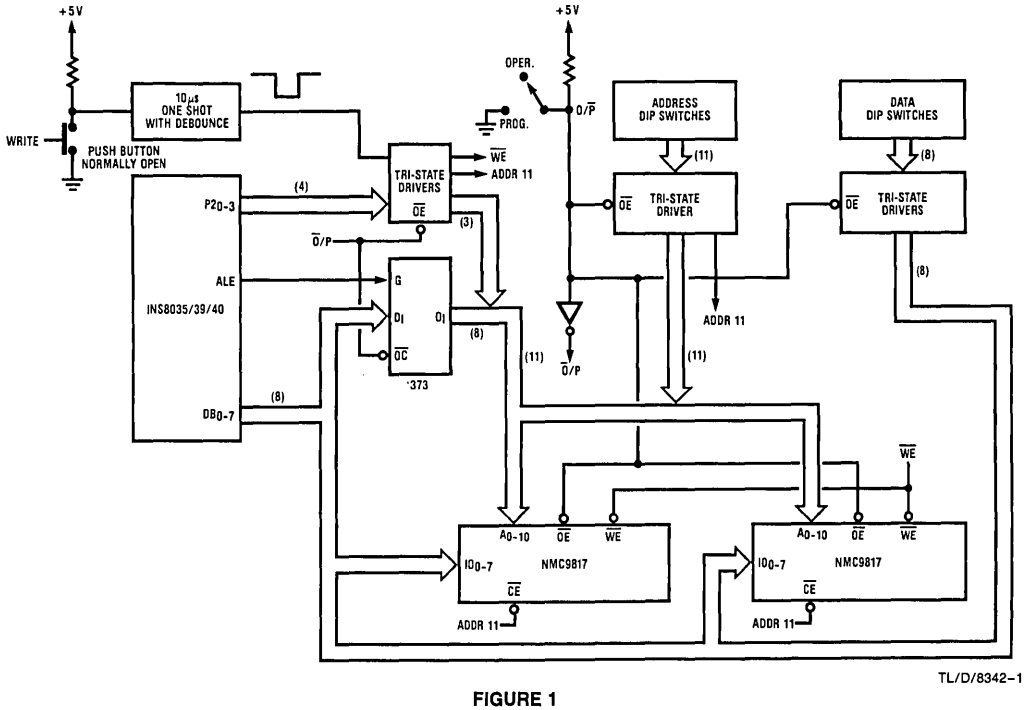
Normal operation of the system occurs when the PROG/OPER switch is in the OPER position. This enables the output of the address latch from the microcontroller, while putting the TRI-STATE® drivers from the DIP switches to the high impedance state. The system functions as current designs using EPROMs.

When a location in memory needs to be changed the switch is set to the PROG position. This enables the DIP SWITCH DRIVERS. The address of the byte to be modified and the data to be written are set on the binary DIP switches. The WRITE push button is pressed, generating a 10 μsec negative going pulse.

The write pulse ($\overline{WE}$) latches the address and the data into the proper 2k page of memory, as selected by ADDR11 and the write cycle takes place.

This technique may be used to change one byte, a few bytes, or to put a patch into the software. If a routine is incorrect a jump instruction to a blank area of memory can be used to create a new routine.

Block Diagram



TL/D/8342-1

FIGURE 1

Internal ROM Verification For The 48-Series Microcontrollers

National Semiconductor Corp.
Application Note 345
Venkata Gobburu

INTRODUCTION

The introduction and development of semiconductor industry has also spawned associated problems in the area of testing the devices. Testing the devices is a major expenditure. As the complexity of the device increases the testing costs escalate. This is especially true for microcontrollers since they incorporate complete microcomputer systems. Verifying the program code sequences mask programmed into the microcontroller forms an integral part of testing the devices. This note briefly discusses the ideas involved in carrying out a ROM dump for the 48-series of microcontrollers from National Semiconductor. At the end of the discussion a procedure to ensure a successful ROM dump is recommended.

THE 48-SERIES MICROCONTROLLERS

The 48-series family of microcontrollers are complete 8-bit computer systems incorporating the required ALU, RAM, ROM and I/O with diverse technologies such as NMOS, CMOS and XMOS. The ideas and principles involved being the same, the discussion revolves around the INS8048 without any loss of generality.

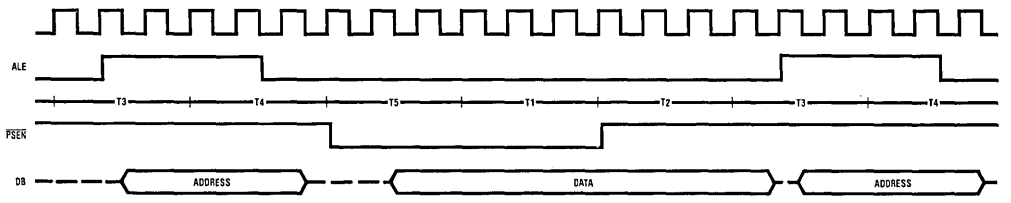
The internal clock circuitry for the INS8048 accepts inputs from two pins viz XTAL1 and XTAL2. A crystal or an externally generated source can be connected across the two pins. The external clock frequency of the oscillator is divided by three to provide the basic clock for the system. Each clock comprises a single state and is referred to as a Machine Cycle. Five machine cycles comprise a single Instruction Cycle. To differentiate between the different machine cycles we refer to them as T1, T2, T3, T4 and T5. *Figure 1* pictorially summarizes these relationships and indicates the sequence of operations that normally take place in an instruction cycle. The microcontroller continuously generates the Address Latch Enable (ALE) signal.

ROM VERIFICATION MODE

In a nutshell the ROM verification program consists of three distinct steps. The first step involves suspending the normal instruction execution mode of the microcontroller. The second step requires presenting specific ROM location addresses to the microcontroller. The microcontroller responds by placing the contents of the addressed ROM location on the bus. Reading the contents of the ROM location from the bus constitutes the final step. Repeating this sequence of three steps allows the contents of a block of ROM locations to be verified. We now detail the three steps.

The microcontroller contains logic which disables normal instruction execution mode and forces it into a special ROM verification mode. The logic triggers the changeover to the ROM verification mode when it senses the voltage at the Enable Access (EA) pin to be greater then +6V. This condition remains active as long as the voltage at the EA pin, typically +12V, is present.

The second step in the sequence requires specific ROM locations to be addressed. The addresses are presented as in the regular mode via the ports DBO-DB7 and P20-P23. To avoid possible bus conflicts it becomes imperative to ensure that the two ports are in an input mode. This can be very simply done by resetting the device. The microcontroller transfers the address presented at DBO-DB7 and P20-P23 to the Program Counter (PC). The transfer is done in two stages. The two stage transfer is necessitated by the fact that the PC is 12 bits wide whereas the internal data bus is only byte wide. An internal control signal clocks the lower order and the higher order sections into the PC. The data transfers are done during T2 and T3 machine cycles. The higher order bits of the PC are loaded in before the lower order bits. Obviously two complete instruction cycles are minimally required to latch in the address. Extra instruction cycles can be included to provide a wider margin of safety.



TL/C/5514–1

FIGURE 1. Instruction Fetch from External Program Memory

The third step in the sequence consists of ensuring that the address has been latched and then reading the ROM contents placed on the bus in response. The microcontroller continues to alternately latch in the two sections of the PC. This can be discontinued by pulling the $\overline{\text{RESET}}$ pin high. Care must be exercised when doing this since a transition occurring during a transfer can latch undefined addresses. A point to be borne in mind when attempting to change the state of the $\overline{\text{RESET}}$ pin is that the microcontroller does not respond immediately. The $\overline{\text{RESET}}$ pin is polled continuously during the T4 machine cycle, at the end of which the machine recognizes the reset signal. The next step consists of disabling the external circuitry holding the address on DB0-DB7 preparatory to reading the contents of the ROM location. The falling edge of the ALE signal can be used to disable the external circuitry. This is possible because of two internal conditions. First the ALE signal goes low during the T4 machine cycle and second the contents of the ROM location are enabled onto DB0-DB7 in the T1 machine cycle.

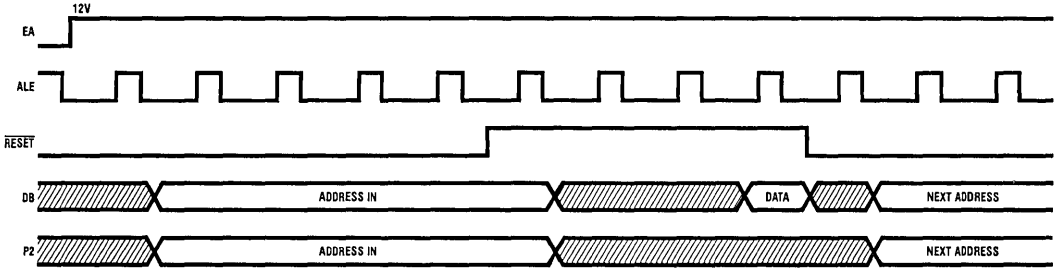
The preceding discussion highlights the different points to be borne in mind when attempting a ROM dump. The factors and their effects have been minutely described. The next section presents a ROM dump procedure.

THE RECOMMENDED ROM DUMP PROCEDURE

A procedure to guarantee reading the contents of the internal ROM locations is presented in this section. It is assumed that a Megatest or Fairchild type test setup is being used. The procedure is based upon the factors outlined in the previous section. The minimum timings obtained from it are suitably modified to include a margin of safety. The timings have been specified in machine cycle units rather then absolute timings so that they are valid over the entire frequency of operation.

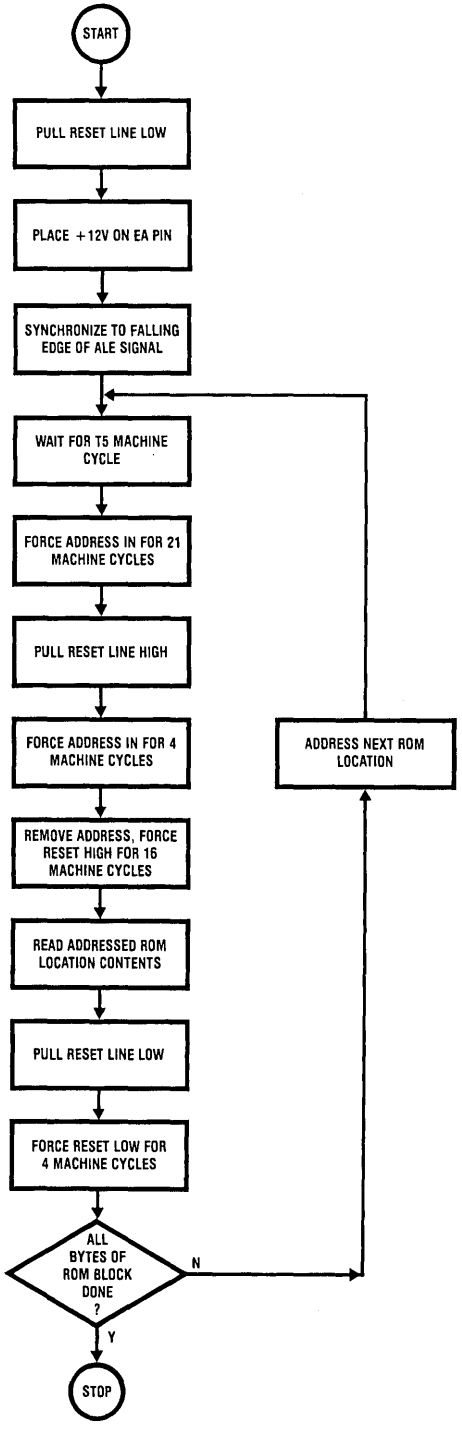
1. $\overline{\text{RESET}}$ must be low before EA goes high.
2. Synchronize to the falling edge of the ALE signal. Falling edge of the ALE indicates T4, the fourth machine cycle.
3. Starting in T5 force address in with $\overline{\text{RESET}}$ low for 21 machine cycles.
4. Force address in with $\overline{\text{RESET}}$ high for 4 machine cycles.
5. Force $\overline{\text{RESET}}$ high for 12 machine cycles.
6. Force $\overline{\text{RESET}}$ high for 4 machine cycles. Data is valid on the bus till the falling edge of $\overline{\text{RESET}}$.
7. Force $\overline{\text{RESET}}$ low for 4 machine cycles.
8. Repeat steps 3 thru 7 for other addresses.

The procedure is presented pictorially *Figure 2. Figure 3* gives a flowchart for the recommended procedure. The timings recommended guarantee correct operation. The timings are obviously greater then the minimum required. The ROM dump circuitry is used only for verifying the ROM contents and was not designed to, in any way, be indicative of normal instruction fetch timing.



TL/C/5514-2

FIGURE 2. ROM Verification Timing Diagram



TL/C/5514–3

FIGURE 3. Flowchart of the Recommended ROM Dump Procedure

NS8050U MICROWIRE PLUS™ Interface

National Semiconductor Corp.
Application Note 358
Rao Gobburu
James Murashige

INTRODUCTION

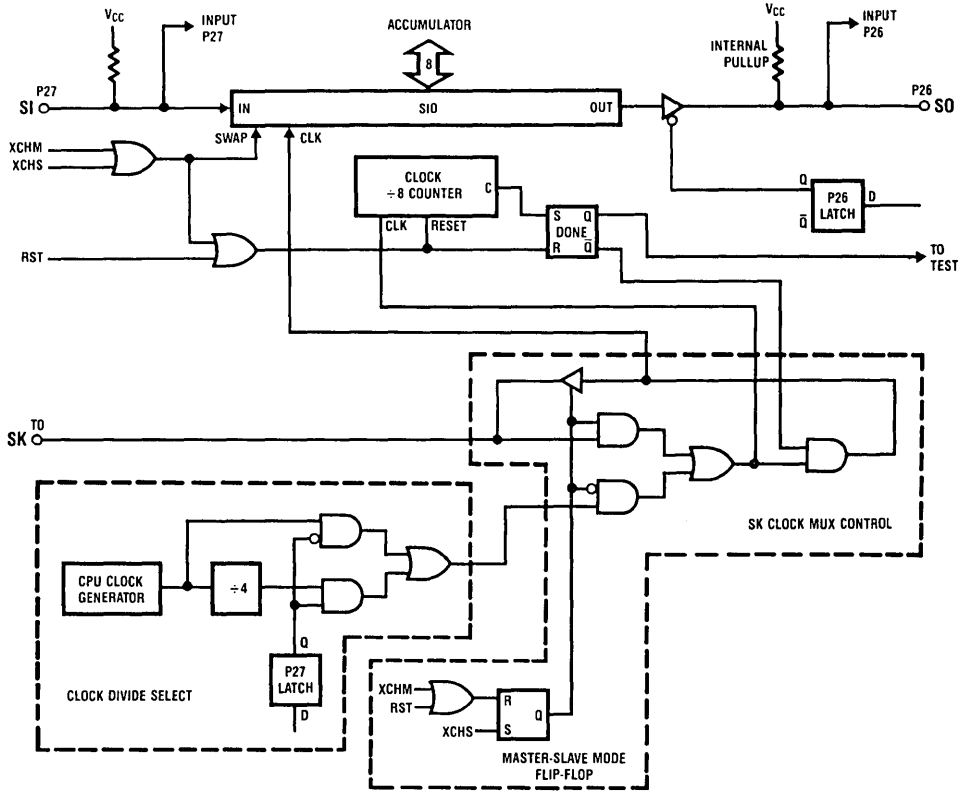
MICROWIRE™ is a serial communications interface from National Semiconductor. Originally devised to allow the COPS™ Microcontrollers to effectively communicate with peripheral devices, it has now been extended to the 8-bit 8048 microcontroller family. This extension is known as MICROWIRE PLUS.

The NS8050 from National Semiconductor, slated shortly for release, incorporates MICROWIRE PLUS. Its existence is entirely transparent to normal 8050 operation and is only activated upon execution of a new MICROWIRE PLUS instruction. This application note describes the features of the extension and presents programming examples to illustrate how to use MICROWIRE PLUS.

MICROWIRE PLUS

The MICROWIRE PLUS protocol utilizes a 3-wire interface working in conjunction with a clocked eight bit input/output shift register, *Figure 1*. The shift register is referred to as the Serial Input/Output (SIO) register. The three interface signals are Serial Output (SO), Serial Input (SI), and Serial Clock (SK). The contents of the accumulator may be exchanged with the SIO register thus providing us a means of performing the parallel to serial data conversion. Data waiting to be transmitted in the SIO register is clocked out on the SO pin on the falling SK clock edge. Serial data is received on the SI pin and clocked into the SIO register on the rising SK clock edge.

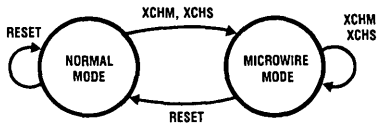
On reset the NS8050 comes up in the normal mode of operation. The MICROWIRE mode of operation may be invoked by executing one of two new instructions, XCHM or XCHS. Both instructions cause the Accumulator and SIO register to exchange data, with the differences being in the SK clock generation. In Master mode, set by XCHM, the SK clock is generated internally and output to other devices. In Slave mode, set by XCHS, the SK clock is input into the chip from a master source. Once the MICROWIRE mode has been selected it remains in effect until a system reset restores the normal mode of operation. The only practical difference



TL/DD/6103–1

FIGURE 1. Microwire Mode Functional Configuration

between the 2 operating modes are the dedication of 3 I/O pins to MICROWIRE operation. However, once in MICROWIRE mode, the chip may switch between Master and Slave operation at will.



TL/DD/6103–2

The 2 new instructions and their associated operations are as follows:

Mnenomic	Opcode	Operation
XCHM	C0	A < = > SIO. Reset DONE flip-flop, Clock counter. SK designated as an output. MICROWIRE mode selected.
XCHS	C1	A < = > SIO. Reset DONE flip-flop, Clock counter. SK designated as an input. MICROWIRE mode selected.

For MICROWIRE PLUS applications, one NS8050 must be designated as the Master. The Master supplies the SK clock for the system and initiates all data transfers between itself and one of the Slave devices. A Slave may be any of the standard MICROWIRE peripheral chips or another MICROWIRE PLUS NS8050 operating in Slave mode. In a typical system the Serial Clock (SK) is tied together on all the chips. The Serial Out (SO) from the Master is tied to the Serial In (SI) on all the Slaves. Similiarly, the Serial in (SI) pin on the Master is tied to the Serial Out (SO) on all the Slaves. General purpose outputs on the Master are used to chip select various Slave devices onto the MICROWIRE PLUS bus.

Among the various standard MICROWIRE peripherals available are display drivers (LCD, VF, LED), memories (RAM, EEROM), A/D converters, and frequency generators/timers.

NS8050 HARDWARE MODIFICATIONS

Three of the general purpose I/O pins on the NS8050 become dedicated signals when operating in MICROWIRE mode as follows:

NS8050 Pin	Function Under MICROWIRE PLUS	
T0	Serial Clock Input or Output	(SK)
P26	Serial Data Output	(SO)
P27	Serial Data Input	(SI)

In addition, upon entering MICROWIRE mode the port latches associated with P26 and P27 are disconnected from the port and used instead as configuration registers. They are still loaded by using the NS8050 port instructions with the following MICROWIRE PLUS functions selectable:

- P36 latch is used to Enable/Disable the TRI-STATE SO output.
  - 1 = Disabled
  - 0 = Enabled
- P27 latch is used to select the SK output frequency.
  - 1 = Instruction cycle clock divided by 4 (crystal freq/60)
  - 0 = Instruction cycle clock divided by 1 (crystal freq/15)

As in other chips in the 8048 family, RESET causes the latches associated with Ports 1 and 2 to set to a "1". Consequently if the P26 and P27 latches are not otherwise initialized, upon entering MICROWIRE mode the SO output will be disabled and divide by 4 selected for SK generation. However P26 and P27 may be modified at any time, even when in MICROWIRE mode.

When in MICROWIRE mode, port pins P26 and P27 may still be read in using the IN A, P2 instruction and will reflect the state of the SO and SI pins respectively. Note however that these pins also have internal pullup devices connected to them as shown.

To facilitate data transfers a DONE flip-flop has been included in the MICROWIRE circuitry. It and the clock counter are Reset upon every exchange between the Accumulator and the SIO register. When the clock counter reaches a count of 8, indicating that the SIO has completely shifted out, DONE is set. The DONE F/F is connected internally to the T0 sense line upon entering the MICROWIRE mode, thus allowing it to be tested by the JT0 and JNT0 instructions. Because of this any of the other T0 functions such as clock generation are precluded from operation while in MICROWIRE mode. It should be noted that the SK clock may only be shut down by the DONE F/F which in turn is driven by the Clock Counter. The Clock Counter may only be preset by 0 by an XCHM or XCHS instruction after which it will immediately start counting clock pulses.

APPLICATION NOTES

MICROWIRE PLUS may be effectively used for Local Area Networks (LANs) and Small Area Networks (SANs). Possible applications range from setting up a communications network within an automobile to home security systems. With the ability to switch between a MICROWIRE Master and Slave device at will, a multi-master NS8050 bus network may be implemented.

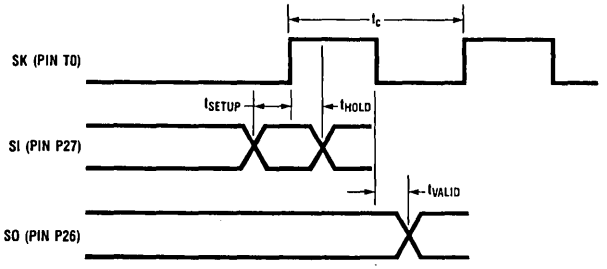
In NS8050 only systems MICROWIRE data may be transferred at the instruction cycle time of 1.36 μs (733 kHz). When using standard MICROWIRE peripherals data transfers must generally be restricted to 4 μs (250 kHz) as shown in *Figure 2a.* Also shown are typical MICROWIRE setup and hold times on the data lines relative to the SK clock. *Figure 2b* illustrates SIO bit shifting relative to instruction execution. When counting out instructions the XCH should be placed so that the last bit will be shifted in and out while the instruction is being read in as shown in *Figure 2b.*

For both divide by 1 and divide by 4 clock generation modes the rising SK clock edge is approximately coincident with the rising ALE edge as shown in *Figure 3.* For divide by 1 SK clock generation the duty cycle is 40% while for divide by 4 it is 50% (the ALE duty cycle is approximately 23%).

Since the same drive circuits are used for both normal and MICROWIRE modes of operation, the DC electrical characteristics are the same for the 3 I/O pins in either mode.

ON THE MICROWIRE 8050 THE CRYSTAL OSCILLATOR PINS ARE REVERSED FROM ORDINARY 8050s.

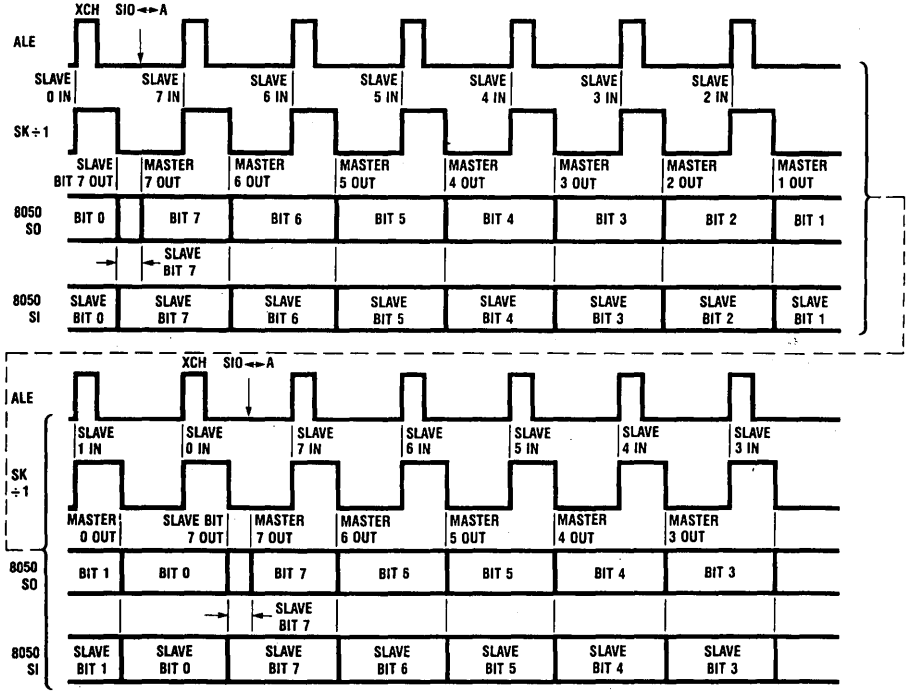
To facilitate 8050 MICROWIRE usage with standard COPS peripherals, a macro based I/O Driver Software Package is available. Written in 8048 assembly, it may be used directly or studied as an example of 8050 MICROWIRE peripheral interfacing.



TL/DD/6103-3

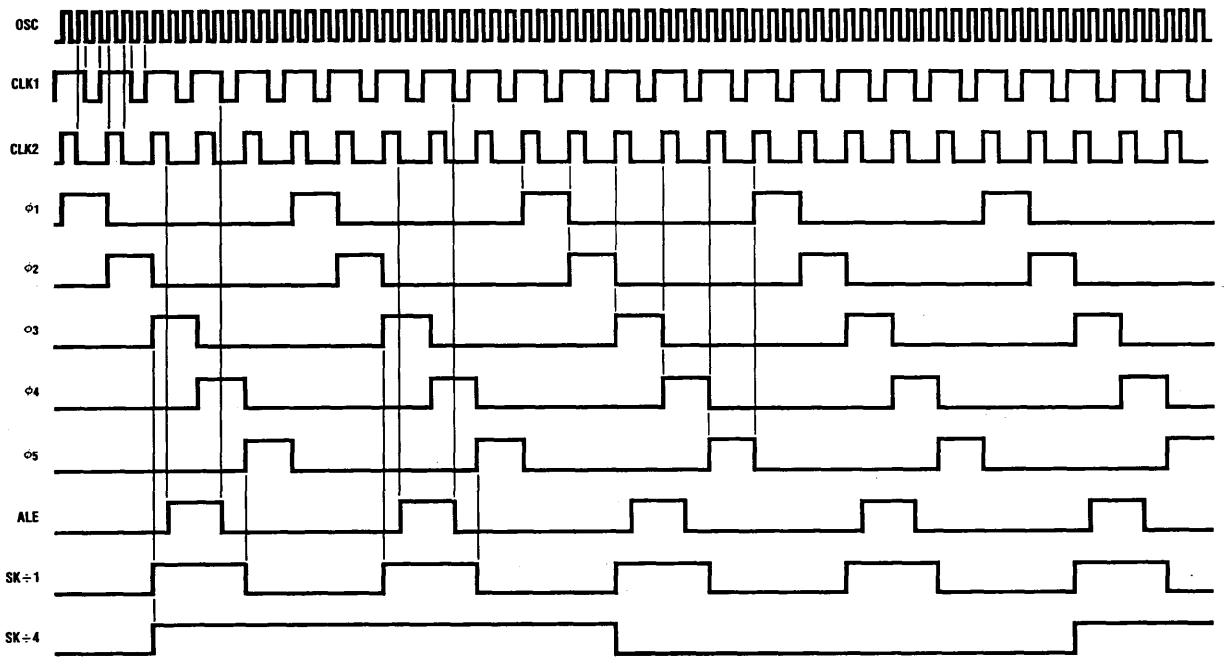
Parameter	Typical	Units
t_{SETUP}	200	ns
t_{HOLD}	200	ns
t_{VALID}	50	ns
t_C	4	ns

FIGURE 2a. Microwire Interface Timing



TL/DD/6103-4

FIGURE 2b. 8050 Microwire Serial Sequence



TL/DD/6103–5

FIGURE 3. NS8050 MIcrowire Clock Generation

MASTER-SLAVE PROTOCOL EXAMPLE

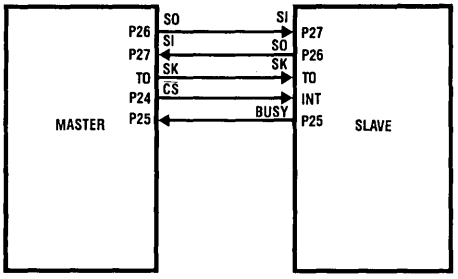
A number of handshake protocols are possible between a Master and Slave NS8050 in a MICROWIRE PLUS system. The following example illustrates one possible method and discusses the timing holds and restrictions on each device.

The hardware configuration for our example is shown in *Figure 4* and described as follows:

- Slave SO connected to Master SI
- Slave SI connected to Master SO
- Slave SK connected to Master SK
- Master pin P24 used to chip select slave, connected to Slave interrupt input
- Slave pin 25 connected to Master pin P25, used by slave as a BUSY output indicator
- Master and Slave have the same crystal frequencies to simplify things

Before starting our example we will also assume the following initial conditions:

- Master DONE flip-flop set causing SK clock generation to be shut down
- Master Clock divide by 4 selected (P27 latch set), SO enabled (P26 latch reset)
- Master P24 (Slave chip select) is high (inactive)
- Slave BUSY output (P25) is high (active)
- Slave DONE flip-flop is indeterminate

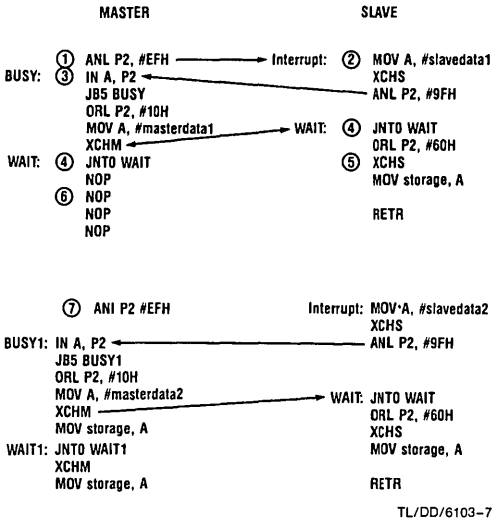


TL/DD/6103–6

FIGURE 4. Example Master-Slave Hardware Configuration

In our example of *Figure 5* the Master is performing a 2-byte write and read from the Slave. The following is a step by step description of the key steps in code execution.

1. Master wishes to send and/or receive data from Slave. He starts by activating the Slave chip select.
2. Slave chip select causes an interrupt service call where Slave loads his first data byte into the SIO, enables SO and resets BUSY. The SIO load also clears out the clock counter and DONE flip-flop.
3. All this time the Master was checking and waiting for BUSY to go away, thus signifying that the Slave has rec-



TL/DD/6103–7

FIGURE 5. Master-Slave Handshake Execution

organized the chip select interrupt. The chip select is turned off and the Master loads his first data byte into the SIO thus commencing transmission.

4. Both Master and Slave now wait for 8 data bits to be transmitted as indicated by the DONE flip-flop.
5. Upon seeing that DONE is set, the Slave immediately turns BUSY back on. The Slave then saves the just received data byte, disables SO and returns from interrupt service.
6. After seeing that DONE is set, Master delays enough to ensure that the Slave has turned BUSY back on. Master delay might also want to be put in to allow the Slave time to return from subroutine and utilize the just received data byte or update pointers in preparation for the next interrupt service. Alternatively the Slave could disable interrupts until housekeeping had been performed.
7. The Master wants to send another data byte so he activates the Slave chip select once again. The sequence for the Slave is the same as before. In the Master the XCHM to load the second Master data byte into the SIO also brings in the first Slave data byte. It is stored and after the current transmission is through the second Slave data byte may be brought in. The first Slave data byte could have been brought in after the first Master WAIT sequence with an XCHM, but that would have initiated another 8 SK Clock sequence which we would have had to wait out since SK must be shut down before initiating another transfer.
8. Rather than use straight line code, memory pointers and loop counters could easily be incorporated if the handshake restrictions described above are accounted for.

Section 7

Display/Terminal Management Processor (TMP)

Section 7 Contents

Terminal Management Processor

The TMP (NS405 series) is a single-chip CRT terminal display controller. The TMP is supported by the MOLE™ development system and replaces all the following LSI circuits commonly found in a terminal:

- Microprocessor
- Program ROM
- 64 x 8 RAM
- CRT controller
- DMA controller
- Character generator
- UART
- BAUD rate generator
- Parallel I/O controller
- Timer

The TMP offers complete CRT control over a wide scope of high-density circuit applications including phones, keyboard integration assignments, logic analyzers and more.

The NS455 Terminal Management Processor (TMP) demo board is available for design support.

Highly compact, the TMP board reduces previously necessary board space dramatically while providing 100% emulation of a classic low-end terminal. The board can also be used for TMP evaluation or as a vehicle for designing-in the NS405 device.

The board which is controlled by a preprogrammed NS455, needs only a video monitor, ASCII encoded keyboard, and power supply to provide your complete terminal. Should you wish to write your own program, no problem.

The cross-assembler software provides the capability. The board will execute custom programs through up to 8k of off-chip memory.

The TMP demo board comes complete with operating manual, program source listing, board schematic, board layout, and all necessary connectors.

When you're ready to design your own TMP system, turn to National's MOLE development system. By using this system-comprised of brain board, personality board and software—you bring dedicated development support to the TMP chip, making design-in extremely fast and simple.

National Semiconductor Corporation

PRELIMINARY

NS405/NS32405-Series Display/Terminal Management Processor (TMP)

General Description

The NS405 is a CRT terminal controller on a chip. It is a microcomputer system which replaces the following LSI circuits commonly found in a CRT data terminal:

- Microcomputer
- CRT Controller
- DMA Controller
- Character Generator
- UART
- Baud Rate Generator
- Interrupt Controller
- Parallel I/O Controller
- Timer

In addition the NS405 includes powerful attribute logic, two graphics display modes, and fast video output circuits.

The NS405 is primarily intended for use in low-cost terminals, but contains many features which make it a superior building block for "smart" terminals and word processing systems.

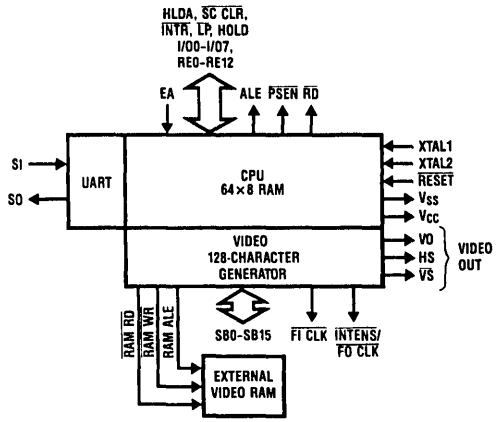
The NS405 interfaces easily to the display monitor, keyboard, display memory, and I/O ports. The architecture and instruction set are derived from the 8048-series microcontrollers. The instruction set has been enhanced and the architecture tailored to allow the NS405 CPU to efficiently manage a large display memory and an extensive interrupt environment.

The TMP can be used to easily and inexpensively add a display to many systems where it was previously impractical, it is not limited to terminal applications.

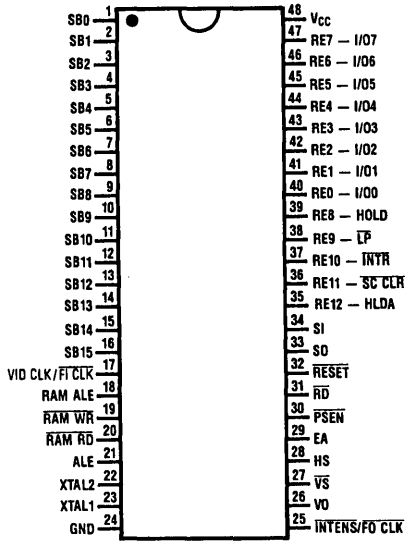
Features

- Enhanced 8048 instruction set and architecture
- Up to 8k x 8 ROM external with ROM expand bus
- On-board RAM 64 x 8
- Programmable display format
- On-board video memory management unit
- 16-bit bidirectional display memory bus (direct video and attribute RAM interface)
- Built-in timer
- Real-time clock (may be programmed for 1 Hz)
- Video control signals
- Eight independent attributes
- Pixel and block graphics display modes
- Programmable cursor characteristics
- Programmable CRT refresh rate
- Light pen feature
- UART, programmable baud rate up to 19.2k baud
- Character generator (128 characters 7 x 11 max)
- Single 5-volt supply @ 110 mA (typ)
- Up to 18 MHz video dot rate (12 MHz CPU clock)
- 48-pin package
- 8-bit parallel I/O port (multiplexed with external ROM)
- Extensive I/O expansion capabilities
- Up to 64k by 8 or 16 video RAM

Block and Connection Diagrams



TL/DD/5526-1



Top View

TL/DD/5526-2

Absolute Maximum Ratings

Specifications for Military/Aerospace products are not contained in this datasheet. Refer to the associated reliability electrical test specifications document.

Temperature Under Bias	0°C to +70°C
Storage Temperature	−65°C to +150°C
All Input or Output Voltages with Respect to V_{SS}*	−0.5V to +7.0V
Power Dissipation	1.5W

*EA, SI and VSYNC may be subjected to V_{SS} + 15V.

Note: *Absolute maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended; operations should be limited to those conditions specified under DC Electrical Characteristics.*

ESD rating is to be determined.

DC Electrical Characteristics

T_A = 0°C to +70°C, V_{CC} = +5V ±10%, V_{SS} = 0V, unless otherwise specified

Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
V_{IL1}	Input Low Voltage (All Except XTAL1, XTAL2, $\overline{RESET}$)		−0.5		0.8	V
V_{IH1}	Input High Voltage (All Except XTAL1, XTAL2, $\overline{RESET}$)		2.0		V_{CC}	V
V_{IL2}	Input Low Voltage (XTAL1, XTAL2, $\overline{RESET}$)		−0.5		0.6	V
V_{IH2}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$)		3.8		V_{CC}	V
V_{OL}	Output Low Voltage (All Except INTENS, VO)	I_{OL} = 2.0 mA			0.4	V
V_{OH}	Output High Voltage (All Except INTENS, VO)	I_{OH} = −125 μA	2.4		V_{CC}	V
V_{OL}	Output Low Voltage (INTENS, VO)	I_{OL} = 5.0 mA			0.4	V
V_{OH}	Output High Voltage (INTENS, VO)	I_{OH} = −500 μA	2.4			V
I_L	Input Leakage Current (EA, $\overline{INT}$, SI)	$V_{SS} \le V_{IN} \le V_{CC}$			±10	μA
I_{OL}	Output Leakage Current (ROM Expand Bus, High Impedance State)	$V_{CC} \ge V_{IN} \ge V_{SS}$ + 0.45			±10	μA
I_{OL}	Output Leakage Current (System Bus, High Impedance State)	$V_{CC} \ge V_{IN} \ge V_{SS}$ + 0.45			±100	μA
I_{CC}	Total Supply Current	T_A = 25°C		110	150	mA

AC Electrical Characteristics

T_A = 0°C to +70°C, V_{CC} = +5V ±10%, V_{SS} = 0V, unless otherwise specified

Symbol	Parameter	Min	Typ	Max	Units
PU AND ROM EXPAND BUS TIMING					
F_{XTAL}	Crystal Frequency	3		18	MHz
F_{CPU}	CPU Frequency	3		12	MHz
t_{CY}	CPU Cycle Time	1.25		7.5	μs
t_{DF}	Video Dot Time	55.5		333.3	ns
t_{LL}	ALE Pulse Width (Note 1)	125			ns
t_{AL}	Address Setup to ALE (Note 1)	55			ns
t_{LA}	Address Hold from ALE (Note 1)	40			ns
t_{CC}	Control Pulse Width $\overline{PSEN}$, $\overline{RD}$ (Note 1)	250			ns
t_{DR}	Data Hold (Notes 1, 4)	0		100	ns
t_{RD}	$\overline{PSEN}$, $\overline{RD}$ to Data In (Note 1)			220	ns
t_{AD}	Address Setup to Data In (Note 1)			360	ns
t_{AFC}	Address Float to $\overline{RD}$, $\overline{PSEN}$ (Note 1)		0		ns
t_{CAF}	$\overline{PSEN}$ to Address Float (Note 1)		±10		ns
t_{DAL}	Data Setup to ALE (RE11, 12) (Note 1)	55			ns
t_{ALD}	Data Hold from ALE (RE11, 12) (Note 1)	40			ns
t_{CIS}	Control Input Setup to ALE (RE8, 9, 10) (Note 1)	240			ns
t_{CIH}	Control Input Hold from ALE (RE8, 9, 10) (Notes 1, 4)	0		125	ns

AC Electrical Characteristics

T_A = 0°C to +70°C, V_{CC} = +5V ±10%, V_{SS} = 0V, unless otherwise specified (Continued)

Symbol	Parameter	Min	Typ	Max	Units
SYSTEM BUS TIMING					
t_{EL}	RAM ALE Low Time (Note 1)	250	280		ns
t_{EH}	RAM ALE High Time (Note 1)	100	120		ns
t_{AS}	Address Setup to RAM ALE (Note 1)	20	70		ns
t_{AH}	Address Hold from RAM ALE (Note 1)	10	50		ns
t_{RCY}	Read or Write Cycle Time (Note 1)		420*		ns
t_{RR}	$\overline{\text{RAM RD}}$ Width (Note 1)	210	240		ns
t_{AR}	Address Setup to $\overline{\text{RAM RD}}$ (Note 1)	80	120		ns
t_{RRD}	Data Access from $\overline{\text{RAM RD}}$ (Note 1)			140	ns
t_{RDR}	Data Hold from $\overline{\text{RAM RD}}$ (Notes 1, 4)	0		60	ns
t_{WFI}	$\overline{\text{FIFO In Clock}}$ Width (Note 1)	210	240		ns
t_{WW}	$\overline{\text{RAM WR}}$ Strobe Width (Note 1)	140	160		ns
t_{AW}	Address Setup to $\overline{\text{RAM WR}}$ (Note 1)	120	200		ns
t_{DW}	Data Setup to $\overline{\text{RAM WR}}$ (Note 1)	10	40		ns
t_{WD}	Data Hold from $\overline{\text{RAM WR}}$ (Note 1)	20	40		ns
VIDEO TIMING					
t_{DF}	Dot Period = $\frac{1}{f_c}$ (Note 1)		55		ns
t_{VID}	Video Blank Time $\frac{t_{DF}}{5}$ (Note 1)		±10		ns
t_{VI}	Skew, $\overline{\text{Intensity}}$ to Dot 0 (Note 1)		±15		ns
t_{FOV}	$\overline{\text{FIFO Out Clock}}$ to Dot 0 (Note 1)		15		ns
t_{WFOH}	$\overline{\text{FIFO Out Clock}}$ Width High (Note 1, Note 2)		55–165**		ns

*⅓ CPU cycle.

**1 Dot time is 55 ns.

Note 1: Control outputs C_L = 80 pF; ROM Expand Bus outputs C_L = 150 pF; System Bus outputs C_L = 100 pF; F_{XTAL} = 18 MHz; F_{CPU} = 12 MHz.

Note 2: $\overline{\text{FO CLK}}$ duty cycle is shown above.

Note 3: Hold request is latched. It is honored at the start of the next vertical retrace.

Note 4: Max spec. listed for user information only, to prevent bus contention.

Character Cell Width	$\overline{\text{FIFO Out}}$ HIGH	$\overline{\text{FIFO Out}}$ LOW
6	1 dot	5 dots
7	2 dots	5 dots
8	2 dots	6 dots
9	3 dots	6 dots
10	3 dots	7 dots

Input Hold Times

T_A = 25°C, V_{CC} = +5V ±10%, V_{SS} = 0V

Input	Min Active Time
Reset	50 ms (power up) 5 CPU Cycles (after power up)
$\overline{\text{External Interrupt}}$	2 CPU Cycle
$\overline{\text{Light Pen}}$	1 CPU Cycle
I/O Input	1 CPU Cycle
Hold Request	1 CPU Cycle (Note 3)

FIFO

Fall through should not be greater than 4 character times (character time = $1/f_{XTAL}$ × #dots/cell).

Throughput rate must be at least the character rate (character rate = 1/character time).

Capacitance $T_A = 25°C, V_{CC} = V_{SS} = 0V$

Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
C_{IN}	Input Capacitance	$F_C = 1$ MHz		6	10	pF
C_{OUT}	Output and Reset	Unmeasured Pins Returned to V_{SS}		10	20	pF

AC Electrical Characteristics in CPU Cycle Time

CPU AND ROM EXPAND BUS TIMING

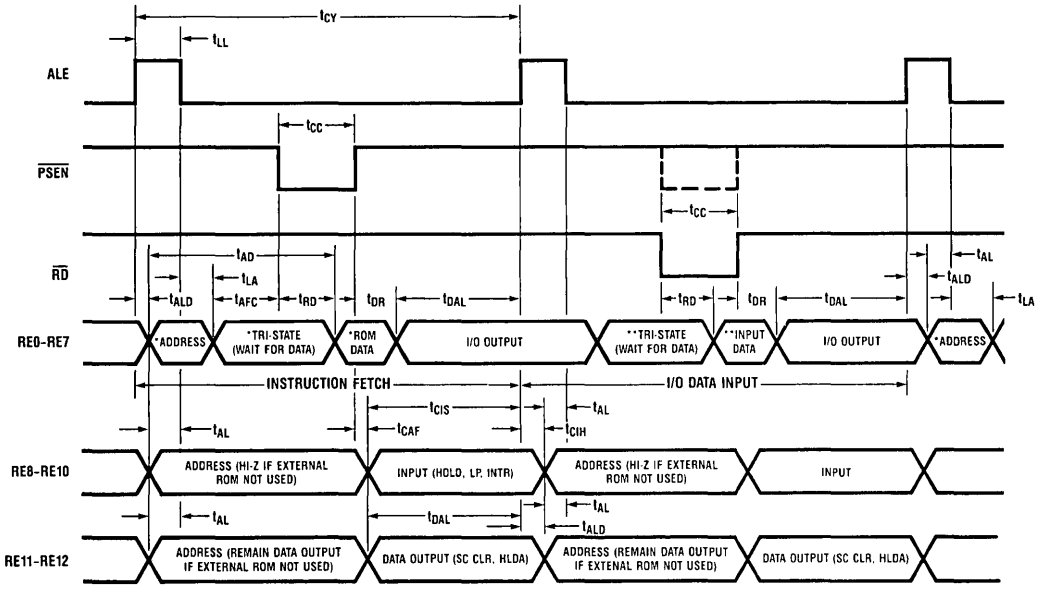
Symbol	Parameter		Typ
t_{LL}	ALE Pulse Width		$14\ t_{CY/60}$
t_{AL}	Address Setup to ALE		$8\ t_{CY/60}$
t_{LA}	Address Hold from ALE		$6\ t_{CY/60}$
t_{CC}	Control Pulse Width	$\overline{PSEN}$ $\overline{RD}$	$24\ t_{CY/60}$ $36\ t_{CY/60}$
t_{CY}	CPU Cycle Time		$60\ t_{CY/60} = 15/f_{CPU} = \frac{15}{f_{XTAL} \div 1 \text{ or } \div 1.5}$
t_{DR}	Data Hold		$-2\ t_{CY/60}$
t_{RD}	Control Pulse to Data In	$\overline{PSEN}$ $\overline{RD}$	$18\ t_{CY/60}$ $30\ t_{CY/60}$
t_{AD}	Address Setup to Data In		$32\ t_{CY/60}$
t_{AFC}	Address Float to	$\overline{PSEN}$ $\overline{RD}$	$2\ t_{CY/60}$ $2\ t_{CY/60}$
t_{CAF}	PSEN to Address Float		$0\ t_{CY/60}$
t_{DAL}	Data Setup to ALE	RE0–7 RE8–10 RE11–12	$6\ t_{CY/60}$ $-2\ t_{CY/60}$ $16\ t_{CY/60}$
t_{ALD}	Data Hold from ALE	RE0–7 RE8–12	$2\ t_{CY/60}$ $6\ t_{CY/60}$

SYSTEM BUS TIMING

Symbol	Parameter	Ticks		
		Min	Typ	Max
t_{EL}	RAM ALE Low Time	$14\ t_{CY/60} - 42$ ns	$14\ t_{CY/60}$	
t_{EH}	RAM ALE High Time	$6\ t_{CY/60} - 25$ ns	$6\ t_{CY/60}$	
t_{AS}	Address Setup to RAM ALE	$4\ t_{CY/60} - 60$ ns	$4\ t_{CY/60}$	
t_{AH}	Address Hold from RAM ALE	$2\ t_{CY/60} - 40$ ns	$2\ t_{CY/60}$	
t_{RCY}	Read or Write Cycle Time		$20\ t_{CY/60}$	
t_{RR}	$\overline{RAM\ RD}$ Width	$12\ t_{CY/60} - 40$ ns	$12\ t_{CY/60}$	
t_{AR}	Address Setup to $\overline{RAM\ RD}$	$6\ t_{CY/60} - 45$ ns	$6\ t_{CY/60}$	
t_{RRD}	Data Access from $\overline{RAM\ RD}$		$10\ t_{CY/60}$	$10\ t_{CY/60} - 70$ ns
t_{RDR}	Data Hold from $\overline{RAM\ RD}$			
t_{WFI}	$\overline{FIFO\ In\ Clock}$ Width	$12\ t_{CY/60} - 40$ ns	$12\ t_{CY/60}$	
t_{WW}	$\overline{RAM\ WR}$ Strobe Width	$8\ t_{CY/60} - 27$ ns	$8\ t_{CY/60}$	
t_{AW}	Address Setup to $\overline{RAM\ WR}$	$10\ t_{CY/60} - 90$ ns	$10\ t_{CY/60}$	
t_{DW}	Data Setup to $\overline{RAM\ WR}$	$2\ t_{CY/60} - 30$ ns	$2\ t_{CY/60}$	
t_{WD}	Data Hold from $\overline{RAM\ WR}$	$2\ t_{CY/60} - 20$ ns	$2\ t_{CY/60}$	

Timing Waveforms

ROM Expand Bus Timing
(In Port Instruction is Shown)

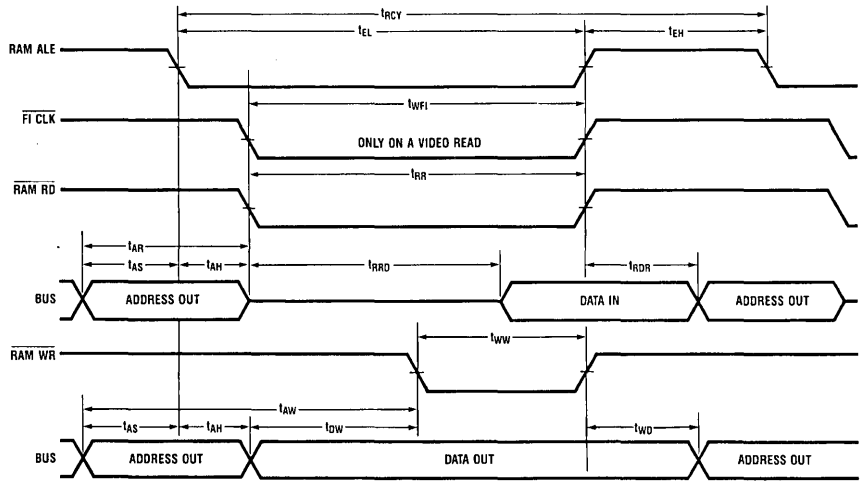


TL/DD/5526-3

*Remain I/O OUTPUT if External ROM not used.

**I/O Data input or 2nd ROM byte of 2 byte instruction. Otherwise remain I/O OUTPUT.

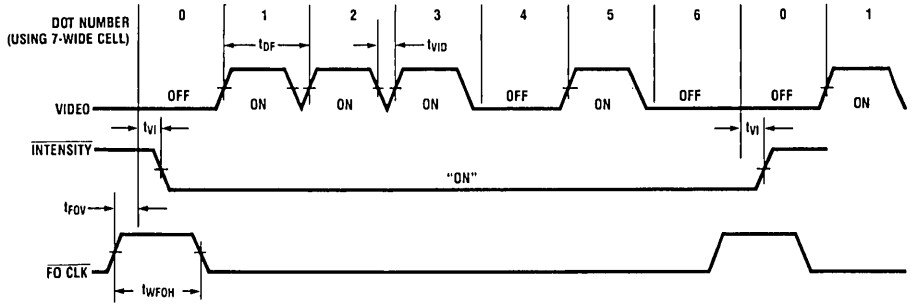
System Bus Timing



TL/DD/5526-4

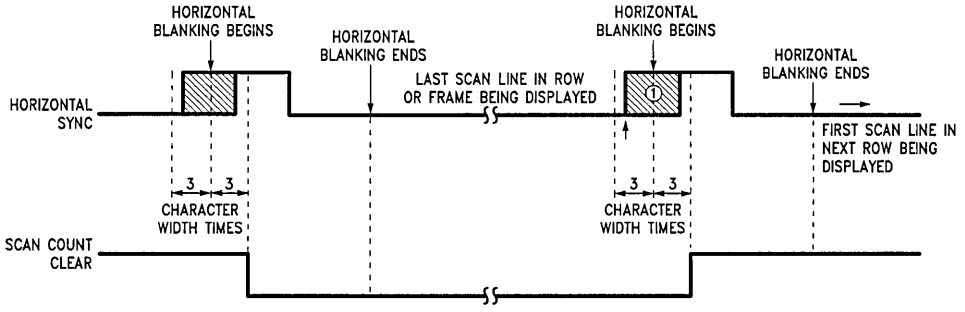
Timing Waveforms (Continued)

Video Timing



TL/DD/5526-5

Scan Count Clear Timing

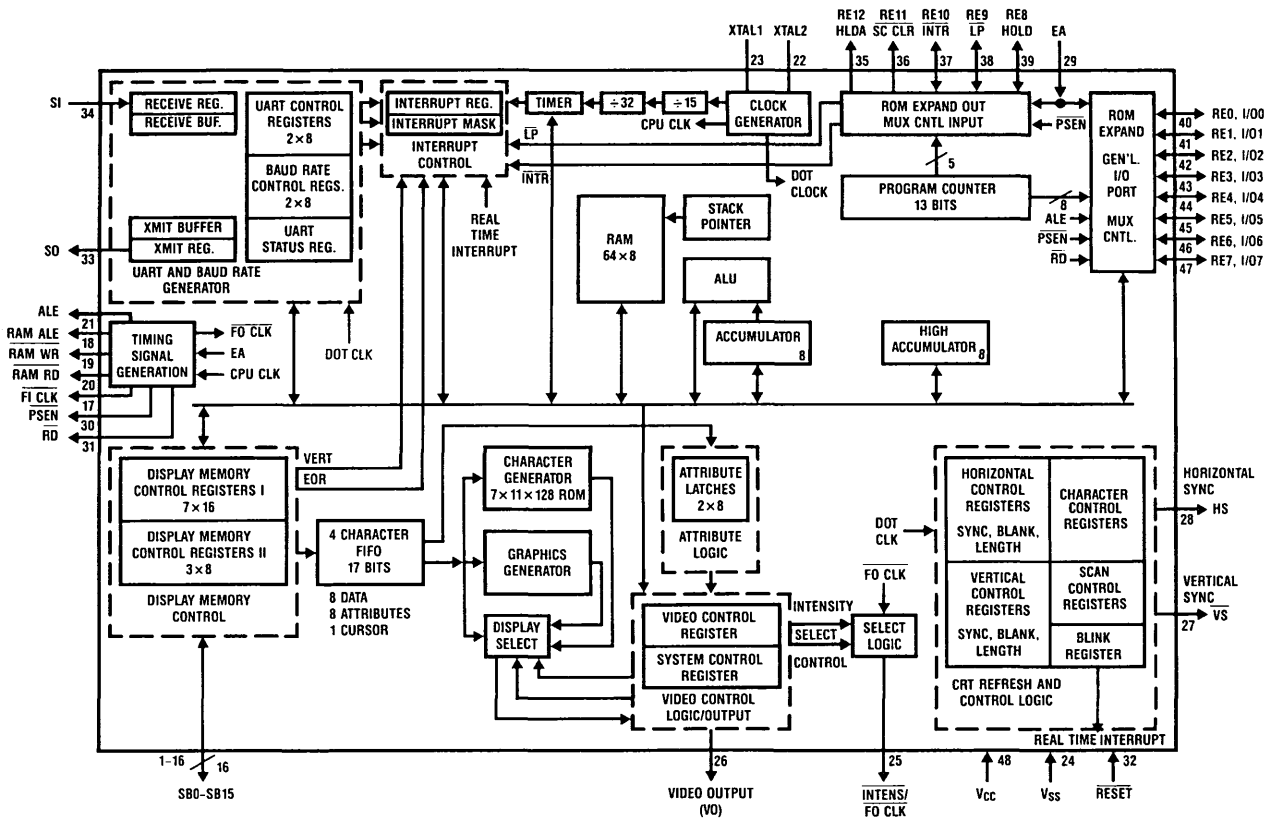


TL/DD/5526-6

For external character generation this edge is used to clock CLEAR into scan line counter. The edge must come before Scan Count Clear goes away, but not before the video controller has brought in all necessary display information for the last scan line.

Block Diagram

NS405-Series Detailed



TL/DD/5526–7

1.0 Functional Pin Descriptions

1.1 SUPPLIES

Pin	Name	Function
48	V_{CC} — Power	5V ±10%
24	V_{SS} — Ground Reference	

1.2 INPUT SIGNALS

23, 22	XTAL1, XTAL2 — Crystal 1, 2:	Crystal connections for clock oscillator (3–18 MHz).
29	EA — External Access:	Pull HIGH (V_{IH_2})
32	$\overline{\text{RESET}}$	An active low input that initializes the processor. The $\overline{\text{RESET}}$ input is also used for internal ROM verification.
34	SI — Serial Input:	Drives receiver section of UART (true data).

1.3 OUTPUT SIGNALS

33	SO — Serial Output:	Driven by transmitter section of UART (true data).
21	ALE — Address Latch Enable:	ROM address is available on the ROM Expand Bus and may be latched on the falling edge of ALE. Port output data may be latched on the rising edge of ALE. ALE pulses are always present, even if EA is tied low.
30	$\overline{\text{PSEN}}$ — Program Store Enable:	Enable external ROM output drivers when low. $\overline{\text{PSEN}}$ is idle (high) when the CPU fetches from internal ROM.
31	$\overline{\text{RD}}$ — Read Port Data:	Accept Port input data on ROM Expand Bus RE0–RE7 while low. ROM Expand Bus is in high impedance state while $\overline{\text{RD}}$ is low.
28	HS — Horizontal Sync	The rising edge of HS is controlled by the Horizontal Sync Begin Register and the falling edge is controlled by the Horizontal Sync End Register. HS is disabled (low) if bit 5 of the Video Control Register = 0.
27	$\overline{\text{VS}}$ — Vertical Sync Output:	The falling edge of $\overline{\text{VS}}$ is controlled by the Vertical Sync Begin Register and the rising edge is controlled by the Vertical Sync End Register. $\overline{\text{VS}}$ is at TRI-STATE if bit 5 of the Video Control Register = 0.
26	VO — Video Output:	High = beam on, low = beam off. VO is disabled (low) if bit 5 of the Video Control Register = 0.
25	$\overline{\text{INTENS}}$/$\overline{\text{FO CLK}}$	(Shared pin) $\overline{\text{INTENS}}$ Signal under attribute control may be used to switch the bistable brightness of display characters. $\overline{\text{FIFO Out Clock}}$ may be used to clock data from an external FIFO in synchronism with data from the internal FIFO. Both CANNOT be used simultaneously.
17	VID CLK/$\overline{\text{FI CLK}}$ — Video Dot Clock Out/ FIFO IN CLOCK	(Shared pin) The rising edge of the Video Dot Clock may be used to clock the data out of the video output pin. FIFO In Clock may be used to clock data from an extended attribute RAM into an external FIFO in synchronism with the data loaded into the internal FIFO. Both CANNOT be used simultaneously.
18	RAM ALE — RAM Address Latch Enable:	RAM address is available on the System Bus and may be latched on the falling edge of RAM ALE. Only operational when Display RAM accesses being performed. Otherwise high.
20	$\overline{\text{RAM RD}}$ — RAM Read:	Enable display RAM data onto the System Bus when $\overline{\text{RAM RD}}$ is low.
19	$\overline{\text{RAM WR}}$ — RAM Write:	Data to RAM is available on the System Bus and may be written at the rising edge of $\overline{\text{RAM WR}}$.

1.4 BUS — I/O

1–8	SB0–SB7 — System Bus 0–7:	Display RAM address is output while RAM ALE is high and may be latched on the falling edge of RAM ALE. System Bus accepts data input while $\overline{\text{RAM RD}}$ is low and outputs data while $\overline{\text{RAM WR}}$ is low.
9–16	SB8–SB15 — System Bus 8–15:	Normally, Display RAM address is output and held on these pins for the full read or write cycle. However, if bit 4 of the System Control Register is set, these pins function bidirectionally like SB0–SB7 to allow 16-bit data words for attribute operation.
35–47	RE0–12 — ROM Expand Bus 0–12:	Used for program ROM expansion as described below. Time multiplexed with I/O port and system control signals. I/O port and system control signals only if no external ROM used.
40–47	RE0–RE7	Low order ROM address is output and may be latched on the falling edge of ALE. Enable ROM data to this Bus when $\overline{\text{PSEN}}$ is low. Enable I/O port input data to the Bus when $\overline{\text{RD}}$ is low. Use the rising edge of ALE to latch port output data.

1.0 Functional Pin Description (Continued)

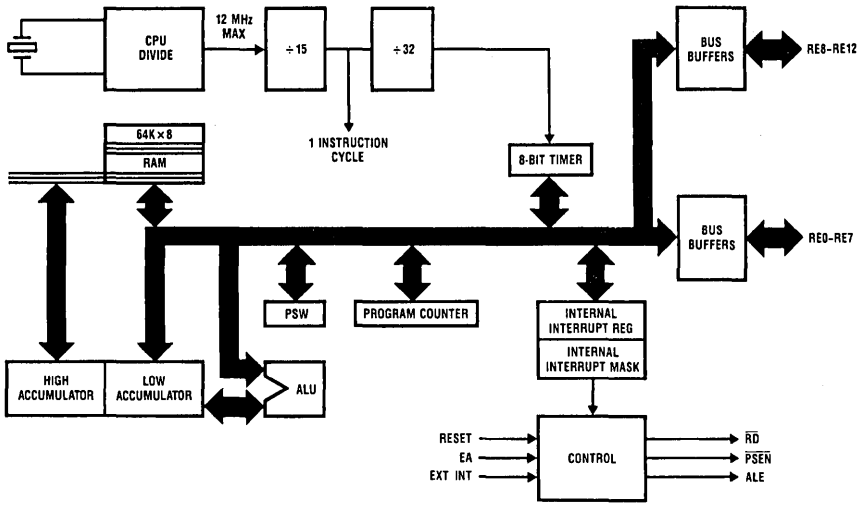
Pin	Name	Function
39–35	RE8–RE12	Five most significant bits of the ROM address are output during ALE and remain stable until data is read in during $\overline{PSEN}$. These pins are multiplexed with the HLDA, $\overline{INTR}$, $\overline{LP}$, $\overline{SC\ CLR}$, and HOLD signals.
37	$\overline{INTR}$ — Interrupt: RE10	An active low input that interrupts the processor if the external interrupt is enabled. Because it shares a pin with RE10, $\overline{INTR}$ may be driven directly only if no external ROM is used (EA is low). Otherwise must be driven through a 3.9k resistor.*
38	$\overline{LP}$ — Light Pen Interrupt: RE9	An active low input that interrupts the processor if internal interrupts are enabled and bit 5 in the Interrupt Mask Register is set. Because it shares a pin with RE9, $\overline{LP}$ may be driven directly only if EA is low. Otherwise, must be driven through a 3.9k resistor.*
39	HOLD — HOLD request: RE8	When high, requests that the NS405 enter the Hold mode. When in the Hold mode the System Bus will be in a high impedance state. The Hold mode is granted at the beginning of the next vertical retrace. Because it shares a pin with RE8, HOLD may be driven directly only if EA is low. Otherwise, must be driven through a 3.9k resistor.*
35	HLDA — Hold Acknowledge: RE12	This output is asserted in response to Hold and provides handshake capability with another processor (active high). For more detailed information see Section 3.0 Slave Processing. Because HLDA shares a pin with RE12, the HLDA state is preset only during the interval preceding the rising edge of ALE. However, if no external ROM is used, HLDA is a steady state output and need not be latched externally.
36	$\overline{SC\ CLR}$ — Scan Count Clear: RE11	This output clears an external scan counter when used with an external character generator. It is a low going pulse which occurs during the horizontal retrace preceding the first scan line of each character row. Because $\overline{SC\ CLR}$ shares a pin with the RE11, the correct $\overline{SC\ CLR}$ state is present only during the interval preceding the rising edge of ALE. However, if no external ROM is used, $\overline{SC\ CLR}$ is a steady state output and need not be latched externally.

*Unused control inputs must be terminated

2.0 Functional Description

2.1 CPU

The CPU of the NS405 is patterned after the 8048 single chip microcomputer (see *Figure 1)*.



TL/DD/5526–8

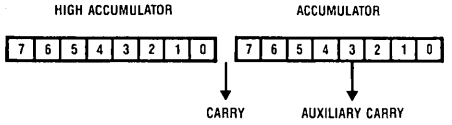
FIGURE 1. NS405 Series CPU Block Diagram

?.0 Functional Description (Continued)

.1.1 Accumulator — High Accumulator

ı addition to the regular 8-bit Accumulator, there is an 8-bit ligh Accumulator extension to facilitate the 16-bit operaons required for display memory management. The HACC/ CC pair is usually used in conjunction with the 16-bit RAM ointer registers (RA, R0 and RB, R1, CURSOR, HOME, EGD and ENDD) to effect video data transfers. In addition, xternal attribute memory is loaded in a 16-bit transfer opertion. Any instruction which causes a carry or borrow out of ıe low accumulator will affect the high accumulator (see *ïgure 2*).

uxiliary carry is used only when converting the accumulaır contents from binary to BCD (binary coded decimal) usıg the DA A instruction. The auxiliary carry flag can be leared by moving a zero into bit 6 of the program status ord.

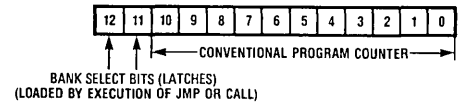


TL/DD/5526–9

FIGURE 2. CPU Accumulator

..1.2 Program Counter (PC)

he Program Counter is a 13-bit wide register which proides program addressing for the CPU. The lower 11 bits ıperate like a conventional program counter while the upper ! bits are actually latches. These 2 latches are automatically ɔaded from the bank select flip-flops (PSW bits 3, 4) whenıver a JMP or CALL instruction is executed. The bank seect flip-flops in turn are only modified upon the execution of ı Select Memory Bank Instruction or modification of the ?SW (see *Figure 3*).



TL/DD/5526–10

FIGURE 3. TMP Program Counter

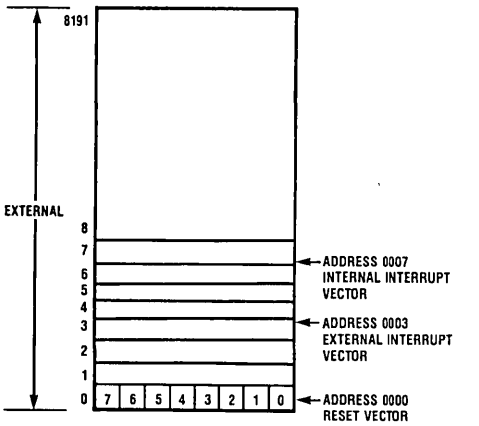
2.1.3 Program Memory

Memory is subdivided into 2k banks with accesses limited to the currently selected bank unless a Bank Change sequence has been executed. Upon reaching the end of a memory bank, the program counter will wrap around and point to the beginning of the current bank.

Each bank is further subdivided into pages of 256 bytes each, with 8 pages in every bank. The conditional JUMP instructions are restricted to operate within the memory page that they reside in.

Because of the sequence which the CALL instruction executes when pushing and loading the PC, it is possible to easily call and return from subroutines located in different memory banks (see *Figure 4*).

Upon executing an RET or RETR instruction for a call from one memory bank into another, a SEL MBx instruction should be excuted to restore the memory bank select flip-flops to their original bank. However, no SEL MBx is needed after an interrupt since the flip-flops were never modified.



TL/DD/5526–11

FIGURE 4. Program Memory Map

2.1.4 Program Status Word Bit Assignments

Bit Position	Contents
0	Stack Pointer Bit, S0
1	Stack Pointer Bit, S1
2	Stack Pointer Bit, S2
3*	Memory Bank Select Bit 0
4*	Memory Bank Select Bit 1
5*	Register Bank Select Bit (0 = Bank 0, 1 = Bank 1)
6*	Auxiliary Carry. A carry from Bit 3 to Bit 4 generated by an add operation. Used only by the decimal adjust (DA A) instruction.
7*	Carry. A bit indicating the preceding operation resulted in an overflow or an underflow from the 8-bit accumulator.

***Note 1:** Bits 3 through 7 are saved on the stack by subroutine calls or interrupts. Bits 3 and 4 are restored upon execution of an RET instruction, whereas all 5 bits are restored by RETR.

Note 2: F0 is not saved on the stack (as in an 8048).

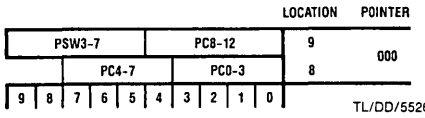
Note 3: Bits 0–5 cleared on a RESET.

2.1.5 Stack Pointer (SP)

The stack pointer is an independent 3-bit counter which points to designated locations in the internal RAM that holds subroutine return parameters. The stack itself is located in RAM locations 8–23 (see *Figure 5*).

Each entry in the stack takes up two bytes and contains both the PC and status bits. When reset to zero, the stack pointer actually points to locations 8 and 9 in RAM. Since the stack pointer is a simple up/down counter, an overflow will cause the deepest stack entry to be lost (the counter overflows from 111 to 000 and underflows from 000 to 111).

Note: If the level of subroutine nesting is less than eight (8), the unneeded stack locations may be used as RAM.



TL/DD/5526–12

Note: The odd numbered RAM bytes in the stack area have two (2) extra bits to allow for storage of the bank select switch bits. This feature allows interrupt routines and subroutines to be located outside the current 2k program memory bank.

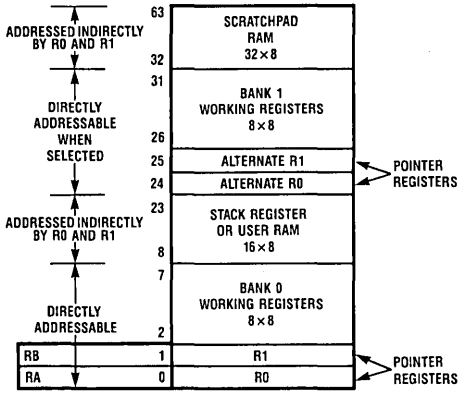
FIGURE 5. Typical Stack Composition

7

2.0 Functional Description (Continued)

2.1.6 Data Memory (On-Chip RAM)

The data memory nominally consists of 64 8-bit locations and is utilized for working registers, the subroutine stack, pointer registers and scratch pad. There are two sets of working/pointer registers (R0–R7) which are selected by the Select RAM Bank instruction. The stack area is located in locations 8–23. Locations 32–63 contain the scratch pad memory. To facilitate 16-bit Video Memory Management there are two 8-bit extension registers (RA and RB) which are associated with the R0 and R1 registers respectively of whichever RAM bank is currently selected (see *Figure 6*). i.e., There is only one RA register and only one RB register.

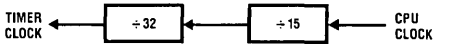


TL/DD/5526–13

FIGURE 6. RAM Memory Map

2.1.7 Timer

The On-Board Timer is an 8-bit up counter which sets the Timer Overflow Flag and generates an internal interrupt (if enabled) whenever it overflows from FF to zero. The Timer may be stopped, started, loaded and read from by the CPU. The Timer clock is derived from the CPU clock as shown in *Figure 7*. Whenever a Start Timer instruction is executed the ÷32 is initialized to its zero state to insure a full count measurement. After overflow the timer keeps counting until the next FF to zero overflow at which time the overflow flag will be set and another interrupt generated. The overflow flag can only be reset through the JTF and JNTF instructions.



TL/DD/5526–14

FIGURE 7. Timer Clock Generation

2.1.8 Interrupts

The interrupt circuitry handles two generic classes of interrupt conditions called Internal and External. Either class has its own master control which can be activated through software enable and disable instructions. On an interrupt service the currently executing instruction is completed, then two CPU cycles are used as the program counter and bits 3–7 of the PSW are pushed onto the stack and stack pointer is incremented.

Then the interrupt vector address (3 or 7) is loaded into the PC and service started. Whenever an interrupt condition is being serviced all other interrupts of either class are locked out until a RETR instruction is executed to conclude interrupt service. If both an external and internal interrupt arrive at the same time, the external interrupt is recognized first.

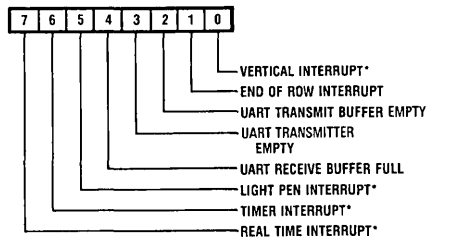
2.1.8.1 External Interrupt

The External Interrupt consists solely of the shared $\overline{\text{INTR}}$/RE10 pin. External interrupts on this pin will be detected if the setup and hold times as shown in the timing diagrams are met. This pin is a level sampled interrupt which means that as long as the pin is low during the sampling window an interrupt will be generated. In addition, the $\overline{\text{INTR}}$ pin is the only external pin whose logic state can be directly tested through software.

2.1.8.2 Internal Interrupts

The Internal Interrupts consist of seven internal operational conditions plus the light pen arranged in an 8-bit wide register as shown in *Figure 8*. Activation of an internal interrupt condition causes a corresponding register bit to be set, *Figure 9*. Each internal interrupt may be individually masked out through the Interrupt Mask register which has the same bit assignments as the Interrupt register and can be loaded from the accumulator. A zero in the Interrupt Mask register inhibits the interrupt and a one enables it. Further interrupt processing is as shown. To determine which of the eight internal conditions caused the interrupt the CPU must read the Interrupt register into the accumulator. To acknowledge receipt of the interrupt certain bits are automatically cleared on a read while others are reset upon service of the particular interrupt.

The conditions under which each of the interrupts are generated and cleared are as follows:



TL/DD/5526–16

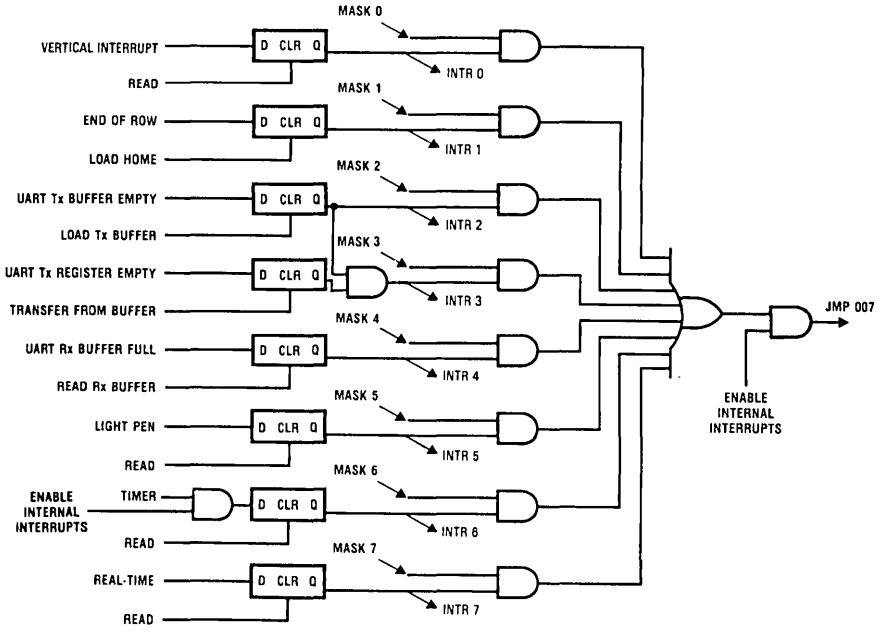
Note: The interrupt flags indicated by an asterisk (*) are cleared when the Interrupt Register is read.

FIGURE 8. Internal Interrupt Register

Bit

0 Vertical Interrupt—Generates an interrupt at the end of the display row designated by the Vertical Interrupt Register. Interrupt bit cleared on a CPU read of the interrupt register. If VIR > Vertical Length Register no interrupt will be generated.

2.0 Functional Description (Continued)



TL/DD/5526–15

FIGURE 9. Internal Interrupt Processing

Bit

1 End of Row Interrupt—Generates an interrupt at the end of each display row when the Current Row Start Register is updated for the next row. Used in conjunction with the Row Sequencing Control Bit (5) in the System Control Register to implement Row Pointer Look-Up Tables and Horizontally Split Screens. Interrupt bit cleared on a CPU write to the Home Register. Does not generate interrupts for those rows blanked during vertical blanking.

2 UART Transmit Buffer Empty—Generates an interrupt when the Transmit Buffer empties out after dumping a character into the Transmit Shift Register. Interrupt bit cleared on a CPU write to the Transmit Buffer.

3 Transmitter Empty—Generates an interrupt when BOTH the Transmit Buffer and Transmit Shift Register are empty. The interrupt bit is cleared when the CPU loads the transmit buffer.

4 UART Receiver Buffer Full—Generates an interrupt when the Receiver Buffer fills up with a character from the Receive Shift Register. Interrupt bit cleared on a CPU read of the Receiver Buffer.

5 Light Pen Interrupt—Generates an interrupt on each falling edge detected on the shared $\overline{LP}$/RE9 pin. Since only falling edges generate interrupts and the input is sampled each CPU Cycle, a high level must be sampled between falling edges in order to be considered a new interrupt. This interrupt is used to latch the light pen position registers. For further information see Light Pen Description. Interrupt bit cleared on a CPU read of the interrupt register.

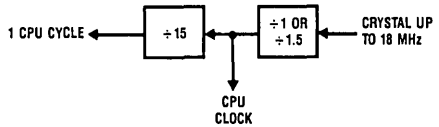
Bit

6 Timer Interrupt—Generates an interrupt when the internal 8-bit Timer overflows from FF to 00. Interrupt bit cleared on a CPU read of the interrupt register.

7 Real-Time Interrupt—Generates interrupts at a software programmable frequency that is generally in the Hertz range. (See CPU Clock Generation.) Thus permitting the implementation of a real-time clock or timer. Interrupt bit cleared on a CPU read of the interrupt register.

2.1.9 Clock Generation

All chip clocks are derived from the one external crystal connected between pins 22 and 23. This master clock also doubles as the video dot clock. The crystal frequency is constrained to lie within the range of 3 to 18 MHz. The CPU clock is derived from the crystal clock by either using it directly or by dividing down by a factor of 1.5 *(Figure 10)*.



TL/DD/5526–17

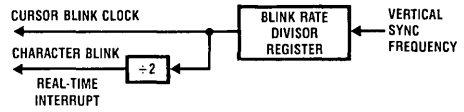
FIGURE 10. CPU Clock Generation

The choice is software programmable through bit 0 in the System Control Register. The exact selection is made in consideration of the fact that the CPU clock must lie within the range of 3 to 12 MHz. In addition, the choice of divide by modes will also impact the display character cell width due to the nature of the video controller. Specifically with ÷1.5

2.0 Functional Description (Continued)

the cell width must be ≥ 8 dots wide whereas with ÷1 the cell width must be ≥ 6 dots wide.

The low clock rates necessary to implement Cursor Blinking, Character Blinking and the Real-Time Interrupt are derived by passing the vertical sync frequency through a 5-bit Blink Rate Divisor Register, *(Figure 11)*. The resultant frequency is used as the Cursor Blink Clock. This clock is then further divided by 2 to yield the Character Blink and Real-Time Interrupt Clocks. For example, to get a 1 Hz real time interrupt, with a 60 Hz system, set the 5 bit Divisor Register to 30 in order to yield a 2 Hz signal which is then divided by 2.

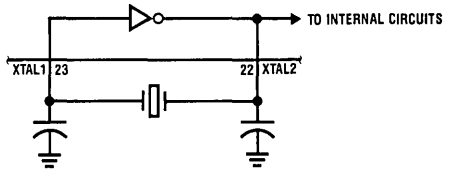


TL/DD/5526–18

FIGURE 11. Blink Clock Generation

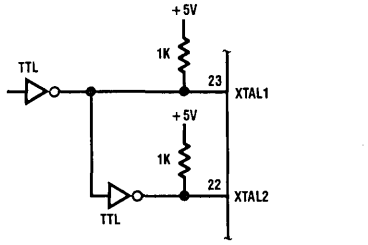
2.1.10 Oscillator Operation

The on-board oscillator circuit consists of a phase inverter which, when used with an external parallel resonant tank, *(Figure 12a)*, will yield the required oscillator clock. Crystals should be specified for AT cut and parallel resonant operation with the desired load capacitance (typically 20 pF). If one desires to externally generate the clock and input it to the chip, he may do so by driving XTAL1 (pin 23) and XTAL2 (pin 22) as shown in *Figure 12b*.



TL/DD/5526–19

FIGURE 12a. TMP Oscillator



TL/DD/5526–20

Note: Use AS TTL devices if faster than 12 MHz.

FIGURE 12b. External Oscillator Mode

2.2 DISPLAY MEMORY CONTROLLER

The video display data resides in the external Video Memory which is managed by the Display Memory Controller (DMC) through the System Bus. Either the CPU or the Video Controller may access the display memory by presenting its requests to the DMC. A maximum of three Video Memory accesses (Reads or Writes) can be performed by the DMC during each CPU instruction execution cycle. Because the CPU can access the Video Memory, one may expand CPU I/O or data memory by memory mapping into the Video Memory space. Up to 64k locations may be addressed ove the 16-bit System Bus. Data word widths may be 8 or 16 bit depending upon whether external character attribute selec tion is used. The actual bus multiplexing mode is controlle by bit 4 in the System Control register. The Video Controlle has the highest priority in obtaining Video Memory accesse with the CPU getting in on a space available basis. If a memory accesses are being taken by the Video Controlle (rarely), the CPU is put into a wait state should it try to ac cess video memory. To ease accessing requirements an boost throughput the Video Controller utilizes a 4-level dat FIFO which is normally kept full of display data.

2.2.1 Display Memory Control Registers

In order to facilitate the management of video data for suc features as a Screen scroll, memory paging and row looku the DMC utilizes a number of registers which address th video RAM space. Each of these pointers is 16 bits wid and writable or readable from the 16-bit HACC/ACC pair a the case may be. There are 2 video data accessing mode as determined by bit 5 in the SCR, Sequential and Tabl Lookup. The functions of the pointer registers vary depend ing upon the accessing mode selected. Their designator are:

HOME = Home address register. Read and write.

BEGD = Beginning of diplay RAM. Write only.

ENDD = End of display RAM. Write only.

CURS = Cursor address register. Read, Write, Increment Decrement.

SROW = Status section register. Write only.

CRSR = Current row start register. Not directly accessed

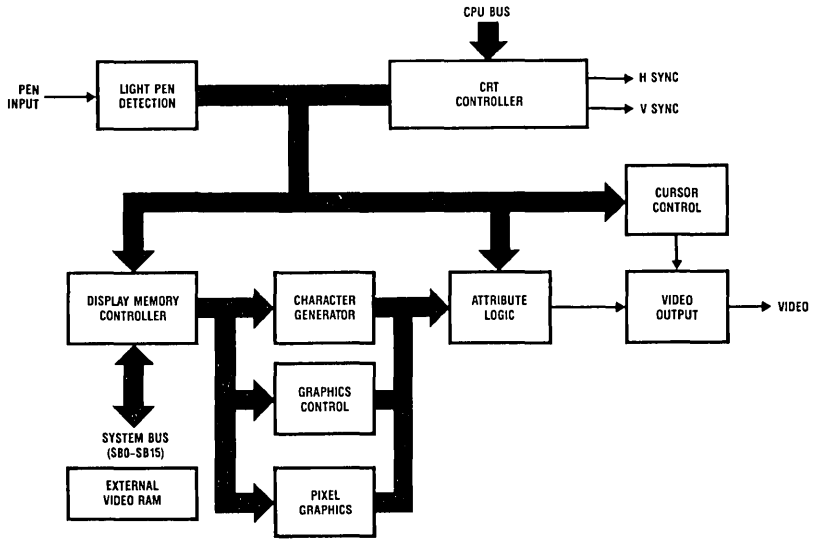
2.2.2 Sequential Access Mode

In this mode display data is accessed from sequential ad dress locations in the video memory until the data require ments for the current screen field are fulfilled. The locatior from which the first display character is taken is the one pointed to by the HOME register. By modifying the contents of HOME one may implement a row scroll or paging opera tion. The BEGD and ENDD are used to control the wrap around condition when HOME gets near the end of avail able display RAM as determined by ENDD. In this instance when sequential accessing brings us to the end of memory as pointed to by ENDD, the controller wraps around by jumping back to the beginning of display memory as pointec to by BEGD. The value in ENDD should be the last location in display memory + 1. Also the size of the display memory between BEGD and ENDD (ENDD − BEGD) must be an integral number of display rows. The CURS in both access ing modes merely identifies the current cursor position in display memory so that the cursor characteristics can be inserted into the video at the appropriate character position.

In addition to the display of normal video data one may elect to have a special status section displayed using data from a separate section of video memory. The status section would consist of an integral number of display rows on the bottom of the screen. This feature operates by reloading the video RAM pointer with the contents of SROW when the desired row position at which to start the status section comes up. The particular row at which the status display starts is defined in the Timing Chain. Once the video RAM pointer is jumped to SROW, data accessing again proceeds sequentially from there until the data requirements for the current field are satisfied.

2.0 Functional Description (Continued)

TMP Video Section



TL/DD/5526-21

Whether a status section is used or not, upon accessing all of the data necessary to display a field, the video RAM pointer is reset to HOME in preparation for the display of a new field.

2.2.3 Table Lookup Mode

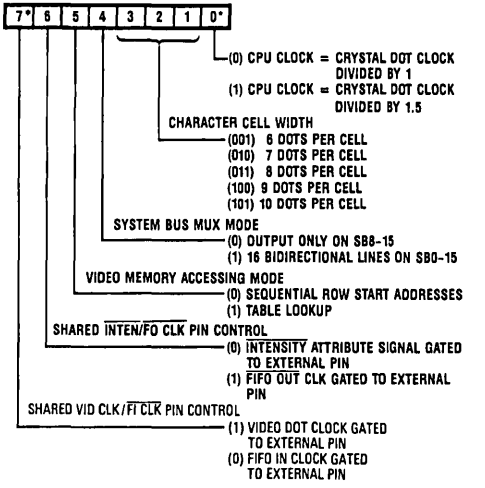
The CRSR (transparent to the user) is a pointer to the address of the first character in a display row. It is required because each time a scan line is displayed, all display characters in the row must be accessed anew. Since a row is made up of a number of scan lines, we must recover the address of the first character in the row for each scan in the row. After a row is done, the CRSR is normally advanced to point to the first character in the next row.

In table look-up mode the starting memory location of the next row is loaded into the CRSR from the HOME register at the end of each row. The HOME register was presumably updated by the CPU since the last end of row.

A CRSR load also generates the internal End of Row interrupt which the CPU will use as a signal to reload HOME. Finally, reloading HOME will clear out the End of Row interrupt. If the status section feature is used, upon reaching the begin status row location the CRSR will be loaded with SROW instead of HOME for that row. After which CRSR will revert back to load from HOME for the remaining rows on the screen.

2.3 SYSTEM CONTROL REGISTER

Through the System Control Register (SCR) the user specifies several important chip operational conditions. It is an 8-bit write only register which is loaded from the CPU accumulator.



TL/DD/5526-22

*Bit 0 is set to 1 by RESET and bit 7 is set to 0 by RESET.

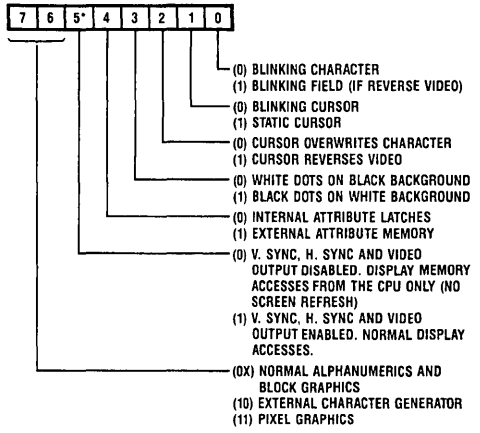
FIGURE 13. System Control Register

2.4 VIDEO CONTROL REGISTER

Through the Video Control Register (VCR) the user specifies several video display features to the chip. It is an 8-bit write only register which is loaded from the CPU accumulator.

7

2.0 Functional Description (Continued)



TL/DD/5526–23

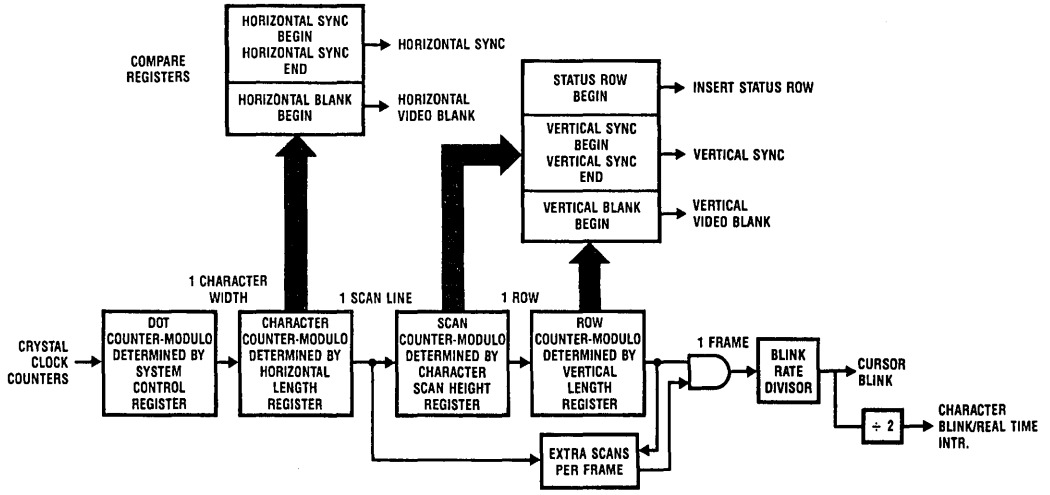
*Bit 5 is set to 0 by RESET.

FIGURE 14. Video Control Register

2.5 CRT REFRESH LOGIC

All video timing and clocking signals are derived from a se ries of counters and comparators called the Video Timing Chain. The chain is driven by the dot/crystal clock and ulti mately divides down to the very slow blink clock, *(Figure 15)*. By having the program initialize the registers in the chain a user may specify all aspects of video generation.

The chain also controls the size and placement of the cur sor and underline attribute within a character cell as well as the cell partitioning for block graphics display. All totaled the chain consists of 14 wire only registers. They are loaded indirectly by using the Timing Chain Pointer (TCP), a 4-bit pointer to registers in the chain, and the MOV @TCP, *R* instruction.



TL/DD/5526–24

FIGURE 15. TMP Video Timing Chain

2.5.1 TMP Timing Chain Registers

TCP	Horizontal Timing
0	Horizontal Length Register — HLR 7 bits — Total number of character cells in a horizontal scan and retrace. — Enter desired count − 1
1	Horizontal Blank Begin Register — HBR 7 bits (Characters/Row) — Character position in horizontal scan after which horizontal blanking begins. — Enter desired number of displayed characters/row − 1.
2	Horizontal Sync Begin Register — HSBR 7 bits — Character position in horizontal scan after which horizontal sync begins (rising edge), HSBR ≤ HLR. — Enter desired count + 2.

2.0 Functional Description (Continued)

2.5.1 TMP Timing Chain Registers (Continued)

TCP — **Horizontal Timing**

3 Horizontal Sync End Register — HSER 7 bits
- — Character position in horizontal scan after which horizontal sync ends (falling edge), HSER ≤ HLR.
- — Enter desired count + 2.

Note: The polarity of the horizontal sync signal can be inverted by switching the values in the two horizontal sync registers.

TCP — **Character Height Definition**

4 High Nibble — Character Scan Height Register — CSHR 4 bits (see *Figure 16a*)
- — Scan line height of a character cell.
- — Enter desired number of scan lines − 1.

4 Low Nibble — Extra Scans/Frame — ES/F 4 bits
- — Number of extra scans to be added to a frame if desired.
- — Enter desired number of extra scans −1.
- — To get no extra scans make ES/F = CSHR. ES/F must be ≤ CSHR.

TCP — **Vertical Timing**

5 Vertical Length Register — VLR 5 bits
- — Total number of display and retrace rows in a frame.
- — Enter desired number of rows − 1.

6 Vertical Blank Register — VBR 5 bits (Rows/Screen)
- — Row position in vertical scan after which vertical blanking begins, VBR < VLR.
- — Enter desired number of displayed rows − 1.

7 High Nibble — Vertical Sync Begin Register — VSBR 4 bits
- — Scan line position in first blank row at which vertical sync begins (falling edge). Sync starts 1 char time after blanking for that line starts (except when VSBR = CSHR sync will start 1 char time after blanking of the last displayed scan line).
- — Enter desired scan line position − 1.

7 Low Nibble — Vertical Sync End Register — VSER 4 bits
- — Scan line position after start of vertical sync at which vertical sync ends (rising edge). Sync ends 1 char time after horizontal blanking for that scan line start.
- — Enter desired scan line position − 1.

Note: If VSER = VSBR there will be no vertical sync signal.

8 Status Row Begin Register — SRBR 5 bits
- — Row count after which the status row is inserted.
- — Enter desired row position − 1.

TCP — **Cursor and Graphics Control**

9 Upper 5 Bits — Blink Rate 5 bits
- — Divider driven by the vertical sync frequency to yield the slow cursor, character and real-time blink rates.
- — Enter desired divisor − 1.

9 Lower 3 Bits — Blink Duty Cycle 3 bits
- — Approximate ON time of blink signal.
- — 000 = shortest, 111 = longest (100 = 50% duty cycle).

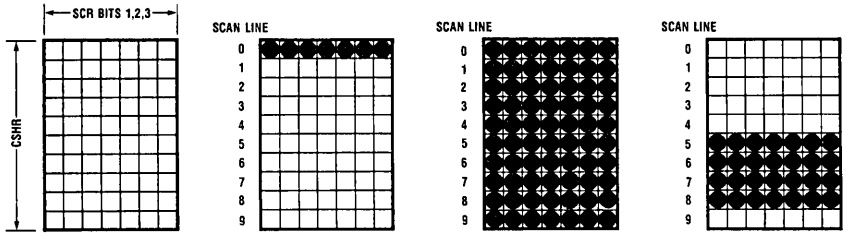
10 Graphics Column Register — GCR 8 bits
- — Assign dot positions to left, middle and right character cell columns for block graphics operation.

11 Graphics Row Register — GRR 8 bits
- — Defines scan count at which middle row for block graphics characters begins (upper nibble) and at which bottom row begins (lower nibble). The middle row (upper nibble) must be ≥ 1.
- — Enter desired scan count − 1.

12 Underline Size Register — USR 8 bits (see *Figures 16a, b, c*)
- — Defines the beginning (upper nibble) and ending (lower nibble) scan lines for the underline attribute. Values must be ≤ CSHR.

13 Cursor Size Register — CSR 8 bits (see *Figures 16a, b, c*)
- — Defines the beginning (upper nibble) and ending (lower nibble) scan lines for the cursor. Values must be ≤ CSHR.

2.0 Functional Description (Continued)



a. Character Cell Format Specification | **b. Underline Size Register = 90** | **c. Cursor Size Register = 9A (0B, 0C, 0D, 0E, 0F) may also be used** | **d. Cursor Size Register = 48**

TL/DD/5526-2

FIGURE 16. Underline and Cursor Register Operation

Note: The internal cursor flip-flop gets set to ON whenever a scan line corresponding to the begin cursor nibble is reached, and gets set to cursor OFF whenever scan line corresponding to the end cursor nibble is reached. The cursor attributes are inserted whenever the character position being displayed corresponds to th one pointed to by the cursor address register. A similar situation applies for characters with the underline attribute selected. Therefore, care should be taken whe setting the ES/F register and setting the cursor and underline sizes. In particular the ES/F value should not be between the upper nibble and lower nibble values c the underline size register or between the upper nibble and lower nibble values of the cursor size register. To use the cursor as a pointer without displaying it, se the lower nibble of the cursor size register to a value less than CSHR and the upper nibble to a value greater than CSHR.

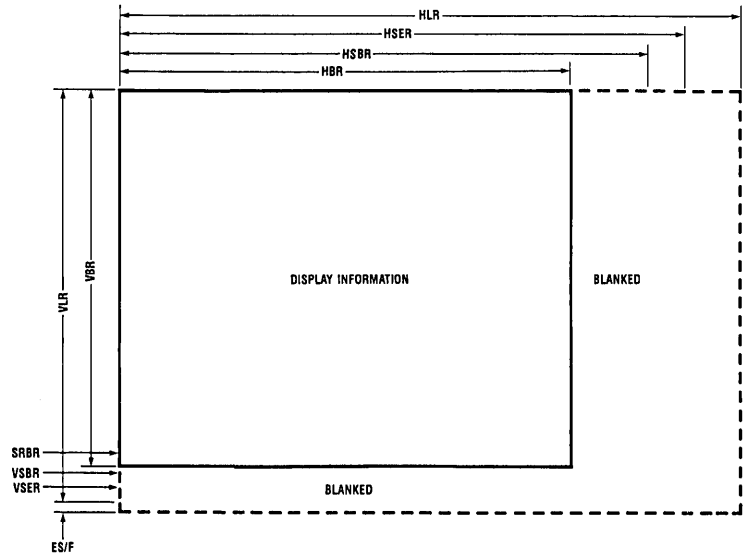
2.5.2 TIMING CHAIN LOAD VALUE EXAMPLE

It is desired to have a display field of 80 columns by 25 rows with the last screen row being a status row. It has been determined that 25 character width times will be necessary to complete horizontal retrace and that Horizontal sync should be positioned to start a full seven character times after blanking and end twenty characters after blanking to give us a total sync width of 13 character times. (See *Figure 17* for example.)

Additionally, vertical retrace will take 23 scan line times to complete with vertical sync starting three scan line times after vertical blanking begins and occupying a total period of 11 scan lines.

It is desired to make the character cells 12 scan lines tall. The cursor will be a block shape and occupy the bottom 11 scan lines in a cell. The underline attribute will actually be a strike through dash occupying the 4th scan line from the to in a cell.

Our line width is 80 displayed characters plus 25 for retrac making HLR = 80 + 25 − 1 = 104. Blanking will start afte the 80th character so HBR = 80 − 1 = 79. To achiev seven character times after horizontal blanking, HSBR = 87 + 2 = 89. To achieve twenty character times afte blanking HSER = 100 + 2 = 102 (note 102 − 89 = 1 total). Cell height is 12 lines so CSHR = 12 − 1 = 11 Since there are 12 scan lines per cell or row, vertical retrac will require 23/12 = 1 row and 11 scan lines. This make our total row count VLR = 25 + 1 − 1 = 25 and ES/F = 11 − 1 = 10. Thus, timing chain location 4 would be coded 1011 1010. We will display 25 rows so VBR = 25 − 1 = 24. Vertical sync will start at the beginning of the fourth scar



TL/DD/5526-26

FIGURE 17. Typical Video Screen Format Specification

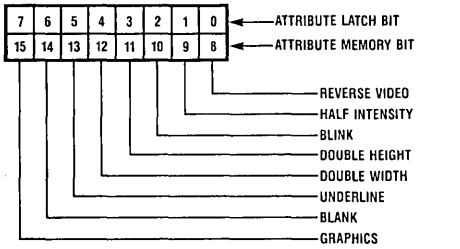
2.0 Functional Description (Continued)

line of the row after blanking begins so VSBR = 4 − 1 = 3. It will run for 11 scan lines or specifically the 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2 ending at the beginning of the 3rd so VSER = 3 − 1 = 2. The status row will be after the 24th so SRBR = 24 − 1 = 23. To specify the underline and cursor sizes one must remember that the first scan line is numbered 0. To get our 11 line block cursor we begin after the 0 line and end at the end of the 11 line making CSR = 0000 1011. The underline dash will be USR = 0011 0100. Note that the CSHR determines the scan counter modulo and if a scan compare register value (ES/F, VSBR, VSER, USR, CSR) is never reached, the signal end or begin will never be initiated.

2.6 ATTRIBUTES

Eight independent attributes may be inserted itno the video dot stream to affect display characters on either an individual or global basis. The eight attributes along with their control word bit assignments are detailed in *Figure 18*. The scope with which a particular set of attributes affects the display depends upon whether attribute control is internal or external as determined by bit 4 in the VCR.

Attributes are present if the corresponding bit is a ZERO (low).



TL/DD/5526-27

FIGURE 18. Attribute Bit Assignments

2.6.1 Internal Attribute Selection

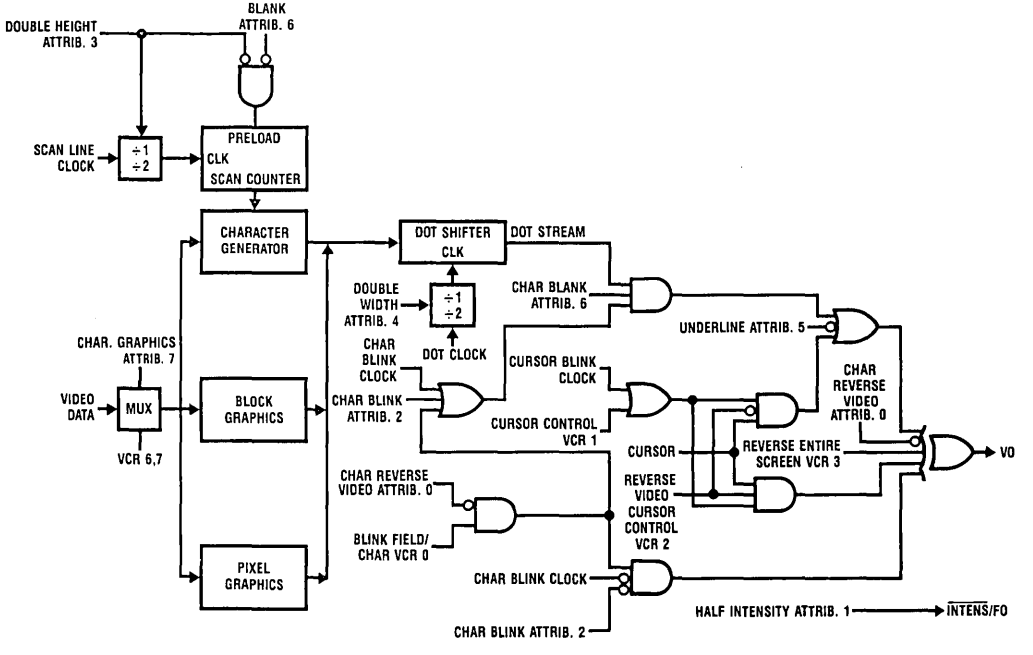
In internal mode attribute control comes from one of two internal attribute latches designated AL0 and AL1, either of which is directly loadable from the CPU accumulator. The choice of which of the two is used for a particular display character is determined by bit 7 (MSB) in the display memory data byte with 0 = AL0 and 1 = AL1. (Characters are represented in display memory as ASCII values occupying the low 7 bits of each 8-bit byte thus leaving bit 7 free for attribute control.)

2.6.2 External Attribute Selection

In external mode each display character has associated with it, a dedicated attribute field in the form of a high 8-bit extension to the regular display memory character byte. To use this mode the system bus msut be configured for 16-bit bidirectional operation (SCR bit 4 = 1) and external attributes must be selected (VCR bit 4 = 1).

2.6.3 Attribute Processing

Each of the eight attributes may be independently enabled thus yielding a number of possible combinations. The exact processing involved is shown in *Figure 19*. Note that attributes are always present. Whether any of them are active depends upon the particular control bit being enabled in the latch or memory.



TL/DD/5526-28

FIGURE 19. TMP Attribute Processing

2.0 Functional Description (Continued)

2.6.4 Attribute Operation

Reverse Video: A character and its surrounding cell are reversed in video from what was selected for the rest of the screen.

Half Intensity: To use the half intensity function the shared $\overline{\text{INTENSITY/FO CLK}}$ pin (25) must be selected for INTENSITY operation by setting SCR bit 6 low. In operation the half intensity pin will be low whenever a character for which the attribute is active is being displayed. To perform the actual attenuation function external circuitry must be connected between the $\overline{\text{INTEN}}$ and Video Output pins. In fact the signal may be used for another purpose such as switching between two colors.

Blink: A character or the field around it blinks as selected by VCR bit 0.

Double Height: A designated character is stretched out so that it will occupy a 2-row tall space. This attribute is implemented by slowing down by half the scan line stepping to the internal character generator. To use this attribute the desired double high character must be placed into the two display memory locations corresponding to the top and bottom row positions. For both locations the double high attribute is set. In addition the Blank attribute for the bottom character is also set to tell the controller it is the bottom half of a double high character. The double high attribute has no effect on element graphics or on pixel graphics displays. If an external character generator is used special circuitry must be employed to implement double high characters.

Double Width: A designated character is stretched out so that it will occupy a 2-character cell wide space. This attribute is implemented by slowing down by half the clock to the video dot shifter. To use this attribute the desired double wide character must be placed in the left character position and the double wide attribute bit set. The following character position (right) can have any character as it will be ignored.

Underline: If set this attribute causes the underline figure to be added to the video dot stream. Since the underline, like the cursor, can be specified as to position and size in the character cell, the underline can be an overline, block, strike through or any one of a number of effects. The underline overwrites any dot where it overlaps the character.

Blank/Double High Bottom: A character is inhibited from being displayed while still allowing it to be stored in the display memory. If this attribute and the double height attribute are set for the same character, the normal blank function is disabled for that character position and the character is displayed as the bottom half of a double height character.

Graphics: This attribute determines whether the video memory data byte as accessed by the display memory controller is routed through the character generator or block graphics control logic. If routed through the block graphics logic (attribute active) the effect on the video display will be as described in the Block Graphics section. Note that because Block Graphics mode is selected as an attribute it may be mixed in with normal alphanumerics characters. Also all other attributes with the exception of double height operate on the block graphics characters.

2.7 CHARACTER GENERATOR

The internal character generator holds 128 characters in a 7 x 11 matrix. The standard character sets are addressed using 7-bit ASCII codes stored in the display memory. When operating with fonts smaller than the maximum of 7 x 11, zeroes are encoded into the unused bits. When putting out a character the video controller always starts character generation on the second scan line of a row, leaving the first scan line blank. Similarly, the first (left) column in a character cell is blanked with character generation starting on the second column. Therefore, the specified cell size must be one greater in height and width than the display characters (including descenders) otherwise they will be chopped off. If the character cells are larger than the internal 7 x 11 matrix, blank dots will be put out after exhausting the internal generator (See *Figure 20* for example.)

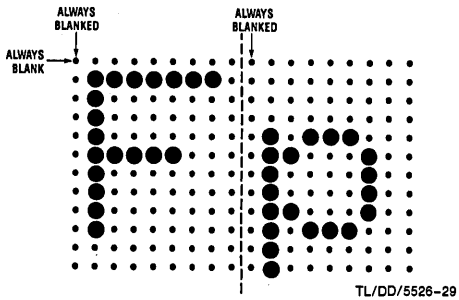


FIGURE 20. Character Cell Format

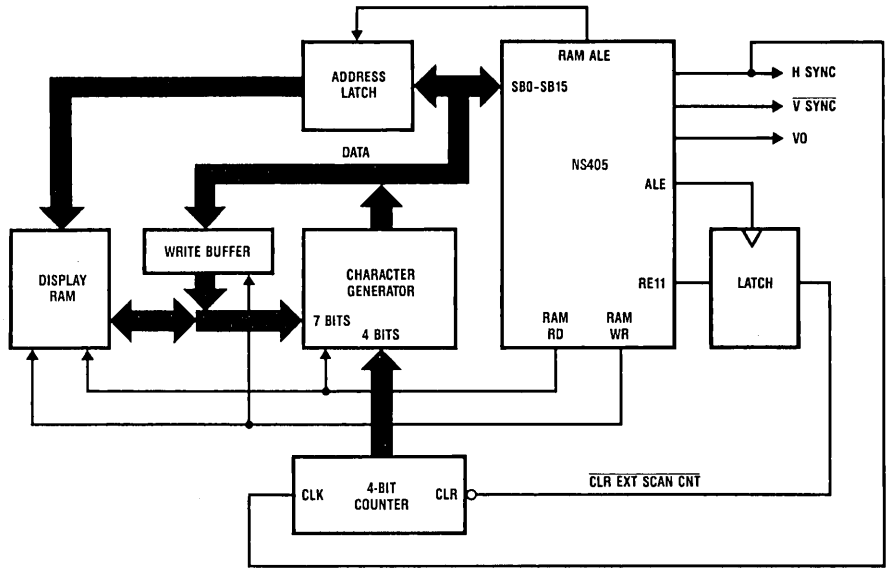
2.7.1 External Character Generation

The chip may be used with an external character generator by switching over to a pixel graphic display mode with modified address stepping as controlled by VCR bits 6, 7. In this mode an external character generator supplies pixel data to the chip as depicted in *Figure 21*. Character addressing comes from the display memory and scan line stepping from a 4-bit counter clocked by the Horizontal Sync. Scan line synchronization is achieved by using the Scan Count Clear signal coming out on RE11, pin 36. After the display of a row it pulses low to initialize the scan line counter for the start of a new row. In pixel mode both the character and any spacing between characters must be encoded into the external character generator. In addition, the chip will access and use at most 8 bits of pixel data for each character cell. However, if the cell width is specified to be 9 or 10, the ninth and tenth dots will repeat what was coded into the first. Therefore, assuming at least one dot spacing between characters, external fonts can at most be seven dots wide.

No limitations apply to the height of a character as long as the external generator can supply all of the scan lines as specified by the CSHR. As in regular pixel mode the LSB brought in is the first dot put out.

Since the eighth data bit is used for character generation it cannot effectively be used for internal attribute latch selection although one of the latches will be selected every data byte. Therefore, both internal attribute latches must be loaded with the same values. If external attribute operation is specified the full 8-bit high order attribute field is available for usage.

2.0 Functional Description (Continued)

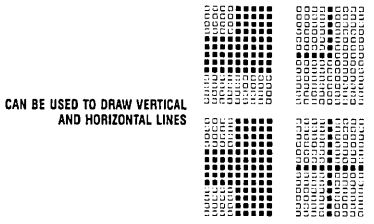


TL/DD/5526-30

FIGURE 21. External Character Set Implementation

2.8 BLOCK GRAPHICS

Block graphics is an alternative display mode to normal alphanumerics which is selected through attribute bit 7. Example *(Figure 22)*. It can operate on a character cell by character cell basis (see Attributes) and words by rerouting display memory bytes through the Block graphics logic instead of the internal character generator.



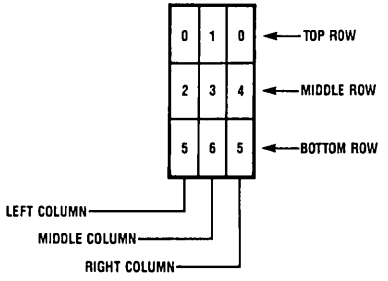
TL/DD/5526-31

FIGURE 22. Example Block Graphics Display Patterns

The Graphics Logic operates by partitioning the character cell space into nine possible areas as shown in *Figure 23* and then using the seven lower bits in the display data byte to turn these areas on or off. In this way one can draw contiguous lines or simple geometric figures while at the same time displaying alphanumeric characters in other cells.

The partitioning of the cell is controlled by two timing chain registers which specify two Horizontal and two Vertical cut off points to the graphics logic. Through these two registers one can make the sections as large or as small as desired, even eliminating sections entirely. Note that data bits 0 and 5 each control two sections as depicted in *Figure 23*.

2.8.1 Graphics Partitioning



TL/DD/5526-32

FIGURE 23. Block Graphics Cell Partitioning

The registers defining the graphics areas function as follows:

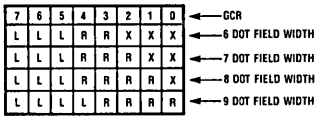
The Graphics Row Register — 8 bits (GRR) is divided into the following two (2) registers:

- Graphics Middle Row, (GMR):
  Defines the scan count at which the middle row begins (4 most significant bits of GRR).
- Graphics Bottom Row, (GBR):
  Defines the scan count at which the bottom row begins (4 least significant bits of GRR).

See *Figure 24.1a* for row example.

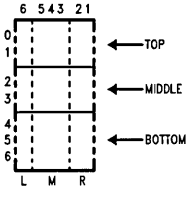
2.0 Functional Description (Continued)

The Graphics Column Register — 8 bits (GCR) controls vertical partitioning through bit patterns as follows: (See *Figure 24.*)



TL/DD/5526–33

FIGURE 24. Block Graphics Column Partitioning



TL/DD/5526–44

GRR = 24
GCR = 60 (0110 0XXX)
cell size = 6 x 7

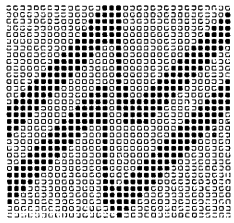
FIGURE 24.1a Block Graphics Example

For all bits in the Graphics Column Register, a one assigns that bit position to the middle column. A zero in an L bit position assigns that bit position to the left column. A zero in an R bit position assigns that bit position to the right column. There is always at least one middle dot although the left and right sections may be eliminated entirely. For 10 dot wide cells the 10th bit will repeat the 9th bit. An easy way to determine the column partitioning is to fill the GCR with all ones, thereby making it one large middle section. Then, starting from the outermost L and R bit positions, put zeros in until the left and right sections are the sizes needed.

2.9 PIXEL GRAPHICS

When bits 6 and 7 of the Video Control Register are both set to 1, the character generator and block graphics circuits are disabled. Video output directly reflects the contents of the display memory byte on a pixel (dot) per bit basis with data output LSB first. Example *(Figure 25).*

Nine bits at a time are accessed from each video memory location with as many bits being used as defined in the character cell width specification. If a cell width of 10 is specified



TL/DD/5526–34

FIGURE 25. Example Pixel Graphics

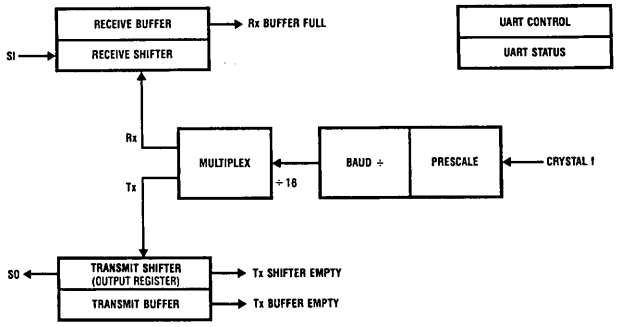
the 10th bit will merely repeat the 9th bit. Attributes are still operable in pixel mode, on a data byte basis, with internal and external operation possible. With internal attribute latch operation the same values must be loaded into both latches since the usual latch select bit is now being used for pixel control. Unless, however, only a 7 dot wide cell is used leaving the 8th bit free. With external attribute operation we are now limited to a 7-bit attribute field since pixel data can now occupy 9 of the 16 bus bits. Because of this the LSB attribute, Reverse Video is totally disabled from operation in Pixel Graphic mode. This also applies to internal attribute latch operation. Note, however, that reverse entire screen video is still operable. Address sequencing through the video memory is sequential with as many data bytes being read in as is necessary to satisfy the pixel requirements of the screen.

2.10 LIGHT PEN

Activation of the light pen interrupt causes the horizontal and vertical screen position of the currently displayed character to be latched into the Horizontal Light Pen Register HPEN (7 bits) and Vertical Light Pen Register VPEN (5 bits) respectively. Both HPEN and VPEN may be read into the CPU accumulator. The values latched remain in VPEN and HPEN until another light pen interrupt latches new values.

2.11 UART

The UART features full duplex operation with double buffered Receive and Transmit sections. Baud rate generation is fully programmable through a 2-stage divider chain. CPU control of the UART is extensive with polled or interrupt driven operation possible.



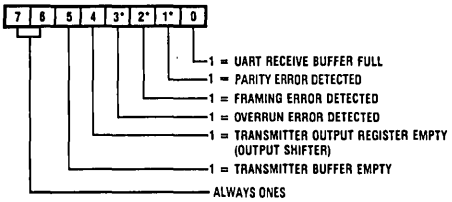
TL/DD/5526–35

FIGURE 26. TMP UART Block Diagram

2.0 Functional Description (Continued)

2.11.1 UART Control

UART Status Register (STAT): Contains error and status bits which reflect the internal state of the UART. Read into CPU accumulator. Bits 0, 5 are the same as those found in the internal interrupt register.



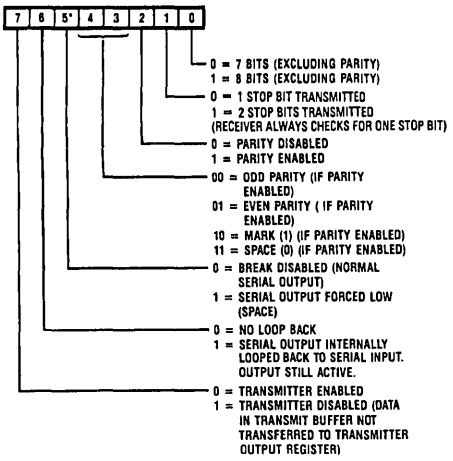
TL/DD/5526-36

UART Status Register bits 1, 2, 3 are only cleared on a chip reset or a read of the UART Receive Buffer. If another word were to come in before the Receive Buffer could be read the errors associated with the new word would add to those already present. The receipt of a new word can cause the three bits to go from a 0 to a 1, but not from a 1 to a 0.

FIGURE 27. UART Status Register

Note: The Transmit Output Register Empty flag is set to one whenever the transmitter is idle. The flag is reset to zero when a data character is transferred from the Transmit Buffer to the Output Register. This transfer does not occur until the next rising edge of the internal UART Transmit Clock. The Transmitter Output Register Empty flag occurs at the beginning of the last stop bit.

UART Control Register (UCR): Contains control bits which configure the format of transmitted data and tests made upon received data. Written to from CPU accumulator.



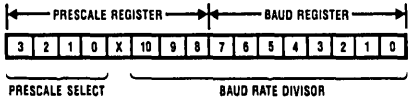
TL/DD/5526-37

*Bit 5 set to 0 by RESET.

FIGURE 28. UART Control Register

2.11.2 Baud Clock Generation

The basic BAUD clock is derived from the crystal frequency through a two-stage divider chain consisting of a 3.5–11 prescale and an 11-bit binary counter. *(Figure 29).* The divide factors are specified through 2 write only registers shown in *Figure 30.* Note that the 11-bit Baud Rate Divisor spills over into the Prescale Select Register. The correspondences between the 4-bit Prescale Select and Prescale factors is shown in Table I. There are many ways to calculate the two divisor factors but one particularly effective method would be to try to achieve a 1.8432 MHz frequency coming out of the first stage then use the BAUD Rate Divisor factors shown in Table II.



TL/DD/5526-39

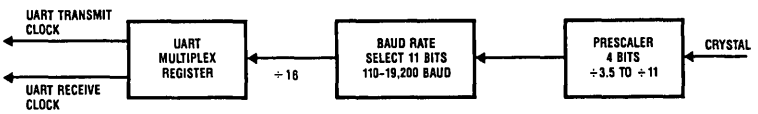
FIGURE 30. UART BAUD Clock Divider Registers

TABLE I. Prescale Factors

Prescale Select	Prescale Factor
0000	3.5
0001	4
0010	4.5
0011	5
0100	5.5
0101	6
0110	6.5
0111	7
1000	7.5
1001	8
1010	8.5
1011	9
1100	9.5
1101	10
1110	10.5
1111	11

TABLE II. Baud Rate Divisors (1.8432 MHz Input)

Baud Rate	Baud Rate Divisor (N − 1)
110 (110.03)	1046
134.5 (134.58)	855
150	767
300	383
600	191
1200	95
1800	63
2400	47
3600	31
4800	23
7200	15
9600	11
19200	5



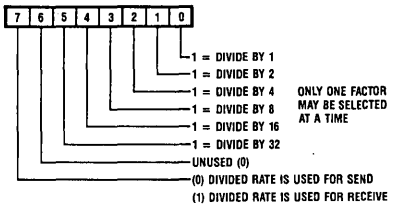
TL/DD/5526-38

FIGURE 29. UART BAUD Clock Generation

2.0 Functional Description (Continued)

The frequency coming out of the BAUD Rate Divisor is then passed through the UART Multiplex Register. Through the UART Multiplex Register one can specify that the Transmitter or Receiver clock be the same or a power of two multiple of the other.

UART Multiplex Register (UMX): Contains the bits which determine the divisor which is used to count down from the primary baud rate when different rates are used for send and receive (eight bits).



TL/DD/5526-40

FIGURE 31. UART Multiplex Register

The actual baud rate may be found from:

BR = Fc/(16*N*P*D)

Where:

BR is the Baud Rate

Fc is the external crystal frequency

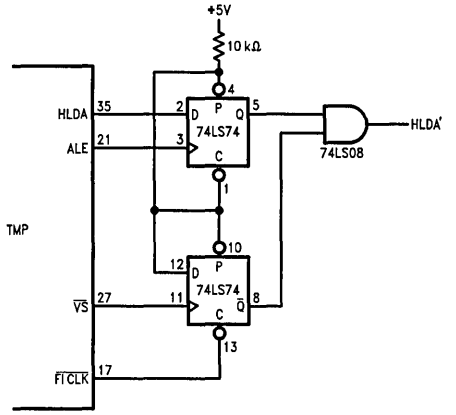
N is one plus the value of the Baud Rate Divisor contained in the Baud Rate Select Register and the Prescale Select Register.

P is the Prescale Divide Factor Selected by the value in the Prescale Select Register.

D is the Multiplex Register Divide Factor

3.0 Slave Processing

The TMP may be used as a slave video controller by having a host system perform Direct Memory Accesses into the display RAM. To assist in implementing such a system the chip features two DMA control pins—HOLD (Hold Request) and HLDA (Hold Acknowledge). These two signals come out on shared ROM Expand Bus pins RE8 and RE12. To request a DMA access a host would activate HOLD (active high and await the acknowledging HLDA from the TMP before proceeding with the DMA. The TMP only allows DMA operations during the vertical blanking period and will activate HLDA in response to a HOLD shortly after vertical blanking starts. In DMA mode all 16 TMP System Bus drivers are tri-stated while the bus control signals RAM ALE, RAM RD, RAM WR go to their inactive (high) states. A HOLD request must arrive two CPU cycles before vertical blanking starts; otherwise it will miss that retrace cycle and will have to wait until the next one, one frame later. Once DMA mode is entered, it is maintained for the duration of vertical blanking regardless of the state of HOLD. Near the end of vertical blanking the DMA mode will terminate in preparation for the display of the next frame, but the HLDA will NOT turn off. Specifically, this will occur one scan time before the end of vertical blanking. It is up to the designer to be sure that the host is off the BUS before this happens or suffer bus contention with the video controller. He can do this by either predetermining the length of time the host has to remain on the bus, or by using the end of vertical sync (as shown in *Figure 32*) to signal the end of a safe DMA period. If during DMA the CPU attempts to do a display memory access it would be put into a wait state until DMA is concluded and normal memory accessing is resumed.



TL/DD/5526-45

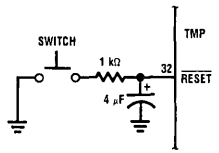
Vertical sync should be programmed to end as late as possible, but must end at least one scan time before the end of vertical blanking.

FIGURE 32

4.0 Reset

The TMP will reset if the $\overline{\text{RESET}}$ (32) pin is held at a logic low (< 0.8V) for at least five CPU cycle times. This pre-supposes that the V_{CC} is up, stable and within operational limits (+5V ±10%) and that the oscillator is running. For a power on reset, time must be allowed for the power supplies to stabilize (typically 50 ms) and the oscillator to start up. If power supply noise or ripple causes V_{CC} to exceed the +5V ±10% limits neither reset nor operation is guaranteed.

Internally, the $\overline{\text{RESET}}$ pin has a depletion load pullup that typically acts as a 30 μA current source from V_{CC} in the voltage range of interest. A typical reset circuit with a 0.5 second reset pulse is shown in *Figure 33*.



TL/DD/5526-41

FIGURE 33. Typical Reset Circuit

4.0 Reset (Continued)

During RESET a number of internal registers are initialized as follows:

4.1 CPU

CPU Clock divide = 1.5 (SCR bit 0 = 1)
Shared VIDCLK/$\overline{\text{FI CLK}}$ = 0 (SCR bit 7 = 0, $\overline{\text{FI CLK}}$ gated to external pin)
Program Counter = 0
Stack Pointer = 0
Program Memory Bank = 0
RAM Register Bank = 0
Timer Stopped
Instruction Register cleared
F0 and F1 cleared

4.2 INTERRUPTS

internal and External Interrupts disabled
Internal Interrupt Register set to 000011X0

4.3 UART

Receiver initialized to look for start bit
Status Register set to 11110000
Transmitter initialized to wait for OUT XMTR instruction
Control Register bit 5 =0 (No BREAK)

4.4 VIDEO

Video generation shutdown (VCR bit 5 = 0)
FIFO Cleared Out
Timing Chain Character Counter = 0
Timing Chain Scan Counter = 0
Timing Chain Row Counter = 0
Timing Chain Blink Counter = 0
} IN TEST MODE ONLY

4.5 PIN STATES AT RESET

Pins 1–8 (SB0–7)	In TRI-STATE during reset and until either the CPU executes a MOVX instruction or bit 5 of the VCR is set.
Pins 9–16 (SB8–15)	If bit 4 of the SCR is set, SB8–15 will behave like SB0–7. If bit 4 of the SCR is cleared, SB8–15 will act as outputs (any of which may be either high or low). Note that bit 4 of the SCR may be one or zero at power-up.
Pin 17 (VID CLK/$\overline{\text{FI CLK}}$)	High during reset and until bit 5 of the VCR is set.
Pin 18 (RAM ALE)	High during reset and until the CPU executes a MOVX instruction or bit 5 of the VCR is set.
Pin 19 ($\overline{\text{RAM WR}}$)	High during reset and until the CPU executes a MOVX (of the output to display RAM variety) instruction.
Pin 20 ($\overline{\text{RAM RD}}$)	High during reset and until either the CPU executes a MOVX instruction or bit 5 of the VCR is set.
Pin 21 (ALE)	Pulses continuously.
Pin 22 (XTAL 2)	Crystal input or master clock input.
Pin 23 (XTAL 1)	Crystal input.
Pin 24 (Gnd.)	
Pin 25 ($\overline{\text{INTENS}}$/$\overline{\text{FO CLK}}$)	May be either high or low during reset.
Pin 26 (VO)	Low (because of asserted blanking signals) from reset until bit 5 of the VCR is set.
Pin 27 ($\overline{\text{VS}}$)	In TRI-STATE mode upon RESET, enabled when bit 5 of the VCR is set.
Pin 28 (HS)	Low from reset until bit 5 of the VCR is set.
Pin 29 (EA)	Input only. (must be tied HIGH (V_{IH_2}))

7

4.0 Reset (Continued)

Pin 30 ($\overline{PSEN}$)	Active during reset.
Pin 31 ($\overline{RD}$)	High during reset and until an IN PORT instruction is executed.
Pin 32 ($\overline{RESET}$)	Input only.
Pin 33 (SO)	High during reset and until an OUT XMTR instruction is executed.
Pin 34 (SI)	Input only.
Pin 35 (RE12/HLDA)	If HOLD is low: low during reset. If HOLD is high: low at falling edge of ALE and during $\overline{PSEN}$, may be low or high at rising edge of ALE.
Pin 36 (RE11/$\overline{SC\ CLR}$)	If reset asserted: low at falling edge of ALE and during $\overline{PSEN}$, sampled value of internal Scan Count Clear signal is output at rising edge of ALE.
Pin 37 (RE10/$\overline{INTR}$) Pin 38 (RE9/$\overline{LPEN}$) Pin 39 (RE8/HLDR)	If reset asserted: low at falling edge of ALE and during $\overline{PSEN}$. Always in TRI-STATE at rising edge of ALE.
Pins 40–47 (RE0–7; I/O0–7)	If reset asserted: low at falling edge of ALE, in TRI-STATE during $\overline{PSEN}$, and may be either high or low at the rising edge of ALE.
Pin 48 (V_{CC})	

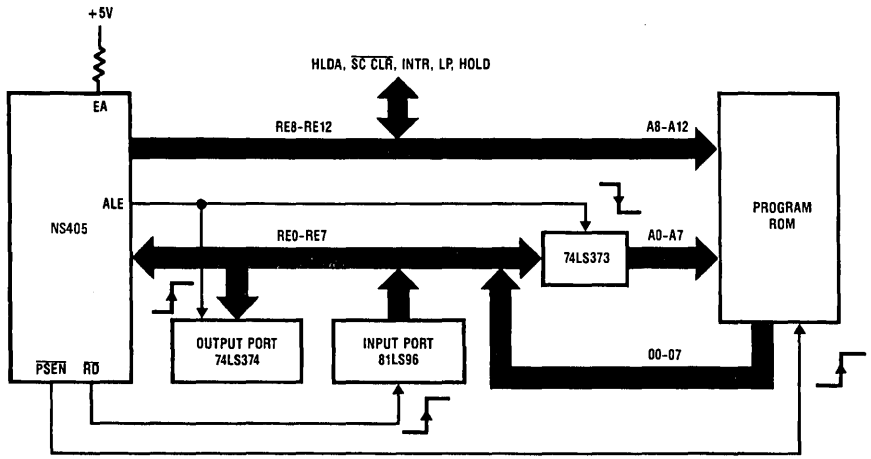
5.0 Extra Attributes

One may want to expand the external attribute field by adding more bits so that functions such as color (Red—Green—Blue drive) or grey scale may be implemented. Like the eight attributes which the chip handles internally these extra attributes would operate on a character cell basis. To add attribute bits one would have to duplicate the internal 4 level character/attribute FIFO externally using fast MSI chips. To assist in handling the external FIFO circuitry the TMP features two FIFO clocking signals on pins 17 and 25. The FIFO IN Clock ($\overline{FI\ CLK}$) is used to strobe attribute data into the external FIFO circuits in synchronism with the internal TMP FIFO. Its timing is identical to $\overline{RAM\ RD}$ but is only active when the video does a display RAM read to load its FIFO. The FIFO OUT Clock ($\overline{FO\ CLK}$) pulses for 1–3 bit times each time the video starts the display of a new character cell. The external FIFO would use the rising edge of this signal to clock out or latch the attribute output.

In order for the TMP CPU to access the additional attribute bits special bus gating arrangements would have to be worked out on the System Bus (Video Data Bus is at most 16 bits wide). Unless one were to run with internal attributes or only use a few of the external attributes in which case the unused bits could be used with the external FIFO. Whenever using the $\overline{FO\ CLK}$ the Intensity attribute is disabled since they both share the same pin.

6.0 TMP BUS Interfacing

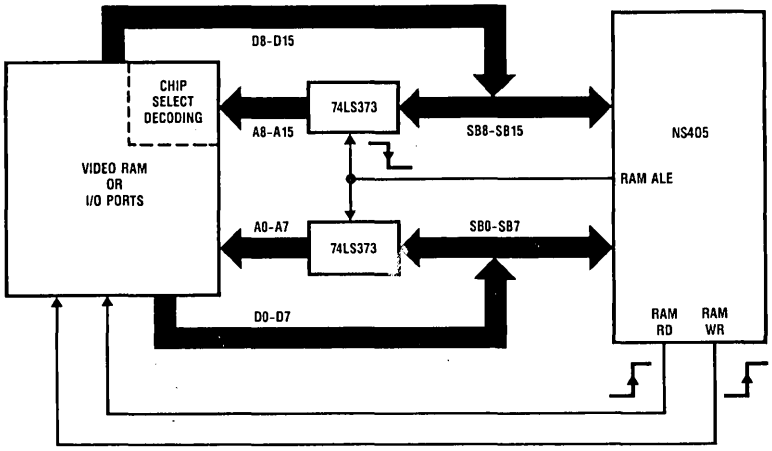
The two external buses on the TMP, ROM Expand and System are easily interfaced to as shown in *Figures 34* and *35*. Important bus information output from the chip is latched using the rising or falling edges of the various control signals. I/O port information is read in through a TRI-STATE® buffer chip such as an 81LS96.



TL/DD/5526–42

FIGURE 34. TMP ROM Expand BUS

6.0 TMP BUS Interfacing (Continued)



TL/DD/5526-43

FIGURE 35. TMP System Bus

TMP Registers (Excluding Timing Chain Registers)

TMP Registers		Associated Intructions			
		CPU SECTION			
A	= Accumulator — 8 bits	ADD A,Rr		MOV A,Rr	XCH A,Rr
#data	= data immediate	ADD A,#data		MOV A,@Rr	XCH A,@Rr
Rr	= Register	ADD A,@Rr		MOV A,#data	XCHD A,@Rr
@Rr	= Register pointed to by R0 or R1	ADDC A, Rr		MOV Rr,A	XRL A,Rr
		ADDC A,#data		MOV Rr,#data	XRL A,@Rr
		ADDC A,@Rr		MOV @Rr,A	XRL A,#data
		ANL A,Rr		MOV @Rr,#data	JBn addr
		ANL A,#data		MOVP A,@A	JNZ addr
		ANL A,@Rr		MOVP3 A,@A	JZ addr
		CLR A		RL A	DJNZ Rr,addr
		CPL A		RLC A	
		DAA		RR A	
		DEC A		RRC A	
		DEC Rr		ORL A,Rr	
		INC A		ORL A,@Rr	
		INC Rr		ORL A,#data	
		INC @Rr		SWAP A	
*HACC	= High Accumulator — 8 bits	*MOV A,HACC		*MOV HACC,A	
C	= Carry Bit	CLR C	CPL C	JNC addr	JC addr
*LONG R0	= Register Pair, R0, RA	*DECL R0		*INCL R0	*MOVL A,R0
		*MOVL R0,A		*MOVX A,@R0	*MOVX @R0,A
*LONG R1	= Register Pair R1, RB	*DECL R1		*INCL R1	*MOVL A,R1
		*MOVL R1,A		*MOVX A,@R1	*MOVX @R1,A
T	= Timer — 8 bits	MOV A,T		MOV T,A	STOP T
		STRT T		*JNTF addr	JTF addr
F0	= Flag 0	CLR F0	CPL F0	JF0 addr	*JNF0 addr
F1	= Flag 1	CLR F1	CPL F1	JF1 addr	*JNF1 addr
INTR	= Interrupt Register — 8 bits	MOV A,INTR		JNXI addr	JXI addr
		*DIS II		DIS XI	*EN II
		EN XI			

7

TMP Registers (Excluding Timing Chain Registers) (Continued)

TMP Registers		Associated Instructions		
CPU SECTION (Continued)				
MASK	= Internal Interrupt MasK — 8 bits	*MOV MASK,A		
PSW	= Program Status Word — 8 bits	MOV A,PSW	MOV PSW,A	
PORT	= 8 bit I/O Port	ANL PORT,#data	IN PORT	
		ORL PORT,#data	OUT PORT	
Miscellaneous Instructions		CALL addr	JMP addr	JMPP @A
		NOP	RET	RETR
		SEL MB0	SEL MB1	*SEL MB2
		*SEL MB3	SEL RB0	SEL RB1
VIDEO MANAGEMENT				
		Associated Instructions		
SCR	= System Control Register — 8 bits	*MOV SCR,A		
VCR	= Video Control Register — 8 bits	*MOV VCR,A		
HOME	= Home Address Register — 16 bits	*MOV A,HOME	*MOV HOME,A	
CURS	= Cursor Address Register — 16 bits	*DEC CURS	*INC CURS	*MOVX A,@CURS
		*MOV CURS,A	*MOV A,CURS	*MOVX @CURS,A
BEGD	= Beginning of Display RAM Register — 16 bits	*MOV BEGD,A		
ENDD	= End of Display RAM Register — 16 bits	*MOV ENDD,A		
SROW	= Status Row Register — 16 bits	*MOV SROW,A		
AL0	= Attribute Latch 0 — 8 bits	*MOV AL0,A		
AL1	= Attribute Latch 1 — 8 bits	*MOV AL1,A		
HPEN	= Horizontal Light Pen Register — 7 bits	*MOV A,HPEN		
VPEN	= Vertical Light Pen Register — 5 bits	*MOV A,VPEN		
VINT	= Vertical Interrupt Register — 5 bits	*MOV VINT,A		
UART CONTROL				
PSR	= Prescale Register (UART) — 8 bits	*MOV PSR,A		
BAUD	= Baud Rate Select Register — 8 bits	*MOV BAUD,A		
UCR	= UART Control Register — 8 bits	*MOV UCR,A		
UMX	= UART Multiplex Register — 8 bits	*MOV UMX,A		
STAT	= Status Latch (UART) — 6 bits	*MOV A,STAT		
RCVR	= UART Receive Buffer — 8 bits	*IN RCVR		
XMTR	= UART Transmit Buffer — 8 bits	*OUT XMTR		
TCP	= Timing Chain Pointer	*MOV TCP,A		
@TCP	= Register Pointed to by TCP	*MOV @TCP,A		

*New instruction added to 8048 subset.

Symbol Definitions

Symbol	Definition
AC	Auxiliary Carry Flag
addr	Program Memory Address
b	Bit Designator (b = 0 – 7)
BS	RAM Bank Switch
data	Number or Expression (8 bits)
DBF	Program Memory Bank Select Bits (2)
EXI	External Interrupt Pin
F0, F1	Internal Flags
P	I/O Port (8 bits)

Symbol	Definition
PC	Program Counter
SP	Stack Pointer
TF	Timer Flag
#	Prefix for Immediate Data
@	Prefix for Indirect Address
( )	Contents of Register
(( ))	Contents of Memory Location pointed to by designated register
←	Replaced by

Instruction Set

Mnemonic	Machine Code	Function	Description	Cycles	Bytes	Flags				
						C	AC	HACC	F0	F1
)D A, Rr	0 1 1 0 1 r r r	(A) ← (A) + (Rr) for r= 0 – 7	Add contents of designated register to the Accumulator (8-bit operation)	1	1	*	*	*		
)D A, #data	0 0 0 0 0 0 1 1 d7 d6 d5 d4 d3 d2 d1 d0	(A) ← (A) + data	Add immediate the specified data to the Accumulator (8-bit operation)	2	2	*	*	*		
)D A, @ Rr	0 1 1 0 0 0 0 r	(A) ← (A) + ((Rr)) for r = 0 – 1	Add indirect the contents of data memory pointed to by Rr to the Accumulator (8-bit operation)	1	1	*	*	*		
)DC A, Rr	0 1 1 1 1 r r r	(A) ← (A) + (C) + (Rr) for r = 0 – 7	Add with carry the contents of the designated register to the Accumulator (8-bit operation)	1	1	*	*	*		
)DC A, # data	0 0 0 1 0 0 1 1 d7 d6 d5 d4 d3 d2 d1 d0	(A) ← (A) + (C) + data	Add immediate with carry the specified data to the Accumulator (8-bit operation)	2	2	*	*	*		
)DC A, @ Rr	0 1 1 1 0 0 0 r	(A) ← (A) + (C) + ((Rr)) for r = 0 – 1	Add indirect with carry the contents of data memory pointed to by Rr to the Accumulator (8-bit operation)	1	1	*	*	*		
)L A, Rr	0 1 0 1 1 r r r	(A) ← (A) AND (Rr) for r = 0 – 7	Logical AND contents of designated register with Accumulator (8-bit operation)	1	1					
)L A, # data	0 1 0 1 0 0 1 1 d7 d6 d5 d4 d3 d2 d1 d0	(A) ← (A) AND data	Logical AND specified Immediate Data with Accumulator (8-bit operation)	2	2					
)L A, @ Rr	0 1 0 1 0 0 0 r	(A) ← (A) AND ((Rr)) for r = 0 – 1	Logical AND indirect the contents of data memory pointed to by Rr with Accumulator (8-bit operation)	1	1					
)L PORT, # data	0 1 1 1 0 0 1 1 d7 d6 d5 d4 d3 d2 d1 d0	(P) ← (P) AND data	Logical AND immediate specified data with output port (8-bit operation)	2	2					
ALL addr	a10 a9 a8 1 0 1 0 0 a7 a6 a5 a4 a3 a2 a1 a0	((SP)) ← (PC0–12) ((SP)) ← (PSW3–7) (SP) ← (SP) + 1 (PC8–10) ← addr 8–10 (PC0–7) ← addr 0–7 (PC11–12 ← DBF 0, 1	Call designated subroutine	2	2					

Instruction Set (Continued)

Mnemonic	Machine Code	Function	Description	Cycles	Bytes	Flags			
						C	AC	HACC	F0
CLR A	0 0 1 0 0 1 1 1	(A) ← 0	Clear the Accumulator	1	1				
CLR C	1 0 0 1 0 1 1 1	(C) ← 0	Clear carry bit	1	1	*			
CLR F0	1 0 0 0 0 1 0 1	(FO) ← 0	Clear Flag 0	1	1				*
CLR F1	1 0 1 0 0 1 0 1	(F1) ← 0	Clear Flag 1	1	1				
CPL A	0 0 1 1 0 1 1 1	(A) ← NOT (A)	Complement the contents of the Accumulator (8-bit operation)	1	1				
CPL C	1 0 1 0 0 1 1 1	(C) ← NOT (C)	Complement carry bit	1	1	*			
CPL F0	1 0 0 1 0 1 0 1	(F0) ← NOT (F0)	Complement Flag 0	1	1				*
CPL F1	1 0 1 1 0 1 0 1	(F1) ← NOT (F1)	Complement Flag 1	1	1				
DA A	0 1 0 1 0 1 1 1		Decimal Adjust the contents of the Accumulator (8-bit operation)	1	1	*	*		
DEC A	0 0 0 0 0 1 1 1	(HACC, A) ← (HACC, A) − 1	Decrement by 1 the contents of HACC/ ACC	1	1	*		*	
DEC CURS	0 0 0 0 1 0 1 0	(CURS) ← (CURS) − 1	Decrement by 1 the contents of the Cursor Address Register	1	1				
DEC Rr	1 1 0 0 1 r r r	(Rr) ← (Rr) − 1	Decrement by 1 the contents of the designated register (8-bit operation)	1	1	*			
DECL Rr	0 0 0 0 1 0 0 r	(Rr) ← (Rr) − 1 for r = 0 − 1	Decrement by 1 the contents of the designated 16-bit register pair	1	1				
DIS II	0 0 1 1 0 1 0 1		Disable internal interrupts	1	1				
DIS XI	0 0 0 1 0 1 0 1		Disable external interrupts	1	1				

DJNZ Rr, addr	1 1 1 0 1 r r r
	a7 a6 a5 a4 a3 a2 a1 a0
		(Rr) ← (Rr) − 1 for r = 0 − 7
		If (Rr) ≠ 0 do (PC0−7) ← addr
		If (Rr) = 0 do (PC) ← PC + 2
			Decrement the specified register and Jump if not zero to designated address within page (8-bit decrement)	2	2				

EN II	0 0 1 0 0 1 0 1		Enable internal interrupts.	1	1				
EN XI	0 0 0 0 0 1 0 1		Enable external interrupt.	1	1				
INC A	0 0 0 1 0 1 1 1	(HACC, A) ← (HACC, A) + 1	Increment by 1 the contents of HACC/A.	1	1	*		*	
INC CURS	0 0 1 1 1 0 1 0	(CURS) ← (CURS) + 1	Increment by 1 the contents of the Cursor Address Register.	1	1				

Instruction Set (Continued)

nemonic	Machine Code	Function	Description	Cycles	Bytes	Flags				
						C	AC	HACC	F0	F1
C Rr	0 0 0 1 1 r r r	(Rr) ← (Rr) + 1 for r = 0 − 7	Increment by 1 the contents of the designated register (8-bit increment)	1	1	*				
C @ Rr	0 0 0 1 0 0 0 r	((Rr)) ← ((Rr)) + 1 for r = 0 − 1	Increment in direct the contents of data memory pointed to by Rr (8-bit increment)	1	1	*				
CL Rr	0 0 1 1 1 0 0 r	(Rr) ← (Rr) + 1 for r = 0 − 1	Increment by 1 the contents of the designated 16-bit register pair	1	1					
PORT	1 1 1 0 0 0 0 1	(A) ← (P)	Input data from port into Accumulator (8-bit transfer)	2	1					
RCVR	1 1 1 0 0 0 0 0	(A) ← (RCVR)	Input contents of UART Receive buffer into Accumulator (8-bit transfer). Also, clears Receive Buffer Full interrupt.	1	1					
b addr	b2 b1 b0 1 0 0 1 0 a7 a6 a5 a4 a3 a2 a1 a0	(PC0–7) ← addr if (b) = 1 (PC) ← (PC) + 2 if (b) = 0 for b = 0 − 7	Jump to specified address within page if Accumulator bit is set	2	2					
addr	1 1 1 1 0 1 1 0 a7 a6 a5 a4 a3 a2 a1 a0	(PC0–7) ← addr if C = 1 (PC) ← (PC) + 2 if C = 0	Jump to specified address within page if Carry flag is set	2	2					
0 addr	1 0 0 1 0 1 1 0 a7 a6 a5 a4 a3 a2 a1 a0	(PC0–7) ← addr if F0 = 1 (PC) ← (PC) + 2 if F0 = 0	Jump to specified address within page if Flag F0 is set	2	2					
1 addr	0 1 1 1 0 1 1 0 a7 a6 a5 a4 a3 a2 a1 a0	(PC0–7) ← addr if F1 = 1 (PC) ← (PC) + 2 if F1 = 0	Jump to specified address within page if Flag F1 is set	2	2					
MP addr	a10 a9 a8 0 0 1 0 0 a7 a6 a5 a4 a3 a2 a1 a0	(PC8–10) ← addr 8–10 (PC0–7) ← addr 0–7 (PC11–12) ← DBF 0, 1	Direct Jump to specified address within 2k Bank	2	2					
MPP @ A	1 0 1 0 0 0 1 1	(PC0–7) ← ((A))	Jump indirect within page to the address specified in the memory location pointed to by the Accumulator	2	1					
NC addr	1 1 1 0 0 1 1 0 a7 a6 a5 a4 a3 a2 a1 a0	(PC0–7) ← addr if C = 0 (PC) ← (PC) + 2 if C = 1	Jump within page to specified address if Carry flag is 0	2	2					

Instruction Set (Continued)

Mnemonic	Machine Code	Function	Description	Cycles	Bytes	Flags			
						C	AC	HACC	F0
JNF0 addr	1 0 0 0 0 1 1 0 a7 a6 a5 a4 a3 a2 a1 a0	(PC0–7) ← addr if F0 = 0 (PC) ← (PC) + 2 if F0 = 1	Jump within page to specified address if F0 is 0	2	2				
JNF1 addr	0 1 1 0 0 1 1 0 a7 a6 a5 a4 a3 a2 a1 a0	(PC0–7) ← addr if F1 = 0 (PC) ← (PC) + 2 if F1 = 1	Jump within page to specified address if F1 is 0	2	2				
JNTF addr	0 0 0 0 0 1 1 0 a7 a6 a5 a4 a3 a2 a1 a0	(PC0–7) ← addr if TF = 0 (PC) ← (PC) + 2 if TF = 1, (TF) ← 0	Jump within page to specified address if Timer flag is reset. If not, continue and reset TF	2	2				
JNXI addr	1 0 1 0 0 1 1 0 a7 a6 a5 a4 a3 a2 a1 a0	(PC0–7) ← addr if EXI = LOW (PC) ← (PC) + 2 if EXI = HIGH	Jump within page to specified address if External Interrupt pin is LOW	2	2				
JNZ addr	1 1 0 1 0 1 1 0 a7 a6 a5 a4 a3 a2 a1 a0	(PC0–7) ← addr if A ≠ 0 (PC) ← (PC) + 2 if A = 0	Jump within page to specified address if Accumulator is not 0	2	2				
JTF addr	0 0 0 1 0 1 1 0 a7 a6 a5 a4 a3 a2 a1 a0	(PC0–7) ← addr if TF = 1, (TF) ← 0 (PC) ← (PC) + 2 if TF = 0	Jump within page to specified address if Timer flag is set. If jump taken Timer flag reset	2	2				
JXI addr	1 0 1 1 0 1 1 0 a7 a6 a5 a4 a3 a2 a1 a0	(PC0–7) ← addr if EXI = HIGH (PC) ← (PC) + 2 if EXI = LOW	Jump within page to specified address if External Interrupt pin is HIGH	2	2				
JZ addr	1 1 0 0 0 1 1 0 a7 a6 a5 a4 a3 a2 a1 a0	(PC0–7) ← addr if A = 0 (PC) ← (PC) + 2 if A ≠ 0	Jump within page to specified address if Accumulator is 0	2	2				
MOV A, CURS	1 0 0 1 1 0 1 1	(HACC/A) ← (CURS)	Copy the contents of the Cursor Address Register into the HACC/A (16-bit transfer)	1	1			*	
MOV A, HACC	1 1 1 0 0 0 1 0	(A) ← (HACC)	Copy contents of the High Accumulator into the Low Accumulator (8-bit transfer)	1	1				
MOV A, HOME	1 0 0 1 1 0 1 0	(HACC/A) ← (HOME)	Copy the contents of the Home Address register into the HACC/A (16-bit transfer)	1	1			*	
MOV A, HPEN	0 0 1 1 1 1 1 1	(A0–6) ← (HPEN) (A7) ← 0	Copy the contents of the Horizontal Light Pen Register into the Accumulator (7-bit transfer, A7 cleared)	1	1				

Instruction Set (Continued)

Mnemonic	Machine Code	Function	Description	Cycles	Bytes	Flags				
						C	AC	HACC	F0	F1
OV A, INTR	1 0 0 0 1 1 0 0	(A) ← (INTR)	Copy the contents of the Interrupt Register into the Accumulator (8-bit transfer)	1	1					
OV A, PSW	1 1 0 0 0 1 1 1	(A) ← (PSW)	Copy contents of the Program Status word into the Accumulator (8-bit transfer)	1	1					
OV A, Rr	1 1 1 1 1 r r r	(A) ← (Rr) for r = 0 – 7	Copy the contents of the designated Register into the Accumulator (8-bit transfer)							
OV A, STAT	1 0 0 1 1 1 0 0	(A0–5) ← (STAT) (A6–7) ← 11	Copy the contents of the UART Status Latch into the Accumulator (6-bit transfer, A6 and A7 set)	1	1					
OV A, T	0 1 0 0 0 0 1 0	(A) ← (T)	Copy the contents of the Timer into the Accumulator (8-bit transfer)	1	1					
OV A, VPEN	0 0 1 1 1 1 1 0	(A0–4) ← (VPEN) (A5–7) ← O	Copy contents of the Vertical Light Pen Register into the Accumulator (5-bit transfer, A5–A7 cleared)	1	1					
IOV A, @ Rr	1 1 1 1 0 0 0 r	(A) ← ((Rr)) for r = 0 – 1	Copy indirect the contents of data memory pointed to by Rr into the Accumulator (8-bit transfer)	1	1					
IOV A, # data	0 0 1 0 0 0 1 1 d7 d6 d5 d4 d3 d2 d1 d0	(A) ← data	Load immediate the specified data into the Accumulator (8-bit load)	2	2					
IOV AL0, A	0 0 1 1 1 1 0 0	(AL0) ← (A)	Copy the contents of the Accumulator into Attribute Latch 0 (8-bit transfer)	1	1					
IOV AL1, A	0 0 1 1 1 1 0 1	(AL1) ← (A)	Copy the contents of the Accumulator into Attribute Latch 1 (8-bit transfer)	1	1					
IOV BAUD, A	0 0 0 0 0 0 1 0	(BAUD) ← (A)	Copy the contents of the Accumulator into the UART Baud Rate Select Register (8-bit transfer)	1	1					
IOV BEGD, A	0 0 0 0 1 1 0 1	(BEGD) ← (HACC/A)	Copy the contents of HACC/A into the Beginning of Display RAM Register (16-bit transfer)	1	1					

Instruction Set (Continued)

Mnemonic	Machine Code	Function	Description	Cycles	Bytes	Flags				
						C	AC	HACC	F0	F
MOV CURS, A	1 0 0 0 1 0 1 1	(CURS) ← (HACC/A)	Copy the contents of HACC/A into the Cursor Address Register (16-bit transfer)	1	1					
MOV ENDD, A	0 0 0 0 1 1 0 0	(ENDD) ← (HACC/A)	Copy the contents of HACC/A into the End of Display RAM Register (16-bit transfer)	1	1					
MOV HACC, A	1 1 0 0 0 0 1 0	(HACC) ← (A)	Copy the contents of the Low Accumulator into the High Accumulator (8-bit transfer)	1	1			*		
MOV HOME, A	1 0 0 0 1 0 1 0	(HOME) ← (HACC/A)	Copy the contents of HACC/A into the Home Address Register (16-bit transfer)	1	1					
MOV MASK, A	1 0 0 0 0 0 1 0	(MASK) ← (A)	Copy the contents of the Accumulator into the Interrupt Mask Register (8-bit transfer)	1	1					
MOV PSR, A	0 0 1 0 0 0 1 0	(PSR) ← (A)	Copy the contents of the Accumulator into the UART Prescale Register (8-bit transfer)	1	1					
MOV PSW, A	1 1 0 1 0 1 1 1	(PSW) ← (A)	Copy contents of the Accumulator into the Program Status Word (8-bit transfer)	1	1	*	*			
MOV Rr, A	1 0 1 0 1 r r r	(Rr) ← (A) for r = 0 − 7	Copy contents of the Accumulator into the designated register (8-bit transfer)	1	1					
MOV SCR, A	0 1 0 1 0 1 0 1	(SCR) ← (A)	Copy contents of the Accumulator into the System Control Register (8-bit transfer)	1	1					
MOV SROW, A	0 0 0 0 1 1 1 0	(SROW) ← (HACC/A)	Copy the contents of HACC/A into the Status Row Register (16-bit transfer)	1	1					
MOV T, A	0 1 1 0 0 0 1 0	(T) ← (A)	Copy the contents of the Accumulator into the Timer (8-bit transfer)	1	1					
MOV TCP, A	1 0 0 0 0 1 1 1	(TCP) ← (A)	Copy the contents of the Accumulator into the Timing Chain Pointer	1	1					

Instruction Set (Continued)

Mnemonic	Machine Code	Function	Description	Cycles	Bytes	Flags				
						C	AC	HACC	F0	F1
IOV UCR, A	0 0 0 0 0 0 0 1	(UCR) ← (A)	Copy the contents of the Accumulator into the UART Control Register (8-bit transfer)	1	1					
IOV VCR, A	0 1 0 0 0 1 0 1	(VCR) ← (A)	Copy the contents of the Accumulator into the Video Control Register (8-bit transfer)	1	1					
IOV VINT, A	1 0 1 0 0 0 1 0	(VINT) ← (A)	Copy the contents of the Accumulator into the Vertical Interrupt Register	1	1					
IOV Rr, # data	1 0 1 1 1 r r r d7 d6 d5 d4 d3 d2 d1 d0	(Rr) ← data for r = 0 – 7	Load immediate the specified data into the designated register (8-bit load)	2	2					
IOV @ Rr, A	1 0 1 0 0 0 0 r	((Rr)) ← (A) for r = 0 – 1	Copy indirect the contents of the Accumulator into the data memory location pointed to by Rr (8-bit transfer)	1	1					
IOV @ Rr, # data	1 0 1 1 0 0 0 r d7 d6 d5 d4 d3 d2 d1 d0	((Rr)) ← data for r = 0 – 1	Load indirect the specified immediate data into the data memory location pointed to by Rr (8-bit load)	2	2					
MOV @ TCP, A	1 0 1 1 0 1 1 1	((TCP)) ← (A) (TCP) ← (TCP) + 1	Copy indirect the contents of the Accumulator into the Timing Chain Register pointed to by TCP. Contents of TCP incremented by 1	1	1					
MOV UMX, A	0 0 1 1 0 0 1 1	(UMX) ← (A)	Copy the contents of the Accumulator into the UART Multiplex Register (8-bit transfer)	1	1					
MOVL A, R0	1 0 0 1 1 0 0 0	(HACC/A) ← (RA, R0)	Copy the contents of RA, R0 into HACC/A (16-bit transfer)	1	1			*		
MOVL A, R1	1 0 0 1 1 0 0 1	(HACC/A) ← (RB, R1)	Copy the contents of RB, R1 into HACC/A (16-bit transfer)	1	1			*		
MOVL R0, A	1 0 0 0 1 0 0 0	(RA, R0) ← (HACC/A)	Copy the contents of HACC/A into RA, R0 (16-bit transfer)	1	1					
MOVL R1, A	1 0 0 0 1 0 0 1	(RB, R1) ← (HACC/A)	Copy the contents of HACC/A into RB, R1 (16-bit transfer)	1	1					

7

Instruction Set (Continued)

Mnemonic	Machine Code	Function	Description	Cycles	Bytes	Flags				
						C	AC	HACC	F0	I
MOVP A, @ A	1 0 1 1 0 0 1 1	(PC0–7) ← (A) (A) ← ((PC)) (PC0–7) ← (old PC0–7) + 1	Replace low 8 bits of PC with A. Load indirect within page the contents of the memory location pointed to by new PC into Accumulator. Restore PC with old value plus 1. Operates in all memory banks.	2	1					
MOVP3 A, @ A	1 1 1 1 0 0 1 1	(PC0–7) ← (A) (PC8–10) ← 011 (A) ← ((PC)) (PC) ← (old PC) + 1	Replace low 8 bits of PC with A. Next 3 bits replaced with 011. Load indirect within page 3 the contents of the memory location pointed to by new PC into the Accumulator. Restore PC with old value plus 1. Operates in all memory banks.	2	1					
MOVX A, @ CURS	1 0 0 1 1 1 0 1	(HACC/A) ← ((CURS))	Copy indirect the contents of display memory as pointed to by CURS into HACC/A (16-bit transfer)	Min. 2	1			*		
MOVX A, @ R0	1 0 0 1 0 0 0 0	(HACC/A) ← ((RA, R0))	Copy indirect the contents of display memory as pointed to by RA, R0 into HACC/A (16-bit transfer)	Min. 2	1			*		
MOVX A, @ R1	1 0 0 1 0 0 0 1	(HACC/A) ← ((RB, R1))	Copy indirect the contents of display memory as pointed to by RB, R1 into HACC/A (16-bit transfer)	Min. 2	1			*		
MOVX @ CURS, A	1 0 0 0 1 1 0 1	((CURS)) ← (HACC/A)	Copy indirect the contents of HACC/A into the display memory location as pointed to by CURS (16-bit transfer)	Min. 2	1					
MOVX @ R0, A	1 0 0 0 0 0 0 0	((RA, R0)) ← (HACC/A)	Copy indirect the contents of HACC/A into the display memory location as pointed to by RA, R0 (16-bit transfer)	Min. 2	1					

Instruction Set (Continued)

Mnemonic	Machine Code	Function	Description	Cycles	Bytes	Flags				
						C	AC	HACC	F0	F1
)VX @ R1, A	1 0 0 0 0 0 0 1	((RB, R1)) ← (HACC/A)	Copy indirect the contents of HACC/A into the display memory location pointed to by RB, R1 (16-bit transfer)	Min. 2	1					
)P	0 0 0 0 0 0 0 0		No Operation	1	1					
;L A, Rr	0 1 0 0 1 r r r	(A) ← (A) OR (Rr) for r = 0 – 7	Logical OR contents of designated register with Accumulator (8-bit transfer)	1	1					
;L A, @ Rr	0 1 0 0 0 0 0 r	(A) ← (A) OR ((Rr)) for r = 0 – 1	Logical OR indirect the contents of the data memory location pointed to by Rr with Accumulator (8-bit operation)	1	1					
;L A, # data	0 1 0 0 0 0 1 1 d7 d6 d5 d4 d3 d2 d1 d0	(A) ← (A) OR data	Logical OR the specified immediate data with the Accumulator (8-bit operation)	2	2					
;L PORT, # data	0 1 1 0 0 0 1 1 d7 d6 d5 d4 d3 d2 d1 d0	(P) ← (P) OR data	Logical OR immediate specified data with output port	2	2					
JT PORT	1 1 0 0 0 0 0 1	(P) ← (A)	Output the contents of the Accumulator to the I/O Port (8-bit transfer)	2	1					
UT XMTR	1 1 0 0 0 0 0 0	(XMTR) ← (A)	Copy the contents of the Accumulator into the UART Transmit Buffer (8-bit transfer). Also clears Transmit Buffer empty interrupt	1	1					
ET	1 0 0 0 0 0 1 1	(SP) ← (SP) – 1 (PC0–12) ← ((SP))	Return from subroutine without restoring Program Status Word bits 5–7	2	1					
ETR	1 0 0 1 0 0 1 1	(SP) ← (SP) – 1 (PC0–12) ← ((SP)) (PSW 3–7) ← ((SP))	Return from Subroutine restoring Program Status Word (use for all returns from interrupts)	2	1	*	*			
;L A	1 1 1 0 0 1 1 1	(A_{n+1}) ← (An) for n = 0 – 6 (A0) ← (A7)	Rotate Accumulator left by 1 bit without carry	1	1					
;LC A	1 1 1 1 0 1 1 1	(A_{n+1}) ← (An) for n = 0 – 6 (A0) ← (C) (C) ← (A7)	Rotate Accumulator left by 1 bit through carry	1	1	*				

7

Instruction Set (Continued)

Mnemonic	Machine Code	Function	Description	Cycles	Bytes	Flags			
						C	AC	HACC	F(
RR A	0 1 1 1 0 1 1 1	(An) ← A_{n+1} for n = 0 − 6	Rotate Accumulator right by 1 bit without carry	1	1				
RRC A	0 1 1 0 0 1 1 1	(An) ← A_{n+1} for n = 0 −6 (A7) ← (C) (C) ← (A0)	Rotate Accumulator right by 1 bit through carry	1	1	*			
SEL MB0	1 1 0 0 0 1 0 1	(DBF) ← 00	Select Bank 0 (0–2047) of Program Memory	1	1				
SEL MB1	1 1 0 1 0 1 0 1	(DBF) ← 01	Select Bank 1 (2048–4095) of Program Memory	1	1				
SEL MB2	1 1 1 0 0 1 0 1	(DBF) ← 10	Select Bank 2 (4096–6143) of Program Memory	1	1				
SEL MB3	1 1 1 1 0 1 0 1	(DBF) ← 11	Select Bank 3 (6144–8191) of Program Memory	1	1				
SEL RBn	1 1 n 0 0 0 1 1	(BS) ← n for n = 0 − 1	Select Data RAM Bank (0–7) or 1 (24–31)	1	1				
STOP T	0 1 1 0 0 1 0 1		Stop Timer	1	1				
STRT T	0 1 1 1 0 1 0 1		Start Timer	1	1				
SWAP A	0 1 0 0 0 1 1 1	(A4–A7) ⟷ (A0–A3)	SWAP 4 bit nibbles in Accumulator	1	1				
XCH A, Rr	0 0 1 0 1 r r r	(A) ⟷ (Rr) for r = 0 − 7	Exchange the Accumulator and contents of designated register (8-bit transfer)	1	1				
XCH A, @ Rr	0 0 1 0 0 0 0 r	(A) ⟷ ((Rr)) for r = 0 − 1	Exchange indirect the contents of the Accumulator and the data memory location pointed to by Rr (8-bit transfer)	1	1				
XCHD A, @ Rr	0 0 1 1 0 0 0 r	(A0–3) ⟷ ((Rr)) 0–3 for r = 0 − 1	Exchange indirect the low 4 bits of the Accumulator and the data memory location pointed to by Rr (4-bit transfer)	1	1				
XRL A, Rr	1 1 0 1 1 r r r	(A) ← (A) XOR (Rr) for r = 0 − 7	Logical XOR contents of designated register with Accumulator (8-bit transfer)	1	1				
XRL A, @ Rr	1 1 0 1 0 0 0 r	(A) ← (A) XOR ((Rr)) for r = 0 − 1	Logical XOR indirect the contents of the data memory location pointed to by Rr with the Accumulator	1	1				
XRL A, # data	1 1 0 1 0 0 1 1 d7 d6 d5 d4 d3 d2 d1 d0	(A) ← (A) XOR data	Logical XOR the immediate specified data with the Accumulator	2	2				

TMP Opcode Chart

MSN \ LSN	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	MOV UCR, A	MOV BAUD, A	ADD A, #data	JMP (page 0)	EN XI	JNTF	DEC A	DECL R0	DECL R1	DEC CURS		MOV ENDD, A	MOV BECD, A	MOV SROW, A	
1	INC @R0	INC @R1	JB0	ADDC A, #data	CALL (page 0)	DIS XI	JTF	INC A	INC R0	INC R1	INC R2	INC R3	INC R4	INC R5	INC R6	INC R7
2	XCH A, @R0	XCH A, @R1	MOV PSR, A	MOV A, #data	JMP (page 1)	EN II		CLR A	XCH A, R0	XCH A, R1	XCH A, R2	XCH A, R3	XCH A, R4	XCH A, R5	XCH A, R6	XCH A, R7
3	XCHD A, @R0	XCHD A, @R1	JB1	MOV UMX, A	CALL (page 1)	DIS II		CPL A	INCL R0	INCL R1	INC CURS		MOV AL0, A	MOV AL1, A	MOV A, VPEN	MOV A, HFEN
4	ORL A, @R0	ORL A, @R1	MOV A,T	ORL A, #data	JMP (page 2)	MOV VCR, A		SWAP A	ORL A, R0	ORL A, R1	ORL A, R2	ORL A, R3	ORL A, R4	ORL A, R5	ORL A, R6	ORL A, R7
5	ANL A, @R0	ANL A, @R1	JB2	ANL A, #data	CALL (page 2)	MOV SCR, A		DA A	ANL A, R0	ANL A, R1	ANL A, R2	ANL A, R3	ANL A, R4	ANL A, R5	ANL A, R6	ANL A, R7
6	ADD A, @R0	ADD A, @R1	MOV T,A	ORL PORT, #data	JMP (page 3)	STOP T	JNF1	RRC A	ADD A, R0	ADD A, R1	ADD A, R2	ADD A, R3	ADD A, R4	ADD A, R5	ADD A, R6	ADD A, R7
7	ADDC A, @R0	ADDC A, @R1	JB3	ANL PORT, #data	CALL (page 3)	STRT T	JF1	RR A	ADDC A, R0	ADDC A, R1	ADDC A, R2	ADDC A, R3	ADDC A, R4	ADDC A, R5	ADDC A, R6	ADDC A, R7
8	MOVX @R0, A	MOVX @R1, A	MOV MASK, A	RET	JMP (page 4)	CLR F0	JNF0	MOV TCP, A	MOVL R0, A	MOVL R1, A	MOV HOME, A	MOV CURS, A	MOV A, INTR	MOVX @CURS, A		
9	MOVX A, @R0	MOVX A, @R1	JB4	RETR	CALL (page 4)	CPL F0	JF0	CLR C	MOVL A, R2	MOVL A, R1	MOV A, HOME	MOV A, CURS	MOV A, STAT	MOVX A, @ CURS		
A	MOV @R0, A	MOV @R1, A	MOV VINT, A	JMPP @A	JMP (page 5)	CLR F1	JNXI	CPL C	MOV R0, A	MOV R1, A	MOV R2, A	MOV R3, A	MOV R4, A	MOV R5, A	MOV R6, A	MOV R7, A
B	MOV @R0, #data	MOV @R1, #data	JB5	MOVP A, @A	CALL (page 5)	CPL F1	JXI	MOV @TCP, A	MOV R0, #data	MOV R1, #data	MOV R2, #data	MOV R3, #data	MOV R4, #data	MOV R5, #data	MOV R6, #data	MOV R7, #data
C	OUT XMTR	OUT PORT	MOV HACC, A	SEL RB0	JMP (page 6)	SEL MB0	JZ	MOV A, PSW	DEC R0	DEC R1	DEC R2	DEC R3	DEC R4	DEC R5	DEC R6	DEC R7
D	XRL A, @R0	XRL A, @R1	JB6	XRL A, #data	CALL (page 6)	SEL MB1	JNZ	MOV PSW, A	XRL A, R0	XRL A, R1	XRL A, R2	XRL A, R3	XRL A, R4	XRL A, R5	XRL A, R6	XRL A, R7
E	IN RCVR	IN PORT	MOV A, HACC	SEL RB1	JMP (page 7)	SEL MB2	JNC	RL A	DJNZ R0	DJNZ R1	DJNZ R2	DJNZ R3	DJNZ R4	DJNZ R5	DJNZ R6	DJNZ R7
F	MOV A, @R0	MOV A, @R1	JB7	MOVP3 A, @A	CALL (page 7)	SEL MB3	JC	RLC A	MOV A, R0	MOV A, R1	MOV A, R2	MOV A, R3	MOV A, R4	MOV A, R5	MOV A, R6	MOV A, R7

Ordering Information

ORDER PART NUMBERS

ROMless	NS405-A12N NS405-B12N NS405-C12N	NS405-B18N

Throughput Considerations In NS405 System Planning

National Semiconductor Corp.
Application Brief 14
James Murashige

The intricate timing relationships inherent in video generation require that a designer have a firm grasp of the fundamentals of NS405 operation in order to achieve his design objectives. Towards this end the key facets of NS405 operation will be examined and examples given.

The NS405 is a complete video controller that reads in video data, processes it and outputs it to a CRT. Given this, one may derive all essential operating parameters from the following two statements:

1. You must be able to read in video data faster than you output it.
2. Video data accesses are based on the CPU cycle which in turn is based on the crystal or dot clock.

Application of these two statements immediately leads to a limitation on the character cell width as follows:

if f = crystal frequency or dot clock

then $(f \div 1) \div 15$ or $(f \div 1.5) \div 15$ = CPU Instruction Execution Clock Frequency

Since there are three video data accesses each CPU Instruction Execution cycle, there are $3 * (f \div 1) \div 15$ or $3 * (f \div 1.5) \div 15$ video data accesses per second.

if w = dot width of character cell then $f \div w$ = number of character cells being displayed per second.

Statement 1 says that video data accesses/sec ≥ display characters/sec

for CPU Clock ÷ 1	for CPU Clock ÷ 1.5
$3 * (f \div 1) \div 15 \geq f \div w$	$3 * (f \div 1.5) \div 15 \geq f \div w$
$f \div 5 \geq f \div w$	$(3 * f) \div 22.5 \geq f \div w$
$1/5 \geq 1/w$	$3/22.5 \geq 1/w$
$w \geq 5$	$w \geq 7.5$

So depending on the CPU clock divide factor (÷1 or ÷1.5) the character cell width must be a minimum as shown.

Cell width also impacts CPU throughput since both the CPU and Video controller vie for video memory access through the DMA controller. The rules of access are simple and straightforward. The Video Controller gets as many of the accesses as it needs with the CPU getting any left over. The maximum access rate as already shown is $f \div 5$ or $f \div 7.5$ depending on the CPU clock divide. If the CPU attempts a video memory access when things are very busy it will be put into a wait state and remain frozen until things clear up. Of course, no display characters are necessary when the display is blanked, so during the horizontal and vertical retrace periods the CPU has unlimited access to video memory.

Normally, the CPU doesn't have to wait until horizontal retrace to get into video memory, but exactly how often it can get in during a display line requires analysis of the worst case video requirements.

Since the results can vary dramatically depending on the parameters chosen, two typical cases will be presented.

I. With a dot clock of 18 MHz the display line consists of 80 character cells, 9 dots across. Since the CPU clock divide must be 1.5 the video memory access rate is 18 MHz ÷ 7.5 = 2.4 MHz.

   To display one line requires (9×80)/18 MHz = 40 us.

   In one line time there are 2.4 MHz × 40 us = 96 video memory accesses. Of the 96, 80 are required for the characters displayed in the line leaving 16 available for the CPU. This is an average of one every six video memory accesses or once every two CPU instruction cycles. This would be fine since all CPU video memory instructions require two instruction cycles to execute anyway. However, in addition to the DMA controller the video circuits also employ a four level FIFO to insure a smooth data flow. The FIFO is normally kept full at four in which case it stops accessing video data and allows the CPU to have all the accesses. However, the FIFO can drop down quite far before starting to fill up again by taking all of the video memory accesses. The net effect is that instead of being evenly distributed, the accesses available to the CPU are clumped together with long gaps between clumps. Taking the worst case condition of the FIFO being completely empty and having to fill to four by taking the accesses which the CPU could have gotten, the longest gap is (4×6) + 5 = 29 accesses ≈ 10 CPU instruction cycles. Generally speaking this tends to happen towards the middle of a line since the FIFO is filled prior to the start of a line and tries to end a line empty. In fact, accesses for video are performed up to the second to the last display character. The FIFO prefetch for the next line is performed shortly after horizontal blanking starts.

II. If the dot clock is now 12 MHz with a display line of 80 character cells 7 dots across the CPU clock divide can be 1.

   The video memory access rate is 12 MHz ÷ 5 = 2.4 MHz.

   To do one line requires (7×80)/12 MHz = 46.7 us. In one line time there are 2.4 MHz × 46.7 us = 112 video memory accesses. Of the 112, 32 are now available to the CPU. This averages out to one every 3.5. Figuring the FIFO in, the worst case wait for the CPU becomes (4×3.5) + 2.5 = 16.5 accesses ≈ 6 CPU instruction cycles. A significant improvement over the first example.

In general, to maximize CPU access to video memory one must maximize the average number of "free" accesses during the display time. The number of free accesses as a fraction of the total number available is:

$(w - 5d)/w$ Where w = character cell dot width
d = CPU divide factor of 1 or 1.5

As can be seen, throughput performance depends entirely on the cell width and CPU clock divide. To maximize performance one would try to choose a large w and a d of 1.

Applying the delay imposed by the four level FIFO, the maximum CPU delay in accessing video memory becomes =

$(4w + 5d)/(w - 5d)$ Memory cycles

NS405-Series TMP External Interrupt Processing

National Semiconductor Corp.
Application Brief 16
James Murashige

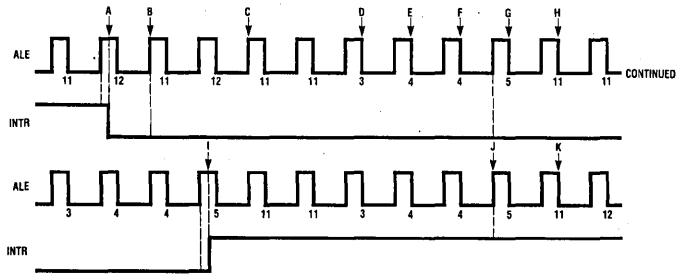
The TMP External Interrupt (INTR) is a level sampled interrupt input. Specifically this means that the input is sampled once each CPU cycle with interrupts being generated as long as the sampled input is a logic low. INTR shares pin 37 with RE10 and is sampled on each ALE rising edge as shown in the data sheet. If a logic low level is detected, interrupt service will commence if interrupts had been previously enabled with an EN XI instruction. Service consists of finishing up the currently executing instruction, pushing the PC and other pertinent information onto the stack, disabling all interrupts while in service and finally performing a JUMP to location 003. Upon completion of service a RETR would be executed to pop the stack and return to where we left off in the main program.

The exact timing involved may be observed through the example program of *Figure 1* and its instruction execution sequence in *Figure 2.* In *Figure 2* the numbers shown on the falling ALE edges are the program addresses put out by the TMP. As written the program will loop endlessly unless diverted by an external interrupt such as point A in *Figure 2.* Since it just missed the previous rising ALE edge it will not be until point B that the logic low INTR is read in. However, by then the CPU will have started execution of the first byte of the JMP 11 instruction. Since instructions are always finished once started, it will not be until point C that we begin interrupt service. At this point the next address would have been back at 11 but we now want to service the interrupt and push the stack. Stack pushing or popping takes 2 CPU cycles so the two address 11's shown following point C are dummies. Finally, we start interrupt service at point D by outputting address 003 and reading in the IN PORT instruction. Since the IN PORT instruction is only 1 byte long but takes 2 CPU cycles to execute, the address "4" at point E is a dummy and isn't really needed until point F when we read in the RETR instruction. Like IN PORT, RETR is a 1 byte instruction that takes 2 CPU cycles to execute. Therefore, the address "5" at point G is redundant. Upon returning from subroutine we immediately push the stack again (point H) since the interrupt is still there. Note that we immediately push the stack and do not execute the JMP at 11. Once more we go through the interrupt service routine but this time the interrupt ends at point I. Since it missed the preceding rising ALE edge where it was still seen as a logic low, we will immediately execute another interrupt service routine as shown. Finally, at point J as we prepare to return from service, INTR will be seen as a logic high and from point K onward execution will proceed normally.

When enabling and disabling interrupts, the rules for when you will and will not service them are predicated on the latest sampled interrupt level and last instruction executed. This is illustrated by the example program of *Figure 3* and instruction execution sequences of *Figure 4.* As shown in *Figure 4a,* the interrupt goes low at point A and will be sampled at the rising ALE of point B. However, since the current executing instruction (DIS XI at location 13) must be completed before starting interrupt service, the interrupt will be

```
ADDRESS          OPCODE          MNEMONIC
  000              04            JMP 010          ;RESET VECTOR
  001              10
  002
  003              E1            IN PORT          ;EXTERNAL INTERRUPT VECTOR
  004              93            RETR
  005
  006
  007
  008
  009
  00A
  00B
  00C
  00D
  00E
  00F
  010              05            EN XI            ;MAIN PROGRAM
  011              04            JMP 001
  012              11
  013
  014
  015
```

FIGURE 1. INTR Service Timing Example Program



TL/DD/6972-1

FIGURE 2. INTR Service Timing

locked out. Execution continues unperturbed until the interrupt is re-enabled with an EN XI from location 11, point F. Although the interrupt went logic high at point E it was still sampled as a logic low at point D.

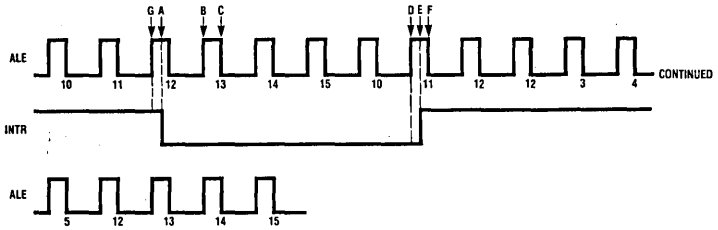
Therefore, after executing the EN XI at location 11, interrupt service will commence as shown. If the interrupt had gone logic high before point D it would have been sampled high and no interrupt service would have been performed.

Returning to the missed interrupt at point A, if the interrupt low had come in time to be sampled at point G, the instruction at 12 would have been the last one executed before interrupt service started as demonstrated in *Figure 4b.*

Although describing the external interrupt, all of the service sequences presented may be directly applied to TMP internal interrupts.

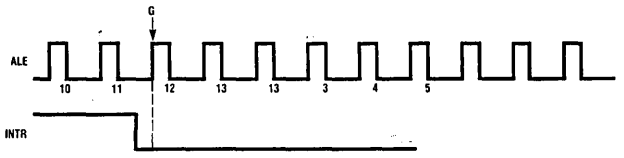
```
ADDRESS     OPCODE      MNEMONIC
  000         04        JMP 010          ;RESET VECTOR
  001         10
  002
  003         00        NOP              ;EXTERNAL INTERRUPT VECTOR
  004         93        RETR
  005
  006
  007
  008
  009
  00A
  00B
  00C
  00D
  00E
  00F
  010         00        NOP              ;MAIN PROGRAM
  011         05        EN XI
  012         00        NOP
  013         15        DIS XI
  014         04        JMP 010
  015         10
  016
```

FIGURE 3. INTR Enable/Disable Timing Example Program



TL/DD/6972-2

FIGURE 4a. INTR Enable/Disable Timing



TL/DD/6972-3

FIGURE 4b

TMP Row and Attribute Table Lookup Operation

National Semiconductor Corp.
Application Note 354
James Murashige

This note describes in detail the operation of the TMP Attribute Demo Program - TAD. Although a short program, it nicely demonstrates row table lookup operation in the TMP while at the same time putting out a visual display of the various video attributes available in the chip. While this display management approach is much more involved than normal sequential lookup mode, it is necessary when attemping to do fast screen updates or line editing with the TMP.

The hardware environment for which the program was written is the TMP Demo board. Appropriate references to and descriptions of the hardware will be made as necessary. For those who have not seen it, the net function of the program is to put up and manage a single frame of video data. In the top half of the display the same message is repeated 5 times but each time with a different set of attributes. In the lower half of the display are 4 rows representing the 128 possible block graphics patterns. All of the attribute effects displayed are achieved by updating the internal ALO attribute latch at the end of each display row. At the same time a message table lookup is performed in order to obtain the appropriate character string that will work with the new attribute set selected.

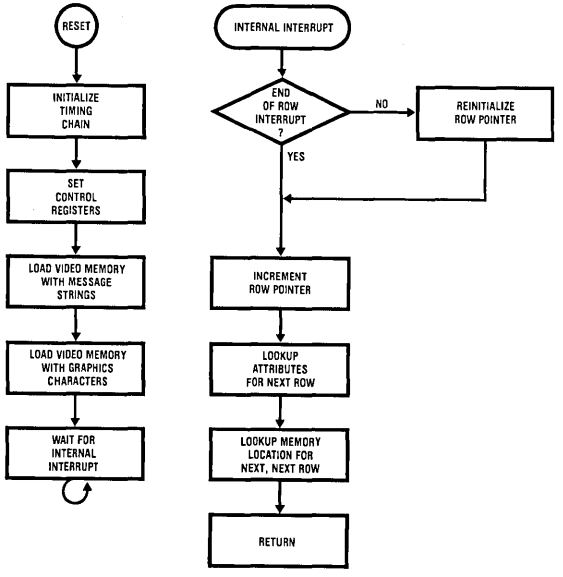
The flowchart for the program is shown in *Figure 1.* As you can see, the program essentially consists of initialization and waiting for and servicing video interrupts to manage the screen display. Initialization starts at BEGIN with the Vertical Interrupt Register and Timing Chain being loaded first. The Vertical interrupt is used for end of frame synchronization and is set to activate after the 27th row. The Timing Chain is loaded as follows:

TCP	0	Horizontal Length	= 104
	1	Characters/Row	= 80
	2	Horizontal Sync Begin	= 84
	3	Horizontal Sync End	= 100
	4	Character Height	= 10
		Extra Scans/Frame	= 2
	5	Vertical Length	= 27
	6	Vertical Blank	= 25
	7	Vertical Sync Begin/End	= 7,3
	8	Status Row Begin	= 31
	9	Blink Rate/D.C.	= F4H
	10	Graphics Column Register	= 30H
	11	Graphics Row Register	= 36H
	12	Underline Size Register	= 89H
	13	Cursor Size Register	= 09H

Given these values, one can ascertain that the display is 80 columns across and 25 rows tall. The character cell height is 10 scan lines and no status line will be displayed. The character underline is the bottom most scan line in a cell and the cursor occupies an entire cell. The partitioning of the block graphics cells is as follows:

```
0011100
0011100
0011100
2233344
2233344
5566655
5566655
5566655
5566655
```

TAD Flowchart



TL/C/5729–1

FIGURE 1

Following timing chain initialization various system registers are set to configure the chip to operate in its hardware environment. The video memory is a 2kX8 NMC2116 located between addresses 000-7FF. The crystal dot clock is 12 Mhz allowing us to use divide by 1 to generate the CPU clock. Accordingly the SCR is set to 24H (SB8-15 address output only, cell width = 7, divide by 1 for CPU clock, row table lookup operation). RAM Bank 0 is selected and HOME, BEGD, RA/RO are cleared. ENDD and CURS are set to 7FFFH and AL1 is set to FFH (no attributes selected). Video display memory (80×25 char) is then cleared out by storing spaces at all of the memory locations. Along with the spaces, attribute latch 1 is specified to be used. Video is then turned on by setting the VCR to 21H (normal alphanumeric display, internal attribute latch operation, normal video).

Next, the message tables are built up in the video memory. By updating the attribute latch AL0 each row, the entire screen display can be constructed from the 7 message rows stored in memory. Each of the message rows consist of 80 consecutive characters and are called up for display by loading the HOME register with the address of the first character in the row. The background characters in each of the rows are the spaces previously stored. Each of the display characters stored use attribute latch AL0 which is updated each row. The first row (0-79) consists entirely of spaces to provide us with a blank display row. The second row (80-159) has the message "tmp does it BETTER!" for normal and double high display. The third row (160-239) contains "ttmmpp ddooeess iitt BBEETTTTEERR!!" for double wide and double size display. Rows 4–7 contain 32 block graphics characters per row for a total of 128 patterns. The 128 characters stored are merely all binary combinations of the low 7 data bits in ascending order. The 32 characters in each row are stored in every other memory location to achieve a blank space between characters. For all of the message rows, data is positioned to give a centered display on the screen.

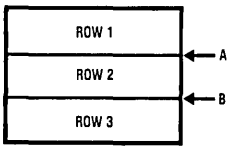
With initialization accomplished, we set the interrupt mask, re-enable interrupts and wait for a video interrupt.

Video display management is performed by the internal interrupt service routine located at 007 and consists of updating the HOME register and AL0 at the end of each display row. To accomplish this, a row counter (R3) is used as a pointer into the data lookup tables which follow the interrupt service routine. The R3 row counter is incremented on each End of Row interrupt or preset and incremented on a re-synching Vertical Interrupt.

Because the next row pointers are pipelined in the video memory controller, an understanding of End of Row and Vertical Interrupt operation is necessary in order to correctly set up the interrupt service routine and lookup tables. In table lookup mode, the Current Row Start Register (CRSR), which is a pointer to the first character address in a row, is automatically reloaded from the HOME register after the display of the last scan line in a row, a few characters into horizontal blanking. The timing of the CRSR reload when operating in sequential lookup mode is the same but in this case the pointer is advanced by the character width of the display row. It is the reloading of CRSR either in sequential or table lookup modes that generates the End of Row interrupt. The duration of the signal is ⅓ CPU cycle making it a one time event each row. The End of Row interrupt register bit is cleared when a reload of HOME, i.e., MOV HOME, A is executed. A simple example will illustrate the pipelining involved. In *Figure 2*, at the end of Row 1 (Point A) an EOR interrupt is generated. In preparation for this event HOME should have been loaded with the starting address of ROW 2 since the interrupt is generated when CRSR reloads from HOME. In service of the EOR, the program would load HOME with the starting address of ROW 3 in preparation for the EOR interrupt at Point B. However, notice that we have an entire row time from A to B to do the HOME reload. Finally note that EOR's are generated at the end of all rows except those blanked during vertical blanking. Vertical Interrupt operates with the same timing as End of Row except that it is specified to occur at the end of a particular row designated by the Vertical Interrupt Register. The row that it is specified to occur on must be < = Vertical Length Register (timing chain rows are counted starting from 0). Otherwise, it will never occur since the row counter will never count up that far. Usually Vertical Interrupt is specified to occur on a row blanked during vertical blanking so that it may be used as a frame sync signal.

Returning to TAD, *Figure 3* shows the interrupt positioning for all of the rows on the screen including the blanked ones. There are 25 displayed rows and 2 blanked ones in a frame for a total of 27. In addition, there are 2 extra scan lines which may be ignored as far as interrupt operation is concerned. Vertical Interrupt is set to occur at the end of the last row in the frame as shown. Row pointer operation for rows 2 to 24 is pipelined as described in *Figure 2*. At the end of ROW 24 (point E) the CRSR will be loading the pointer to ROW 25 and the interrupt service will load HOME with the pointer to ROW 1. At the end of ROW 25 (point F) the CRSR will load the pointer to ROW 1 and save it for the next frame. Since no EOR's are generated during vertical blanking, CRSR will remain static until ROW 1. At this point, it doesn't matter what the interrupt service loads into HOME and AL0 since the Vertical Interrupt at ROW 27 will reset the row counter and perform a new lookup for HOME and AL0. A Vertical Interrupt will not do a CRSR load, thus the pointer to ROW 1 will be preserved. At Vertical Interrupt, the row counter will be reset to 0 and we will want to do a pointer lookup for ROW 2 in preparation for the CRSR load at the end of ROW 1 (point A). Correspondingly, the row pointer lookup tables are organized 2 to 25, 1. Since the attribute latches aren't pipelined, the AL0 lookup table is arranged 1 to 25 since the new attribute set will be needed immediately for the display of the next row.

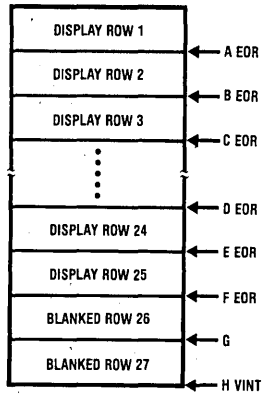
Row Table Lookup Pipelining



TL/C/5729–2

FIGURE 2

TAD Interrupt Positioning



TL/C/5729-3

FIGURE 3

TMP Attribute Demo Program

```
 1
 2
 3                .TITLE MAIN, "TMP ATTRIBUTE DEMO - TAD'
 4
 5                ;                              James Murashige 10/05/83
 6                ;This program displays the various character attributes available with
 7                ;the TMP by dynamically updating the attribute latch each display row.
 8                ;In addition it uses End of Row and Vertical interrupts to perform row
 9                ;table lookup screen refreshing.
10
11       0000     LINE 1   =     0      ;LINE 1 START, ALL BLANKS
12       0050     LINE 2   =    80      ;LINE 2 START, NORMAL MESSAGE
13       00A0     LINE 3   =   160      ;LINE 3 START, DOUBLE WIDE MESSAGE
14       00F0     LINE 4   =   240      ;LINE 4 START, FIRST GRAPHICS LINE
15       0140     LINE 5   =   320      ;LINE 5 START, SECOND GRAPHICS LINE
16       0190     LINE 6   =   400      ;LINE 6 START, THIRD GRAPHICS LINE
17       01E0     LINE 7   =   480      ;LINE 7 START, FOURTH GRAPHICS LINE
18
19
20
21       0000     . = 00                ;START AT PROGRAM LOCATION 0
22
23 0000 0468      RESET;  JMP BEGIN     ;VECTOR TO RESET CODE
24
25       0003     . = 03
26
27                EXI:                  ;VECTOR TO EXTERNAL INTERRUPT PROCESSING
28
29       0007     . = 07
30
31                INI:                  ;VECTOR TO INTERNAL INTERRUPT PROCESSING
32
33 0007 8C                MOV A, INTR     ;READ INTERRPUT REGISTER
34 0008 320C              JBI EOR         ;HAVE AN EOR INTERRPUT
35 000A BBFF              MOV R3, #0FF    ;VINT INTERRUPT
36 000C 1B       EOR:     INC R3          ;INCREMENT TO DO NEXT ROW
37 000D 234F              MOV A, #ATT0    ;GET ATTRIBUTE LATCH 0
38 000F 6B                ADD A, R3
39 0010 B3                MOVP A, @A
40 0011 3C                MOV AL0, A      ;LOAD ATTRIBTE LATCH 0
41
42 0012 231D              MOV A, #HOMHIG  ;GET HOME HIGH ORDER BYTE
43 0014 6B                ADD A, R3
```

TMP Attribute Demo Program (Continued)

```
 44 0015 B3              MOVP A, @A
 45 0016 C2              MOV HACC, A
 46 0017 2336            MOV A, #HOMLOW ;GET HOME LOW ORDER BYTE
 47 0019 6B              ADD A, R3
 48 001A B3              MOVP A, @A
 49 001B 8A              MOV HOME, A      ;LOAD HOME
 50 001C 93              RETR
 51              .FORM
 52
 53              ;HOME HIGH ORDER BYTE LOOKUP TABLE
 54
 55 001D 00      HOMHIG: .BYTE 0             ;ROW 2
 56 001E 00              .BYTE 0             ;ROW 3
 57 001F 00              .BYTE 0             ;ROW 4
 58 0020 00              .BYTE 0             ;ROW 5
 59 0021 00              .BYTE 0             ;ROW 6
 60 0022 00              .BYTE 0             ;ROW 7
 61 0023 00              .BYTE 0             ;ROW 8
 62 0024 00              .BYTE 0             ;ROW 9
 63 0025 00              .BYTE 0             ;ROW 10
 64 0026 00              .BYTE 0             ;ROW 11
 65 0027 00              .BYTE 0             ;ROW 12
 66 0028 00              .BYTE 0             ;ROW 13
 67 0029 00              .BYTE 0             ;ROW 14
 68 002A 00              .BYTE 0             ;ROW 15
 69 002B 00              .BYTE 0             ;ROW 16
 70 002C 00              .BYTE 0             ;ROW 17
 71 002D 00              .BYTE 0             ;ROW 18
 72 002E 01              .BYTE H(LINE5)      ;ROW 19
 73 002F 00              .BYTE 0             ;ROW 20
 74 0030 01              .BYTE H(LINE6)      ;ROW 21
 75 0031 00              .BYTE 0             ;ROW 22
 76 0032 01              .BYTE H(LINE7)      ;ROW 23
 77 0033 00              .BYTE 0             ;ROW 24
 78 0034 00              .BYTE 0             ;ROW 25
 79 0035 00              .BYTE 0             ;ROW 1
 80 0036 00              .FORM
 81
 82              ;HOME LOW ORDER BYTE LOOKUP TABLE
 83
 84 0036 00      HOMLOW: .BYTE 0             ;ROW 2 BLANK
 85 0037 50              .BYTE L(LINE2)      ;ROW 3 NORMAL
 86 0038 00              .BYTE 0             ;ROW 4 BLANK
 87 0039 A0              .BYTE L(LINE3)      ;ROW 5 DOUBLE WIDE
 88 003A 00              .BYTE 0             ;ROW 6 BLANK
 89 003B 50              .BYTE L(LINE2)      ;ROW 7 DOUBLE HIGH
 90 003C 50              .BYTE L(LINE2)      ;ROW 8 DOUBLE HIGH
 91 003D 00              .BYTE 0             ;ROW 9 BLANK
 92 003E A0              .BYTE L(LINE3)      ;ROW 10 DOUBLE SIZE
 93 003F A0              .BYTE L(LINE3)      ;ROW 11 DOUBLE SIZE
 94 0040 00              .BYTE 0             ;ROW 12 BLANK
 95 0041 A0              .BYTE L(LINE3)      ;ROW 13 DOUBLE SIZE
 96 0042 A0              .BYTE L(LINE3)      ;ROW 14 DOUBLE SIZE
 97 0043 00              .BYTE L(LINE1)      ;ROW 15 BLANK
 98 0044 00              .BYTE L(LINE1)      ;ROW 16 BLANK
 99 0045 F0              .BYTE L(LINE4)      ;ROW 17 GRAPHICS
100 0046 00              .BYTE L(LINE1)      ;ROW 18 BLANK
101 0047 40              .BYTE L(LINE5)      ;ROW 19 GRAPHICS
102 0048 00              .BYTE L(LINE1)      ;ROW 20 BLANK
103 0049 90              .BYTE L(LINE6)      ;ROW 21 GRAPHICS
104 004A 00              .BYTE L(LINE1)      ;ROW 22 BLANK
105 004B E0              .BYTE L(LINE7)      ;ROW 23 GRAPHICS
106 004C 00              .BYTE L(LINE1)      ;ROW 24 BLANK
107 004D 00              .BYTE L(LINE1)      ;ROW 25 BLANK
108 004E 00              .BYTE L(LINE1)      ;ROW 1 BLANK
```

TMP Attribute Demo Program (Continued)

```
109                     .FORM
110
111                     ;ATTRIBUTE LATCH 0 LOOKUP TABLE
112
113 004F FF     ATT0:   .BYTE 0FF           ;ROW 1
114 0050 FF             .BYTE 0FF           ;ROW 2
115 0051 FF             .BYTE 0FF           ;ROW 3
116 0052 FF             .BYTE 0FF           ;ROW 4
117 0053 EF             .BYTE 0EF           ;ROW 5
118 0054 FF             .BYTE 0FF           ;ROW 6
119 0055 F7             .BYTE 0F7           ;ROW 7
120 0056 B7             .BYTE 0B7           ;ROW 8
121 0057 FF             .BYTE 0FF           ;ROW 9
122 0058 E7             .BYTE 0E7           ;ROW 10
123 0059 A7             .BYTE 0A7           ;ROW 11
124 005A FF             .BYTE 0FF           ;ROW 12
125 005B E2             .BYTE 0E2           ;ROW 13
126 005C 82             .BYTE 082           ;ROW 14
127 005D FF             .BYTE 0FF           ;ROW 15
128 005E FF             .BYTE 0FF           ;ROW 16
129 005F 7F             .BYTE 07F           ;ROW 17
130 0060 FF             .BYTE 0FF           ;ROW 18
131 0061 7F             .BYTE 07F           ;ROW 19
132 0062 FF             .BYTE 0FF           ;ROW 20
133 0063 7F             .BYTE 07F           ;ROW 21
134 0064 FF             .BYTE 0FF           ;ROW 22
135 0065 7F             .BYTE 07F           ;ROW 23
136 0066 FF             .BYTE 0FF           ;ROW 24
137 0067 FF             .BYTE 0FF           ;ROW 25
138                     .FORM
139
140                     ;START OF INITIALIZING CODE
141
142 0068 15     BEGIN;  DIS XI              ;INTERRUPTS OFF FOR NOW
143 0069 35             DIS II
144 006A 65             STOP T
145 006B 231A           MOV A, #26
146 006D A2             MOV VINT, A
147 006E 27             CLR A               ;SET UP TIMING CHAIN FOR DEMO BOARD
148 006F 87             MOV TCP, A
149 0070 2367           MOV A, #103         ;HORIZONTAL LENGTH
150 0072 B7             MOV @TCP, A
151 0073 234F           MOV A, #79          ;CHARACTERS/ROW
152 0075 B7             MOV @TCP, A
153 0076 2353           MOV A, #83          ;HORIZONTAL SYNC BEGIN
154 0078 B7             MOV @TCP, A
155 0079 2363           MOV A, #99          ;HORIZONTAL SYNC END
156 007B B7             MOV @TCP, A
157 007C 2391           MOV A, #091         ;CHARACTER HEIGHT/EXTRA SCANS
158 007E B7             MOV @TCP, A
159 007F 231A           MOV A, #26          ;VERTICAL LENGTH
160 0081 B7             MOV @TCP, A
161 0082 2318           MOV A, #24          ;VERTICAL BLANK
162 0084 B7             MOV @TCP, A
163 0085 2362           MOV A, #062         ;VERTICAL SYNC BEGIN/END
164 0087 B7             MOV @TCP, A
165 0088 231E           MOV A, #30          ;STATUS ROW BEGIN
166 008A B7             MOV @TCP, A
167 008B 23F4           MOV A, #0F4         ;BLINK RATE
168 008D B7             MOV @TCP, A
169 008E 2330           MOV A, #030         ;GRAPHICS COLUMN REGISTER
170 0090 B7             MOV @TCP, A
171 0091 2336           MOV A, #036         ;GRAPHICS ROW REGISTER
172 0093 B7             MOV @TCP, A
173 0094 2389           MOV A, #089         ;UNDERLINE SIZE REGISTER
```

ΓMP Attribute Demo Program (Continued)

```
.74 0096 B7              MOV @TCP, A
.75 0097 2309            MOV A, #009     ;CURSOR SIZE REGISTER
.76 0099 B7              MOV @TCP, A
.77
.78
.79              .FORM
.80 009A 2324            MOV A, #024     ;SET SYSTEM CONTROL REGISTER
.81 009C 55              MOV SCR, A      ;8 BI,7 DOTS, DIVIDE 1, TABLE LOOKUP
.82
.83 009D C3              SEL RB0         ;SELECT RAM BANK 0
.84 009E 27              CLR A           ;SET RAM POINTERS
.85 009F C2              MOV HACC, A
.86 00A0 8A              MOV HOME, A
.87 00A1 0D              MOV BEGD, A
.88 00A2 88              MOVL R0, A      ;CLEAR MEMORY POINTER
.89
.90 00A3 237F            MOV A, #07F
.91 00A5 C2              MOV HACC, A
.92 00A6 23FF            MOV A, #0FF
.93 00A8 0C              MOV ENDD, A
.94 00A9 8B              MOV CURS, A
.95 00AA 3D              MOV AL1, A      ;NO ATTRIBUTES FOR LATCH 1
.96
.97              ;CLEAR OUT MEMORY
.98
.99 00AB BD19            MOV R5, #25     ;DO 25 ROWS
:00 00AD BA50            MOV R2, #80     ;DO 80 CHARACTERS PER ROW
:01 00AF 23A0            MOV A, #0A0     ;INITIALIZE FOR A SPACE, ATTRIBUTE LATCH 1
:02
:03 00B1 80      LOOP:   MOVX @R0, A     ;STORE A CHARACTER
:04 00B2 38              INCL R0         ;INCREMENT POINTER
:05 00B3 EAB1            DJNZ R2, LOOP   ;TEST IF ROW DONE
:06 00B5 BA50            MOV R2, #80
:07 00B7 EDB1            DJNZ R5, LOOP   ;TEST IF SCREEN DONE
:08
:09 00B9 2321            MOV A, #021     ;SET VCR FOR INTERNAL ATTRIBUTES
:10 00BB 45              MOV VCR, A      :INTERNAL CHARACTER GENERATOR
:11
:12
:13              ;FIRST LINE ARE ALL BLANKS, SECOND LINE HAS SINGLE SPACING MESSAGE
:14 00BC 2300            MOV A, #H(LINE2+30)   ;SET R0 POINTER TO FIRST LINE
:15 00BE C2              MOV HACC, A
:16 00BF 236E            MOV A, #L(LINE2+30)
:17 00C1 88              MOVL R0, A
:18 00C2 BAE0            MOV R2, #L(MSG1)   ;SET R2 TO MESSAGE #1
:19 00C4 BB13            MOV R3, #19     ;SET R3 TO MESSAGE LENGTH
:20 00C6 FA      DISP1:  MOV A, R2
:21 00C7 B3              MOV A, @A       ;DISPLAY NORMAL MESSAGE
:22 00C8 80              MOVX @R0,A
:23 00C9 38              INCL R0
:24 00CA 1A              INC R2
:25 00CB EBC6            DJNZ R3, DISP1
:26              .FORM
:27              ;THIRD LINE HAS DOUBLE WIDE MESSAGE
:28 00CD 98              MOVL A, R0                ;SET R0 POINTER
:29 00CE 0334            ADD A, #(31 + 21)        ;LINES3 + 21
230 00D0 88              MOVL R0, A
231 00D1 BAE0            MOV R2, #L(MSG1)
232 00D3 BB13            MOV R3, #19
233 00D5 FA      DISP2:  MOV A, R2
234 00D6 B3              MOVP A, @A      ;DISPLAY DOUBLE WIDE
235 00D7 80              MOVX @R0, A
236 00D8 38              INCL R0
237 00D9 80              MOVX @R0, A
238 00DA 38              INCL R0
```

TMP Attribute Demo Program (Continued)

```
239 00DB 1A             INC R2
240 00DC EBD5           DJNZ R3, DISP2
241 00DE 04F3           JMP FOURTH
242
243 00E0 74       MSG1:   .BYTE 'tmp does it BETTER!'
244
245               ;FOURTH LINE STARTS GRAPHICS CHARACTERS DISPLAY
246 00F3 98       FOURTH: MOVL A, R0
247 00F4 031D             ADD A, #(21 + 8)     ;LINE4 + 8
248 00F6 88               MOVL R0, A
249 00F7 BB04             MOV R3, #4          ;DO 4 LINES
250 00F9 BA20             MOV R2, #32         ;DO 32 GRAPHICS CHARACTERS PER LINE
251 00FB 2300             MOV A, #000          ;ATTRIBUTE LATCH 0 SELECTED
252 00FD 2400             JMP BLOOP
253
254       0100      . = 0100
255
256 0100 80       BLOOP:  MOVX @R0, A         ;STORE CHARACTER
257 0101 38               INCL R0
258 0102 38               INCL R0
259 0103 17               INC A
260 0104 EA00             DJNZ R2, BLOOP
261
262 0106 BA20             MOV R2, #32      ;INITIALIZE FOR NEW ROW
263 0108 AC               MOV R4, A        ;TEMPORARY SAVE A
264 0109 98               MOVL A, R0
265 010A 0310             ADD A, #(8+8)  ;POINT TO NEXT LINE
266 010C 88               MOVL R0, A
267 010D FC               MOV A, R4        ;RESTORE A
268 010E EB00             DJNZ R3, BLOOP  ;CONTINUE IF NOT THROUGH
269
270               ;REENABLE INTERNAL INTERRUPTS AND MASK OFF UNUSED ONES
271 0110 2303             MOV A, #03
272 0112 82               MOV MASK, A
273 0113 25               EN II            ;REENABLE INTERNALS
274 0114 2414     PAU:    JMP PAU          ;WAIT FOR A VIDEO INTERRUPT
275               END
ATTO    004F    BEGIN   0068    BLOOP   0100    DISP1   00C6
DISP2   00D5    EOR     000C    EXI     0003 *  FOURTH  00F3
HOMHIG  001D    HOMLOW  0036    INI     0007 *  LINE1   0000
LINE2   0050    LINE3   00A0    LINE4   00F0    LINE5   0140
LINE6   0190    LINE7   01E0    LOOP    00B1    MSG1    00E0
PAU     0114    RESET   0000 *

NO ERROR LINES
  272 ROM BYTES USED

SOURCE CHECKSUM=CF60
OBJECT CHECKSUM=0576

INPUT FILE A: TAD. MAC
LISTING FILE A: TAD. PRN
OBJECT FILE A: TAD. LM
```

TMP - Dynamic RAM Interfacing

National Semiconductor Corp.
Application Note 355
James Murashige

TMPs Interface easily and directly to dynamic RAMs as illustrated in the basic TMP system schematic of *Figure 1*. In addition to providing the necessary Read/Write cycle control, the TMP will also automatically refresh the memories through the video controller, further easing interface requirements.

The circuitry to the right of the TMP provides program memory interfacing and I/O support while to the left lie the dynamic video RAM circuits. The memory width shown here is 8 bits although 16 bits can easily be accommodated. Using the 64K × 1 dynamic RAMs shown the entire video memory space is filled with RAM. However, by using a slightly modified addressing configuration smaller memroy chips could be substituted.

The requisite dynamic RAM control signals $\overline{RAS}$, $\overline{CAS}$ and $\overline{WE}$ are generated directly from the system bus control signals RAM ALE, $\overline{RAM\ RD}$ and $\overline{RAM\ WR}$. RAM ALE is used directly as $\overline{RAS}$ while $\overline{RAM\ WR}$ serves as $\overline{WE}$. $\overline{CAS}$ is the logical AND of $\overline{RAM\ RD}$ and $\overline{RAM\ WR}$. The 16 system bus bits are multiplexed down to the 8 bit RAM address vector by the two 74LS157's under the control of the RAM ALE. As configured, the row and column addresses strobed in are SBO-7 and SB8-15 respectively.

With the configuration shown, the pertinent TMP Read and Write cycle timing parameters for *Figure 2* are listed in Table 1. Going through the table one sees that the TMP easily interfaces to 150 ns access RAMs and will routinely work with 200 ns RAMs. The four parameters which may be a tight squeeze for 200 ns RAMs are:

1. t_{RAC}— Access Time from $\overline{RAS}$ is max 150 ns, typ 220 ns. This is a basic access time requirement which necessitates fast parts.
2. t_{RAH}— Row Address Hold Time is min 10 ns, typ 15 ns. This parameter is entirely dependent on the switching speed of the 74LS157.
3. t_{RCD}— $\overline{RAS}$ to $\overline{CAS}$ Delay Time is min 10 ns, typ 50 ns. This parameter isn't too critical since most dynamic RAMs internally gate the $\overline{CAS}$ signal should it come along too early.
4. t_{RP}— $\overline{RAS}$ Precharge Time is min 100 ns, typ 135 ns. Since $\overline{RAS}$ is actually the RAM ALE signal t_{RP} is the high time of RAM ALE.

However, rather than getting faster RAMs one could also meet spec by running the TMP CPU slower, thereby stretching out the allowable access time.

Since the TMP video controller will regularly and automatically access video memory in order to obtain characters for display, one may have dynamic RAM refreshing performed automatically by making sure that the required number of consecutive address locations (ROW Addresses) are accessed in the alloted time. Typically this is 128 ROW addresses in 2 ms.

For example, in a typical system we may have an 80 column by 25 row display with each row consisting of 10 scan lines. Each scan line has a period of 60.67 us. The vertical blank period consists of 25 scan lines for a total duration of 1.52 ms. Assuming sequential rather than table lookup operation, 80 consecutive character addresses are accessed each scan line and a 160 consecutive character addresses are accessed every 2 rows; more than enough to refresh all of the $\overline{RAS}$ rows. Of course one must be sure that the memory addresses of any two consecutive rows encompass all 128 possible $\overline{RAS}$ addresses. In the middle of the screen the worst case refresh period is 11 scan lines (.667 ms), since to do 160 consecutive addresses requires one complete row plus the first scan line of the next row. At the bottom of the screen the refresh period must also include the vertical blank time since no video characters are accessed then. In this case refresh stretches out to a worst case 2.184 ms.

Although in this example we exceeded the 2 ms refresh period, there are a number of things that we could do to get things back into spec. For example, we could cut down on vertical blank time, use memory chips with longer refresh periods, or have the CPU refresh video memory during vertical retrace. Taking the case of using different memory chips, another popular refreshing arrangement is 256 row addresses in 4 ms. In the middle of the screen this gives us a worst case period of 10 + 10 + 10 + 1 scan lines or 31 × 60.67 us = 1.88 ms. Adding in the vertical blanking period the absolute worst case refresh delay is 1.88 + 1.52 = 3.4 ms. Of course in this arrangement, making sure that any four consecutive rows encompass all 256 RAS addresses is much more difficult.

When operating in pixel mode meeting refresh requirements isn't as difficult since each scan line will access a different set of consecutive RAM addresses.

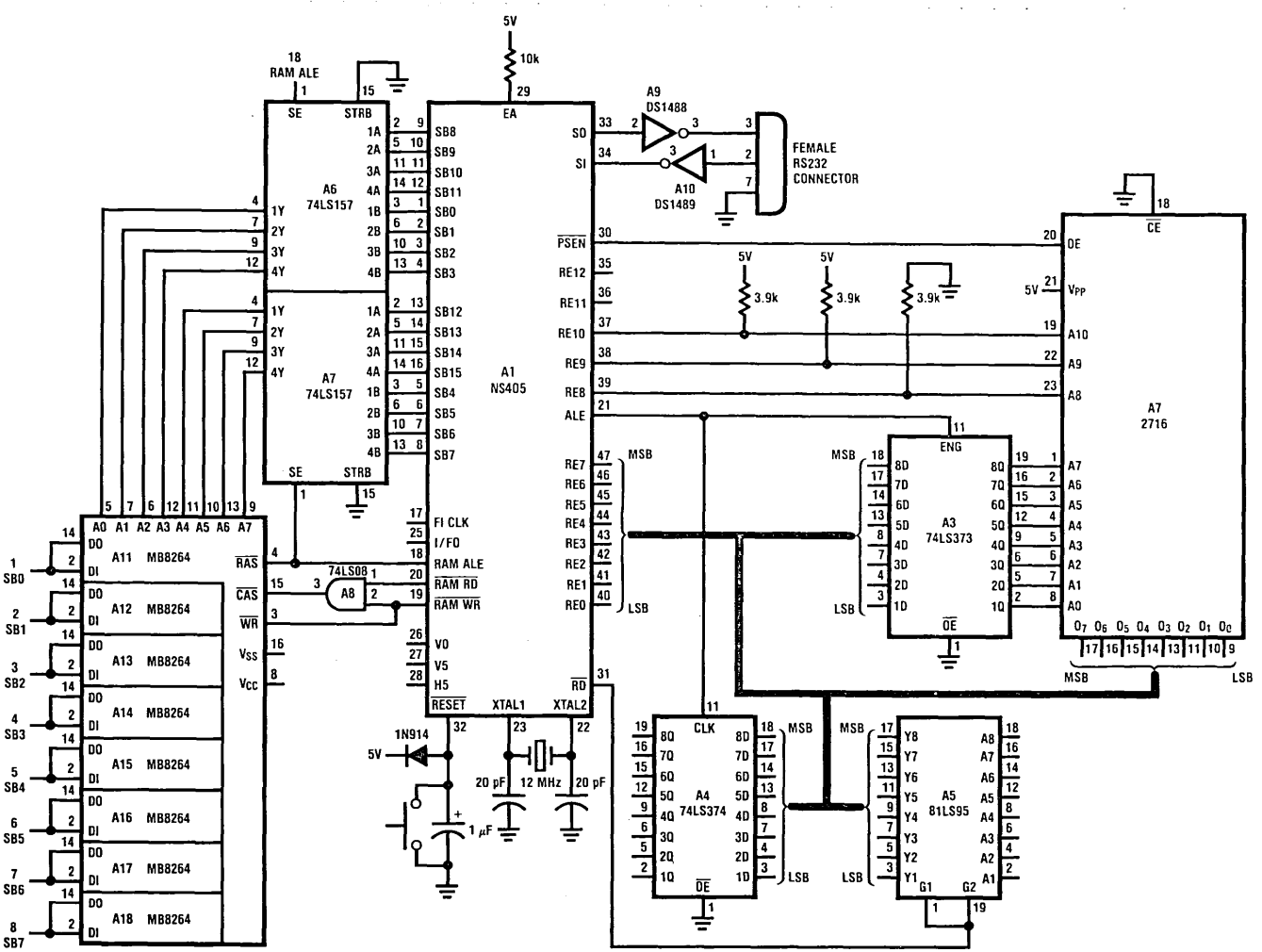
Returning to the circuit of *Figure 1*, we have assumed that SB0-7 are multiplexed address/data while SB8-15 output addresses only. Since the RAM addresses are latched in 8 bits at a time there is no need for a separate latch for SB0-7 since all 8 bits are clocked in on the falling ALE edge. However, when operating with smaller 8K or 16K RAMs where only 7 bits are clocked in at a time, latching arrangements for SB7 must be made. An example of this is shown in *Figure 3* where bits SB0-7 are all latched by the 74LS373.

Normally I/O registers, as well as other memory banks, will also be memory mapped into the 64K video RAM space. In order to do this some sort of chip enabling scheme must be worked out since the dynamic RAMs have no direct enable control. One possibility is shown in *Figure 4* where the RAM bank $\overline{CAS}$ and $\overline{WE}$ are disabled unless selected by the 74LS138 decoder. In this way the RAM output drivers will remain TRI-STATE® and no data will be written unless the bank is selected. However, memory refreshing as controlled by $\overline{RAS}$ will still be performed on each RAM bank.

By expanding on these basic examples a memory configuration for the TMP utilizing dynamic memories may be quickly and easily worked out.

TABLE 1. TMP Dynamic RAM Interface Timing 12 MHz CPU

Symbol	Parameter	Min	Typ	Max	Units
t_{AR}	Column Address Hold Time Referenced to $\overline{RAS}$	250	280		ns
t_{ASC}	Column Address Set Up Time–Dependent on Switching of 74LS157	25	35		ns
t_{ASR}	Row Address Set Up Time	20	90		ns
t_{CAC}	Access Time from $\overline{CAS}$		180	140	ns
t_{CAH}	Column Address Hold Time	140	250		ns
t_{CAS}	$\overline{CAS}$ Pulse Width	140	160		ns
t_{CP}	$\overline{CAS}$ Precharge Time	140	166		ns
t_{CRP}	$\overline{CAS}$ to $\overline{RAS}$ Precharge Time	100	136		ns
t_{CSH}	$\overline{CAS}$ Hold Time	250	280		ns
t_{CWL}	Write Command to $\overline{CAS}$ Lead Time	140	160		ns
t_{DH}	Data In Hold Time	160	175		ns
t_{DHR}	Data In Hold Time Referenced to $\overline{RAS}$	180	310		ns
t_{DS}	Data In Set Up Time	10	50		ns
t_{OFF}	Output Buffer Turn Off Delay	0		60	ns
t_{RAC}	Access Time from $\overline{RAS}$		220	150	ns
t_{RAH}	Row Address Hold Time–Dependent on Switching of 74LS157	10	15		ns
t_{RAS}	$\overline{RAS}$ Pulse Width	250	280		ns
t_{RC}	Random Read/Write Cycle Time	416			ns
t_{RCD}	$\overline{RAS}$ to $\overline{CAS}$ Delay Time	10	50		ns
t_{RCH}	Read Command Hold Time	100	175		ns
t_{RCS}	Read Command Set Up Time	100	175		ns
t_{RP}	$\overline{RAS}$ Precharge Time	100	135		ns
t_{RRH}	Read Command Hold Time Referenced to $\overline{RAS}$	100	175		ns
t_{RSH}	$\overline{RAS}$ Hold Time	140	160		ns
t_{RWL}	Write Command to $\overline{RAS}$ Lead Time	140	150		ns
t_{WCH}	Write Command Hold Time	140	150		ns
t_{WCR}	Write Command Hold Time Referenced to $\overline{RAS}$	160	275		ns
t_{WCS}	Write Command Set Up Time–Dependent on Delay of 74LS08	5	11		ns
t_{WP}	Write Command Pulse Width	140	150		ns

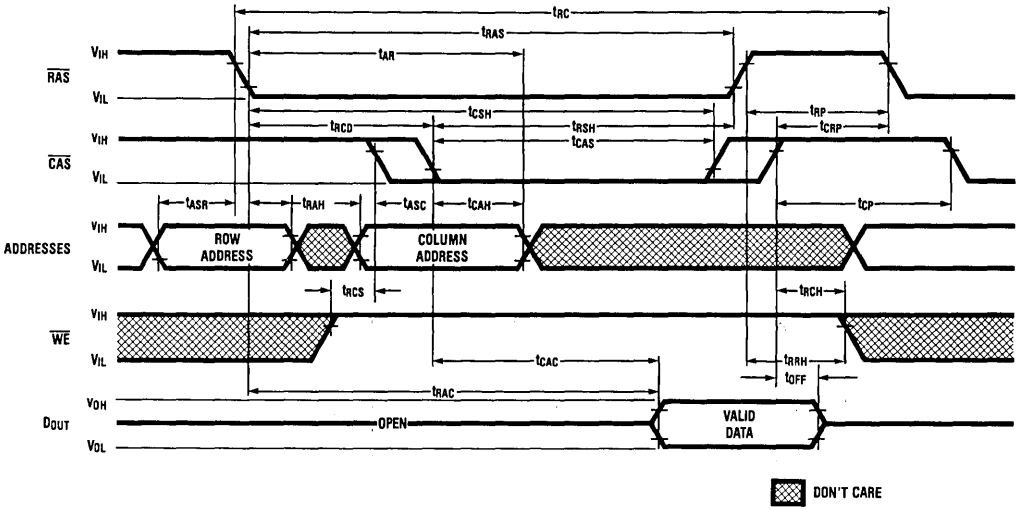


TL/C/5732–1

FIGURE 1. TMP with 64K Dynamic Memory

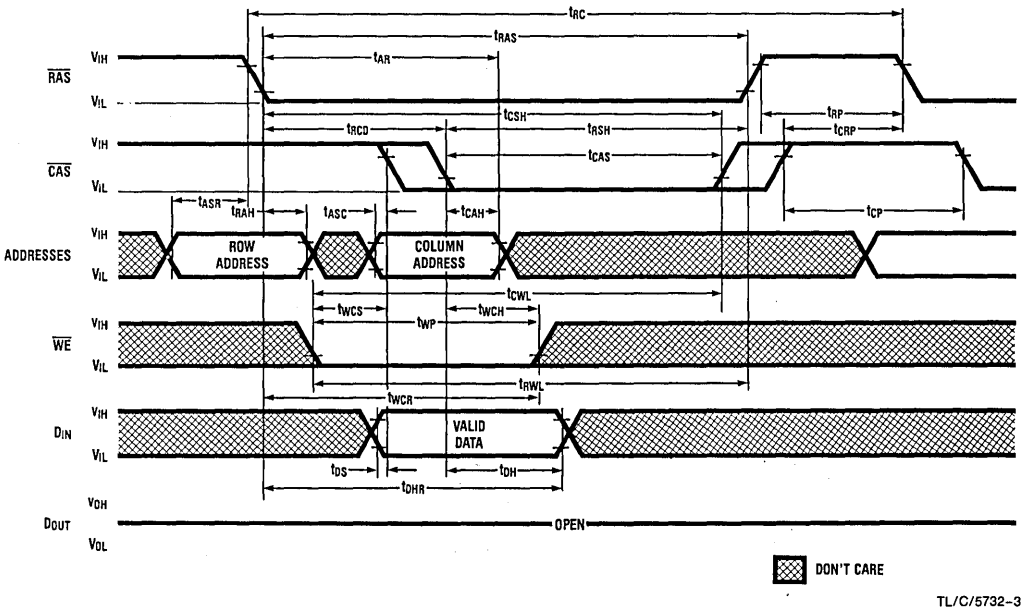
Timing Diagrams

Read Cycle Timing Diagram



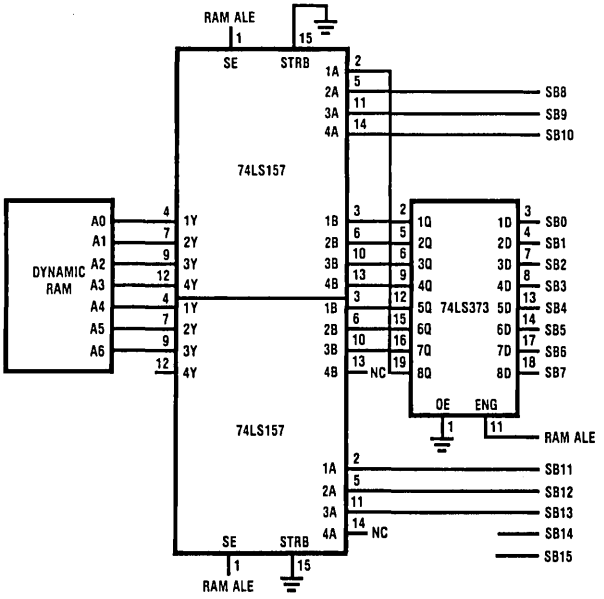
TL/C/5732-2

Write Cycle (Early Write)



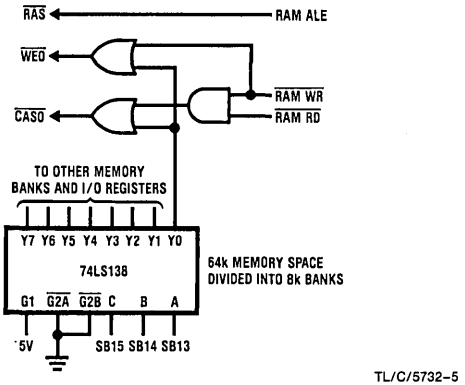
TL/C/5732-3

FIGURE 2.



TL/C/5732-4

FIGURE 3. TMP Address Multiplexing for 16K Dynamic RAMs



TL/C/5732-5

FIGURE 4. Chip Enabling Dynamic RAMs

TMP External Character Generation

National Semiconductor Corp.
Application Note 367
James Murashige

Built into the TMP video circuitry is the ability to access an external character generator to display custom FONT sets. In addition to the flexibility afforded by user selectable FONTs, by going "external" the number of different character patterns directly addressable is virtually limitless. On the other hand the disadvantages of going external are the additional hardware necessary to control data routing and the general need to use faster memories.

Figure 1 shows a minimum configuration with which to do external character generation. In the TMP, external character generation is selected through Video Control Register bits 6, 7 and is a cross between normal alphanumeric and pixel graphics display modes. Like normal alphanumeric mode the TMP sequences through the video memory address space based upon the screen format specification. But instead of routing the data through the internal character generator, it is treated as pixel data and directly inserted into the video dot stream. In effect what we are doing externally is duplicating the internal character generator ROM. In external mode video attributes are fully operational except for double height and block graphics.

Operation of the circuit shown is straight forward and follows a pipe-line approach. On a video data read the display memory address is output onto the system bus with the 8 low order bits being latched by the 74LS373. On the $\overline{\text{RAM}}$ $\overline{\text{RD}}$ signal the 2116 display RAM ouputs a data character onto the pipeline bus which is used to address the MM52116 character generator which in turn deposits the required pixel data onto the system bus so that it may be read in. The 2116 determines which character is to be looked up in the 52116 while the 74LS163 tells the character generator which row in the character we wish to look at. The 74LS163 is a counter which is appropriately clocked by the horizontal sync pulse so that we will advance each scan line to point to the next row in the character FONT. At the end of each screen row the counter must be cleared in preparation for the display of a new row. This is the function of the Scan Count Clear signal which is available as a multiplexed output on the RE11 pin. It is a low going signal which pulses for 1 scan line time during the last scan line in a screen ROW. Its timing is shown in Figure 2. Note that since the 74LS163 is a fully synchronous counter the clear input will not be accepted until the very last H-Sync clock pulse in the screen row. Because of the necessity to not clear the counter before all pixel data is brought in, nor to delay clocking lest the Scan Count Clear pulse be missed, the starting H-Sync clock edge must be postioned close to the start of horizontal blanking.

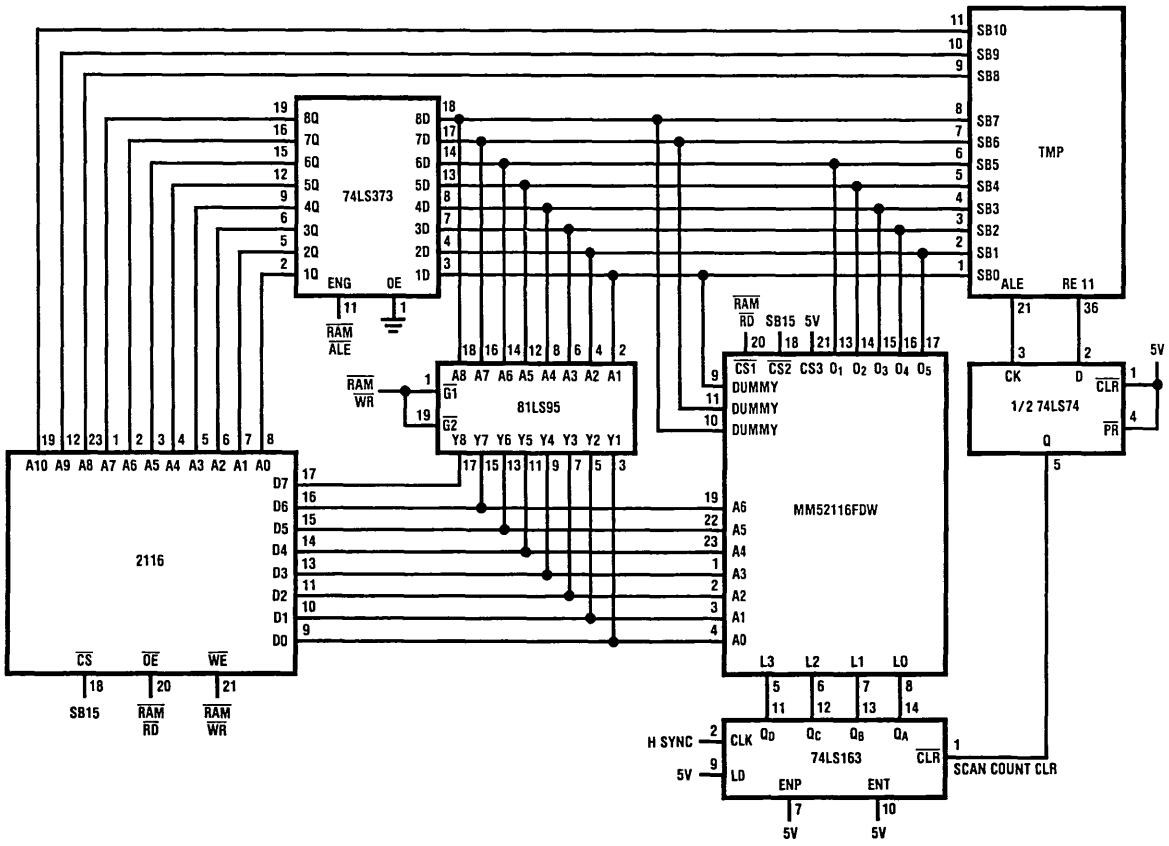
Continuing with the read operation, we see that video RAM is only accessed if SB15 is low, i.e., the lower 32K. Note that the 52116 used here contains 128 characters in a 5 × 7 FONT. Consequently, it has 5 data output lines connected to the system bus. The other three "dummy" lines shown connected are actually output bits which are always 0 by default, thus giving us blank spaces. There are two reasons why the character bits start on SB1. The first is that since everything brought in is considered pixel data, spaces between characters must be externally inserted. The second is that the video controller always brings in 8 bits even though the cell width can be defined to be 9 or 10. In these instances the 9th and 10th bits repeat what was encoded into the SB0 bit. As a result external characters can practically be at most 7 dots wide although the cells can be up to 8, 9, or 10 dots wide. Cell and/or character heights can be up to 16 lines tall as specified by the Character Scan Height Register.

On a video memory write, data is routed through the 81LS95, onto the pipe-line bus and into the 2116. Writing into the 2116 is controlled by $\overline{\text{RAM WR}}$ as shown. Ordinarily the MSB data bit is used for internal attribute latch selection and could be directly connected to the SB7 line if character cells were specified to at most be 7 dots wide. Otherwise SB7 will be needed for pixel generation as shown in Figure 1, thereby rendering internal attribute latch selection useless. In this case both internal attribute latches would have to be loaded with the same values. As shown here, 7 video RAM data bits are used to address the 128 possible characters in the 52116. If a larger character generator were available, additional data bits could be used to select from a larger character set. Since the TMP features a 16-bit multiplexed address/data bus, by using all 16 available data bits we could address 65,536 different character patterns!

With the video data pipe-lined as shown, very fast memory circuits are required for external character generation. With a 12 MHz CPU clock, character pixel data must be available within a max of 220ns (typ. 300ns) after an address goes out. To accomplish this the character generator will typically have to be bipolar and the video RAM fast MOS. However, if faster memories are a problem, access times may be stretched out by slowing down the CPU clock since video RAM cycling is based on the CPU clock. For instance with a CPU clock of 8 MHz, access time stretches out to 385 ns max, 500 ns typ. If using the divide by 1.5 factor on the crystal to obtain the slower CPU clock, remember that due to system constraints the character cell MUST BE AT LEAST 8 DOTS WIDE. In Figure 1 the 2116 output enable is shown being driven by RAM RD. Although this may seem redundant and will slow things down (why not just leave the output enabled?) it is necessary in order to avoid bus conflict when doing a memory write operation.

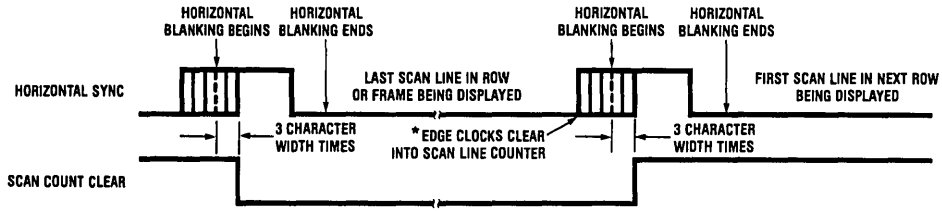
By expanding on this basic circuit, numerous options such as external attributes, expanded character sets and dynamic RAM may be added to achieve the desired end system.

TMP EXTERNAL CHARACTER GENERATOR



TL/C/5731–1

SCAN COUNT CLEAR TIMING



TL/C/5731–2

* Edge must come before Scan Count Clear goes away but not before the video controller has brought in all necessary display information for the last scan line Edge should not be more than 3 character widths from the beginning of blanking.

NS405 TMP Logic Analyzer

National Semiconductor Corp.
Application Note 369
James Murashige

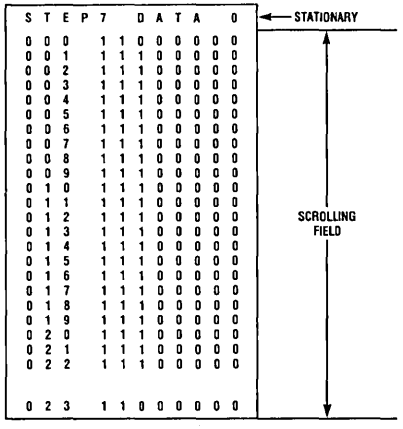
INTRODUCTION

The NS405 TMP is ideally suited for use in Test and Instrumentation equipment as the system or display controller. To demonstrate this, the following note describes how to turn the NS405 Demo Board into a simple 8 bit Logic State Analyzer. Featured in this system is a data capacity of 156 eight bit words, 21 μs data acquisition time, keyboard command entry, UP/DOWN rolling scroll and 24 line data display.

SYSTEM ARCHITECTURE

All of the necessary resources to build our system are available in a TMP Demo Board system when normally set up as a data terminal. Commands are entered through the attached ASCII encoded keyboard with data being strobed on the external interrupt. Data words are input through the switch configuration register SW2 by strobing the Light Pen interrupt. Video is output to the attached display monitor. The only real difference between our Logic Analyzer and the Data Terminal is the ROM software in U9 running the TMP. An overview of the system is shown in *Figure 1*.

In order to maximize the available 2k of video RAM, a display line length of 13 was chosen. This yields 157 lines of display information (157 × 13 = 2041), one of which is used to display title information. Thus our display data field consists of 156 lines of information, any 24 of which may be displayed at any given time. On each line is displayed the STEP number, followed by 2 spaces and 8 bits of 1 or 0 information. A typical display pattern is illustrated in *Figure 2*. By manipulating the pointer registers in the TMP DMA controller, the Title line is made to be stationary while the rest of the screen scrolls. This is accomplished by reversing the roles of the HOME register and Status Section SROW pointer. Specifically HOME points to the last row in memory which holds the title information while the status section is set to start after the display of the first row. Scrolling is accomplished by bumping the SROW pointer up or down 1 line width and checking for end of memory conditions.



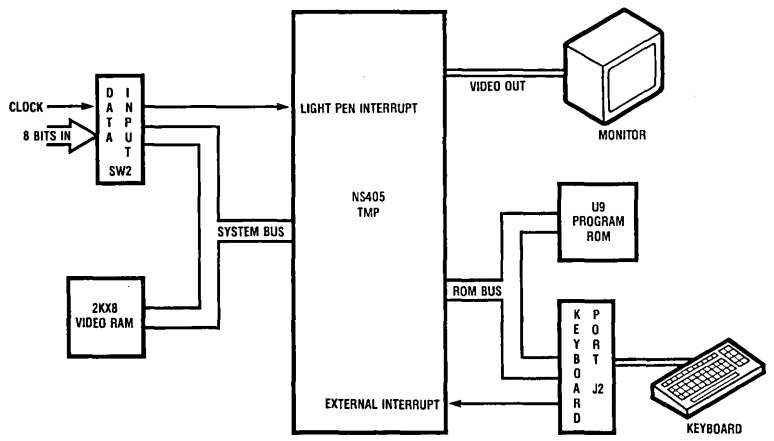
TL/DD/6970-2

FIGURE 2. TMP Logic Analyzer Screen Format

SYSTEM SOFTWARE

Since the system must rely on external events at several points before proceeding with processing, an interrupt driven approach was taken in structuring the software. A flowchart for the main program is shown in *Figure 3*. After system initialization there are 2 levels of processing associated with our logic analyzer operation. The first is a wait for an external interrupt signifying a new keyboard command. Referring to the keyboard service routine in *Figure 4*, the key is first read in and decoded as to function. In our simple system there are only 3 commands:

S or s = Start data acquisition
U or u = Scroll display up
I or i = Scroll display down



TL/DD/6970-1

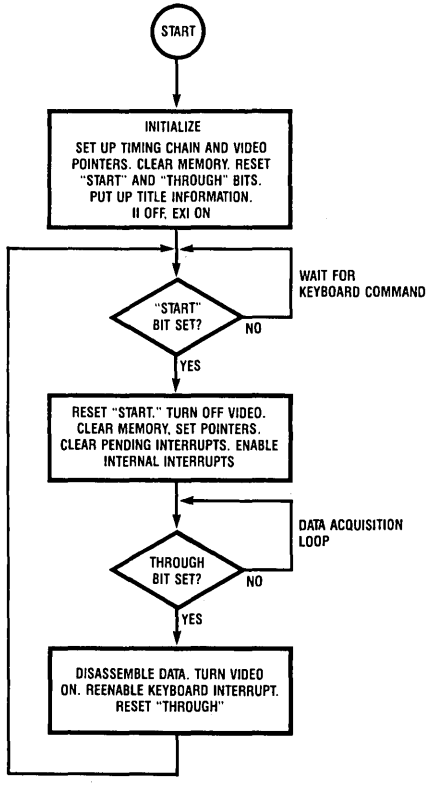
FIGURE 1. TMP Logic Analyzer (System Overview)

The scrolling functions are easily handled in the service routine by bumping the memory pointers and checking for an end of memory condition. A command to start data acquisition moves us to our second level of processing—the actual acquisition and display of data.

In both the keyboard and data acquisition interrupt service routines, flags F0 and F1 are used to pass system status back and forth from the main program. In this way the main program holds at major points while the service routines accomplish their functions. The data acquisition routine does nothing more than read data in from the SW2 port, store it in video memory and check a loop counter to see whether we have read in enough data. Since the Light Pen interrupt is being used, only high to low transitions will initiate an SW2 read. While very little is being done in data acquisition, it is time consuming because it's done in software. A count of instructions yields a worst case processing time of 21 μs between data strobes. In addition, since the data isn't latched it must remain stable until the actual read occurs. Following data acquisition, the stored data words are disassembled into their ASCII "1's" and "0's" patterns and the data entires numbered. With data acquistion completed, the program returns to await another keyboard command.

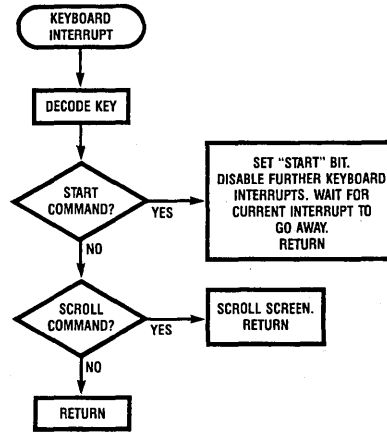
SUMMARY

As demonstrated, the NS405 is very effective as a display controller in a video instrumentation system. Certain functions, however, such as data acquisition are better left to dedicated hardware controllers. Nevertheless, the system as presented is still a very useful diagnostic tool. Through small enhancements to the hardware and software, features such as word recognition, number base conversion, wider data words and loop delay may readily be added.



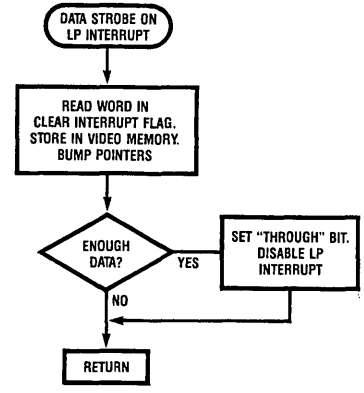
TL/DD/6970-3

FIGURE 3. TMP Demoboard Logic Analyzer Main Program



TL/DD/6970-4

FIGURE 4. TMP Demoboard Logic Analyzer Command Input Routine



TL/DD/6970-5

FIGURE 5. TMP Demoboard Logic Analyzer Data Acquisition Routine

```
1
2
3                .TITLE  MAIN,'TMP LOGIC ANALYZER DEMO'
4
5                ;                       James Murashige 2/09/84
6                ;This program turns the TMP Demo board into a simple 8 bit logic analyzer.
7                ;Command inputs are entered from the attached ASCII keyboard while data
8                ;acquisition takes place through the switch configuration socket, SW2.
9                ;The DIP switch may have to be unsoldered from the board.  Data is strobed
10               ;in with an external clock applied to Light Pen Interrupt on W8A.  Each time
11               ;data acqusition is started 156 words of 8 bits each are acquired and displayed.
12               ;Display is in the form of STEP location and the associated 8 bit 1's and 0's
13               ;pattern.
14               ;Commands are  S = Start data acqusition
15               ;              U = Scroll display up
16               ;              I = Scroll display down
17
18
19
20       07DF    LSTLIN = 07DF       ;START OF LAST LINE
21       07EB    MEMEND = 07EB       ;END OF MEMORY
22       07EC    STLIN  = 07EC       ;START OF TITLE LINE
23       0020    VON    = 020         ;VIDEO ON
24       0000    VOFF   = 000         ;VIDEO OFF
25
26
27
28       0000    . = 00              ;START AT PROGRAM LOCATION 0
29
30 0000 0452     RESET:  JMP BEGIN       ;VECTOR TO RESET CODE
31
32       0003    . = 03
33
34 0003 0412     EXI:    JMP KEY           ;VECTOR TO KEYBOARD COMMAND DECODE
35
36       0007    . = 07
37
38               INI:                  ;DATA STROBE INTERRUPT SERVICE
39 0007 8C               MOV A,INTR       ;CLEAR OUT INTERRUPT
40 0008 91               MOVX A,@R1       ;GET DATA CHARACTER
41 0009 80               MOVX @RO,A       ;STORE CHARACTER AWAY
42 000A 98               MOVL A,RO        ;BUMP RO POINTER
43 000B 6B               ADD A,R3
44 000C 88               MOVL RO,A
45 000D EA11             DJNZ R2,NOTRU    ;CHECK IF THROUGH
46
47 000F B5               CPL F1     ;YES THROUGH, SET INDICATOR BIT
48 0010 35               DIS II     ;DISABLE LP INTERRUPT
49
50 0011 93       NOTRU:  RETR        ;RETURN
51               .FORM
52               ;KEYBOARD COMMAND DECODE
53 0012 E1       KEY:     IN PORT          ;KEYBOARD DATA READ
54 0013 53DF             ANL A,#ODF       ;CONVERT LOWER TO UPPER CASE
55 0015 AA               MOV R2,A         ;SAVE COPY IN R2
56 0016 D353             XRL A,#'S'
57 0018 C627             JZ START         ;GOTO START
58 001A FA               MOV A,R2
59 001B D355             XRL A,#'U'
60 001D C62B             JZ UP       ;GOTO SCROLL UP
61 001F FA               MOV A,R2
62 0020 D349             XRL A,#'I'
63 0022 C63C             JZ DOWN  ;GOTO SCROLL DOWN
64 0024 A624     CKOFF:  JNXI CKOFF       ;NOT A VALID KEY & WAIT FOR EXI TO GO AWAY
65 0026 93               RETR         ;RETURN
66
67 0027 95       START:  CPL FO    ;START BIT SET
68 0028 15               DIS XI    ;DISABLE FURTHER KEYBOARD INTERRUPTS
69 0029 0424             JMP CKOFF
70
71 002B 99       UP:      MOVL A,R1        ;SCROLL UP
```

TL/DD/6970-6

7

```
 72 002C 030D          ADD A,#13       ;ADVANCE TO NEXT ROW
 73 002E 89            MOVL R1,A       ;SAVE NEW VALUE
 74 002F 0314          ADD A,#L(-L(STLIN))     ;CHECK FOR END OF DISPLAY
 75 0031 E2            MOV A,HACC      ;SUBTRACT STLIN FROM A
 76 0032 03F8          ADD A,#L(-H(STLIN) - 1) ;CARRY WILL BE SET IF A WAS > OR =
 77 0034 E639          JNC UPTRU               ;NEW VALUE OK, LOAD SROW AND RETURN
 78 0036 27            CLR A      ;RESET SROW TO BEGINNING
 79 0037 C2            MOV HACC,A
 80 0038 89            MOVL R1,A
 81 0039 99      UPTRU:  MOVL A,R1       ;LOAD R1 INTO SROW
 82 003A OE            MOV SROW,A
 83 003B 93            RETR
 84
 85 003C 99      DOWN:   MOVL A,R1       ;SCROLL DOWN
 86 003D 03F3          ADD A,#-13      ;SUBTRACT TO NEXT ROW
 87 003F AA            MOV R2,A        ;TEMP SAVE OF LOW ORDER
 88 0040 E2            MOV A,HACC      ;NOW DO UPPER HALF
 89 0041 03FF          ADD A,#OFF      ;CARRY WILL BE SET IF A WAS 12 OR MORE
 90 0043 F64D          JC DNTRU        ;NEW VALUE OK, LOAD VALUE INTO SROW
 91 0045 2307          MOV A,#H(LSTLIN)        ;RESET SROW TO LAST ROW
 92 0047 C2            MOV HACC,A
 93 0048 23DF          MOV A, #L(LSTLIN)
 94 004A 89            MOVL R1,A
 95 004B OE            MOV SROW,A
 96 004C 93            RETR
 97 004D C2      DNTRU:  MOV HACC,A
 98 004E FA            MOV A,R2
 99 004F 89            MOVL R1,A
100 0050 OE            MOV SROW,A
101 0051 93            RETR
102             .FORM
103
104             ;START OF INITIALIZING CODE
105
106 0052 C5     BEGIN:  SEL MBO
107 0053 C3             SEL RBO
108 0054 15             DIS XI    ;INTERRUPTS OFF FOR NOW
109 0055 35             DIS II
110 0056 65             STOP T    ;TIMER OFF
111 0057 27             CLR A      ;SET UP TIMING CHAIN FOR DEMO BOARD
112 0058 87             MOV TCP, A
113 0059 2367           MOV A, #103     ;HORIZONTAL LENGTH
114 005B B7             MOV @TCP, A
115 005C 230C           MOV A, #12      ;CHARACTERS/ROW
116 005E B7             MOV @TCP, A
117 005F 2353           MOV A, #83      ;HORIZONTAL SYNC BEGIN
118 0061 B7             MOV @TCP, A
119 0062 2363           MOV A, #99      ;HORIZONTAL SYNC END
120 0064 B7             MOV @TCP, A
121 0065 2391           MOV A, #091     ;CHARACTER HEIGHT/ EXTRA SCANS
122 0067 B7             MOV @TCP, A
123 0068 231A           MOV A, #26      ;VERTICAL LENGTH
124 006A B7             MOV @TCP, A
125 006B 2318           MOV A, #24      ;VERTICAL BLANK
126 006D B7             MOV @TCP, A
127 006E 2362           MOV A, #062     ;VERTICAL SYNC BEGIN/END
128 0070 B7             MOV @TCP, A
129 0071 2300           MOV A, #00      ;STATUS ROW BEGIN
130 0073 B7             MOV @TCP, A
131 0074 23F4           MOV A, #0F4     ;BLINK RATE
132 0076 B7             MOV @TCP, A
133 0077 2330           MOV A, #030     ;GRAPHICS COLUMN REGISTER
134 0079 B7             MOV @TCP, A
135 007A 2336           MOV A, #036     ;GRAPHICS ROW REGISTER
136 007C B7             MOV @TCP, A
137 007D 2389           MOV A, #089     ;UNDERLINE SIZE REGISTER
138 007F B7             MOV @TCP, A
139 0080 2309           MOV A, #009     ;CURSOR SIZE REGISTER
140 0082 B7             MOV @TCP,A
141
142
```

TL/DD/6970-7

```
143                 .FORM
144 0083 2304               MOV A, #004     ;SET SYSTEM CONTROL REGISTER
145 0085 55                 MOV SCR, A      ;8 BI, 7 DOTS, DIVIDE 1, SEQUENTIAL LOOKUP
146
147 0086 27                 CLR A       ;SET VIDEO RAM POINTERS
148 0087 C2                 MOV HACC, A     ;ACCUMULATOR CLEARED
149 0088 0D                 MOV BEGD, A
150 0089 0E                 MOV SROW, A     ;SROW WILL BE OUT HOME
151 008A 88                 MOVL RO, A      ;CLEAR MEMORY POINTER
152 008B 89                 MOVL R1, A      ;1 IS SROW IMAGE
153 008C 2307               MOV A, #H(MEMEND +1)
154 008E C2                 MOV HACC, A
155 008F 23EC               MOV A,#L(MEMEND +1)
156 0091 0C                 MOV ENDD,A      ;SET END OF MEMORY POINTER
157 0092 8A                 MOV HOME, A     ;SET POINTER TO TITLE ROW
158 0093 23FF               MOV A, #OFF
159 0095 C2                 MOV HACC,A
160 0096 8B                 MOV CURS, A     ;NO CURSOR
161 0097 3C                 MOV ALO, A      ;NO ATTRIBUTES FOR LATCH O
162 0098 3D                 MOV AL1, A      ;NO ATTRIBUTES FOR LATCH 1
163 0099 85                 CLR FO    ;FO IS "START" BIT
164 009A A5                 CLR F1    ;F1 IS "THROUGH" BIT
165 009B 2320               MOV A,#020
166 009D 82                 MOV MASK,A      ;SET INTERRUPT MASK
167
168 009E 3456               CALL MEMCLR     ;CLEAR VIDEO MEMORY
169
170 00A0 05                 EN XI       ;REENABLE EXTERNAL INTERRUPTS
171 00A1 2400               JMP LINNUM      ;DISPLAY TITLE INFORMATION
172
173 00A3 86A3       KEYIN:  JNFO KEYIN      ;WAIT FOR KEYBOARD INPUT
174
175                 .FORM
176
177                 ;DATA ACQUISITION ROUTINES
178
179 00A5 85                  CLR FO    ;CLEAR START BIT
180 00A6 2300               MOV A,#VOFF     ;VIDEO OFF
181 00A8 45                 MOV VCR,A
182 00A9 3456               CALL MEMCLR     ;CLEAR VIDEO MEMORY
183 00AB 27                 CLR A
184 00AC C2                 MOV HACC,A
185 00AD 0E                 MOV SROW,A      ;RESET SROW TO BEGINNING
186 00AE BA9C               MOV R2, #156    ;SET LOOP COUNTER FOR # WORDS TO READ
187 00B0 2305               MOV A,#5
188 00B2 88                 MOVL RO,A       ;SET MEMORY POINTER TO FIRST DATA DEPOSIT
189 00B3 23C0               MOV A,#0C0
190 00B5 C2                 MOV HACC,A
191 00B6 89                 MOVL R1,A       ;LOAD R1 WITH ADDRESS OF SWITCH REGISTER
192 00B7 BB0D               MOV R3,#13      ;LOAD R3 WITH POINTER BUMP CONSTANT
193 00B9 8C                 MOV A,INTR      ;CLEAR OUT ANY PENDING INTERRUPTS
194 00BA 25                 EN II       ;REENABLE LP INTERRUPT
195
196 00BB 66BB       DATAIN: JNF1 DATAIN     ;WAIT FOR "THROUGH" BIT TO SET
197
198                 ;DISASSEMBLE DATA INTO DISPLAY FORMAT
199 00BD A5                 CLR F1    ;RESET "THROUGH"
200 00BE BA9C               MOV R2,#156     ;DISASSEMBLE DATA, LOAD WORD COUNTER
201 00C0 27                 CLR A
202 00C1 C2                 MOV HACC,A
203 00C2 2305               MOV A,#5
204 00C4 88                 MOVL RO,A       ;SET MEMORY POINTER TO FIRST DATA DEPOSIT
205 00C5 BB08               MOV R3,#8       ;DO 8 BITS
206 00C7 90         UNASS:  MOVX A,@RO        ;LOAD IN DATA BYTE
207 00C8 04CB               JMP DEPST
208 00CA FC         RETRIV: MOV A,R4        ;RETRIEVE CHARACTER
209 00CB F7         DEPST:  RLC A     ;ROTATE BIT INTO CARRY
210 00CC AC                 MOV R4,A        ;TEMPORARY SAVE
211 00CD F6D3               JC ONE    ;STORE A "1"
212 00CF 2330               MOV A,#'0'      ;STORE A "0"
213 00D1 04D5               JMP CONT
```

TL/DD/6970–8

```
214 00D3 2331    ONE:    MOV A,#'1'      ;STORE A "1"
215 00D5 80      CONT:   MOVX @RO,A      ;STORE CHARACTER
216 00D6 38              INCL RO  ;INCREMENT POINTER
217 00D7 EBCA            DJNZ R3,RETRIV  ;CONTINUE IF WORD NOT DONE
218
219 00D9 98              MOVL A,RO       ;BUMP MEMORY POINTER TO NEXT WORD
220 00DA 0305            ADD A,#5
221 00DC 88              MOVL RO,A
222 00DD BB08            MOV R3,#8       ;RESET BIT COUNTER
223 00DF EAC7            DJNZ R2,UNASS   ;CONTINUE IF ALL LOCATIONS NOT DONE
224 00E1 2400            JMP LINNUM      ;JUMP TO NEXT PAGE
225              .FORM
226              ;LINE NUMBERING ROUTINES
227      0100    . = 0100
228
229 0100 27      LINNUM: CLR A
230 0101 C2              MOV HACC,A
231 0102 88              MOVL RO,A       ;CLEAR MEMORY POINTER
232 0103 BA9C            MOV R2,#156     ;DO 156 LINES
233 0105 BB0A            MOV R3,#10      ;SET HUNDREDS POINTER
234 0107 BC0A            MOV R4,#10      ;SET TENS POINTER
235 0109 BD0A            MOV R5,#10      ;SET ONES POINTER
236
237 010B FB      NUMLP:  MOV A,R3        ;LOOK UP HUNDREDS ASCII CODE
238 010C 343B            CALL LKUP
239 010E 80              MOVX @RO,A      ;STORE HUNDREDS ASCII CODE
240 010F 38              INCL RO
241 0110 FC              MOV A,R4        ;LOOK UP TENS ASCII CODE
242 0111 343B            CALL LKUP
243 0113 80              MOVX @RO,A      ;STORE TENS ASCII CODE
244 0114 38              INCL RO
245 0115 FD              MOV A,R5        ;LOOK UP ONES ASCII CODE
246 0116 343B            CALL LKUP
247 0118 80              MOVX @RO,A      ;STORE ONES ASCII CODE
248
249 0119 98              MOVL A,RO       ;BUMP RO TO NEXT LINE
250 011A 030B            ADD A,#11
251 011C 88              MOVL RO,A
252
253 011D ED26            DJNZ R5,CONNUM  ;INCREMENT ONES POINTER
254 011F BD0A            MOV R5,#10      ;MUST NOW INCREMENT TENS
255 0121 EC26            DJNZ R4,CONNUM  ;INCREMENT TENS POINTER
256 0123 BC0A            MOV R4,#10      ;MUST NOW INCREMENT HUNDREDS
257 0125 CB              DEC R3
258
259 0126 EA0B    CONNUM: DJNZ R2,NUMLP   ;DO ANOTHER ROW
260
261 0128 9A              MOV A,HOME      ;PUT UP TITLE LINE
262 0129 88              MOVL RO,A
263 012A BA0D            MOV R2,#13      ;LOOKUP 13 CHARACTERS
264 012C BB49            MOV R3,#L(TITLE)        ;LOAD POINTER TO ASCII TITLE STRING
265
266 012E FB      TITLP:  MOV A,R3
267 012F B3              MOVP A,@A
268 0130 80              MOVX @RO,A
269 0131 38              INCL RO
270 0132 1B              INC R3
271 0133 EA2E            DJNZ R2,TITLP
272
273              .FORM
274
275 0135 2320            MOV A,#VON      ;TURN VIDEO BACK ON
276 0137 45              MOV VCR,A
277 0138 05              EN XI      ;REENABLE KEYBOARD INTERRUPTS
278 0139 04A3            JMP KEYIN      ;RETURN AND WAIT FOR NEXT START
279
280              ;SUBROUTINES
281
282 013B 033E    LKUP:   ADD A,#L(CHAR)-1;
283 013D B3              MOVP A,@A       ;LOOK UP ASCII NUMBER
284 013E 93              RETR        ;RETURN
```

TL/DD/6970-9

```
285
286 013F 39       CHAR:   .BYTE '9876543210'
287
288 0149 53       TITLE:  .BYTE 'STEP 7 DATA O'
289
290 0156 27       MEMCLR: CLR A    ;VIDEO MEMORY CLEAR LOOP
291 0157 C2               MOV HACC,A       ;ACCUMULATOR CLEAR
292 0158 88               MOVL RO,A        ;RO CLEAR
293 0159 BA00             MOV R2,#0        ;INNER LOOP COUNTER SET
294 015B BB08             MOV R3, #8       ;OUTER LOOP COUNTER SET
295 015D 2320             MOV A,#020       ;SPACE CHARACTER
296 015F 80       MCLRLP: MOVX @RO,A       ;STORE A CHARACTER
297 0160 38               INCL RO  ;INCREMENT POINTER
298 0161 EA5F             DJNZ R2,MCLRLP  ;TEST INNER LOOP
299 0163 BA00             MOV R2,#0        ;RELOAD INNER LOOP COUNTER
300 0165 EB5F             DJNZ R3, MCLRLP ;TEST OUTER LOOP
301 0167 93               RETR         ;THROUGH, RETURN
302
303               .END
BEGIN   0052    CHAR    013F    CKOFF   0024    CONNUM  0126
CONT    00D5    DATAIN  00BB    DEPST   00CB    DNTRU   004D
DOWN    003C    EXI     0003 *  INI     0007 *  KEY     0012
KEYIN   00A3    LINNUM  0100    LKUP    013B    LSTLIN  07DF
MCLRLP  015F    MEMCLR  0156    MEMEND  07EB    NOTRU   0011
NUMLP   010B    ONE     00D3    RESET   0000 *  RETRIV  00CA
START   0027    STLIN   07EC    TITLE   0149    TITLP   012E
UNASS   00C7    UP      002B    UPTRU   0039    VOFF    0000
VON     0020

NO ERROR LINES

  328  ROM BYTES USED

SOURCE CHECKSUM = 40D8
OBJECT CHECKSUM = 0649

INPUT   FILE A:LOGIC.MAC
LISTING FILE A:LOGIC.PRN
OBJECT  FILE A:LOGIC.LM
```

TL/DD/6970-10

Building an Inexpensive but Powerful Color Terminal

National Semiconductor Corp.
Application Note 374
Leigh Cropper

Historically, the design of a color CRT terminal has involved a significant upgrade of the circuit for a monochrome terminal. The result was a stiff increase in price for the electronics as well as for the monitor when going from monochrome to color. As a result, most companies built monochrome terminals and a few built color terminals only.

On the personal computer front where separate monitors are common, manufacturers have started to offer video cards which will support either a monochrome or a color monitor. More recently, color terminals have begun to appear which are extensions of monochrome terminal families. They require a board full of I.C.s for even the most space efficient designs. But now, using the TMP, you can do the same job with just one VLSI chip and a half dozen 7400 family TTL chips.

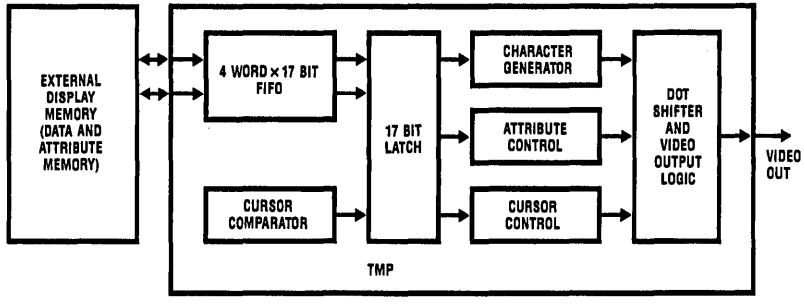
The National Semiconductor NS405 Series Terminal Management Processor (TMP) was originally conceived as a monochrome "terminal on a chip". However, the design team took special pains to build in "hooks" to allow users to augment the basic features of the TMP. In particular the TMP supports almost unlimited attribute expansion, and therein lies the key to adding color to a TMP-based terminal. Even nicer, the addition of color attributes does not sacrifice any of the other powerful features provided by the TMP.

Here, we will delve into a little of the mechanics of TMP attribute handling. The diagram in *Figure 1* shows the path of the attribute bits (normally 8) from the display memory into the TMP, through the FIFO, the attribute control logic, and finally to the video output section where the attributes are combined with the serialized video output.

Because the display memory space may be large (up to 64k x 16), it is easy to store many more attribute bits by adding display memory chips. A 2k x 8 RAM will hold 8 attribute bits for every location on an 80 row by 25 line display. However, in order to implement color attributes, three problems must be examined: (1) how to let both the CPU and the display controller address the extra attribute memory in a practical manner; (2) how to imitate the behavior of the internal FIFO and maintain proper synchronization; and (3) how to combine the color attributes and the video output signal.

Before addressing the three problems in detail, a discussion of the number and type of color attributes is in order. The simplest type of color display would require only 3 bits (red, green, and blue). That allows a character to be displayed in any of 7 colors over a black background or, when reverse video is asserted, the character is black on a colored background. For independent control of both the foreground and background colors, 6 bits are required. To get more shades of color, add more bits.

A practical approach employs a 2k x 8 RAM for the color attribute memory. Three of the bits control the foreground color, three control the background color and the remaining two may be used to adjust intensity (1 for foreground and 1 for background).



TL/DD/7923-1

FIGURE 1. TMP Attribute Processing

Now let's tackle the problems one by one.

1. COLOR ATTRIBUTE MEMORY ADDRESSING. When fetching data for the display, we need to get 24 bits in parallel (8 data, 8 attribute and 8 color attribute). But when the CPU accesses memory, it can handle only 16 bits at a time, so the CPU must be able to read and write color attributes in a different bank of memory from that where the data and ordinary attributes are stored. For an 80 character by 25 row display the memory could be mapped as shown in *Figures 2* and *3:*

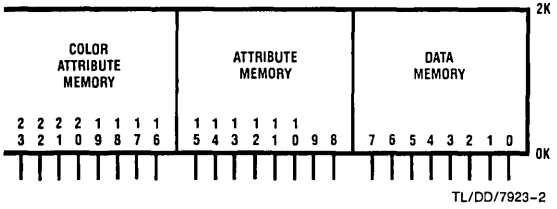


FIGURE 2. Memory Map as Selected for Screen Refresh

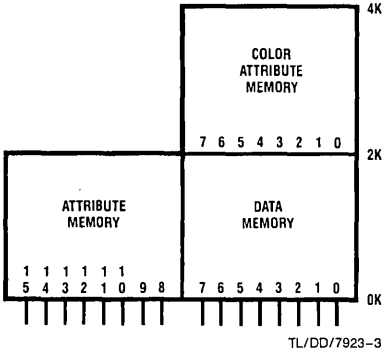


FIGURE 3. Memory Map as Selected for CPU Access

The mapping is implemented by the following circuit:

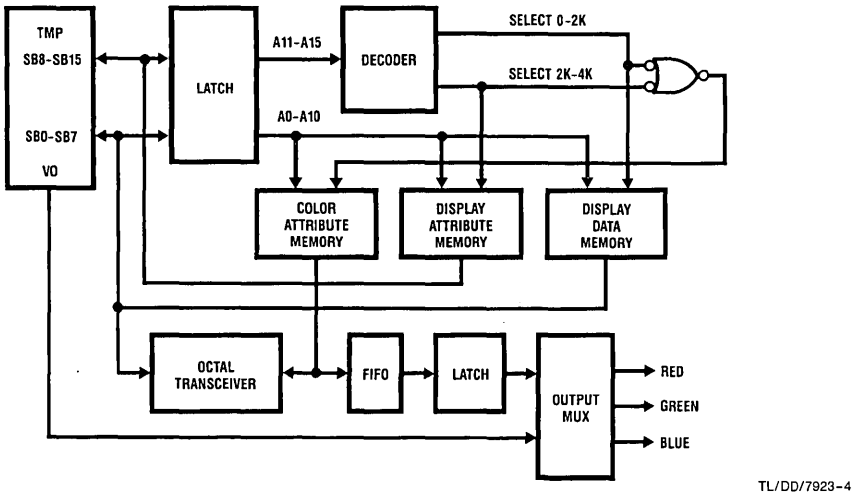


FIGURE 4. Color Attribute Memory Mapping Circuit

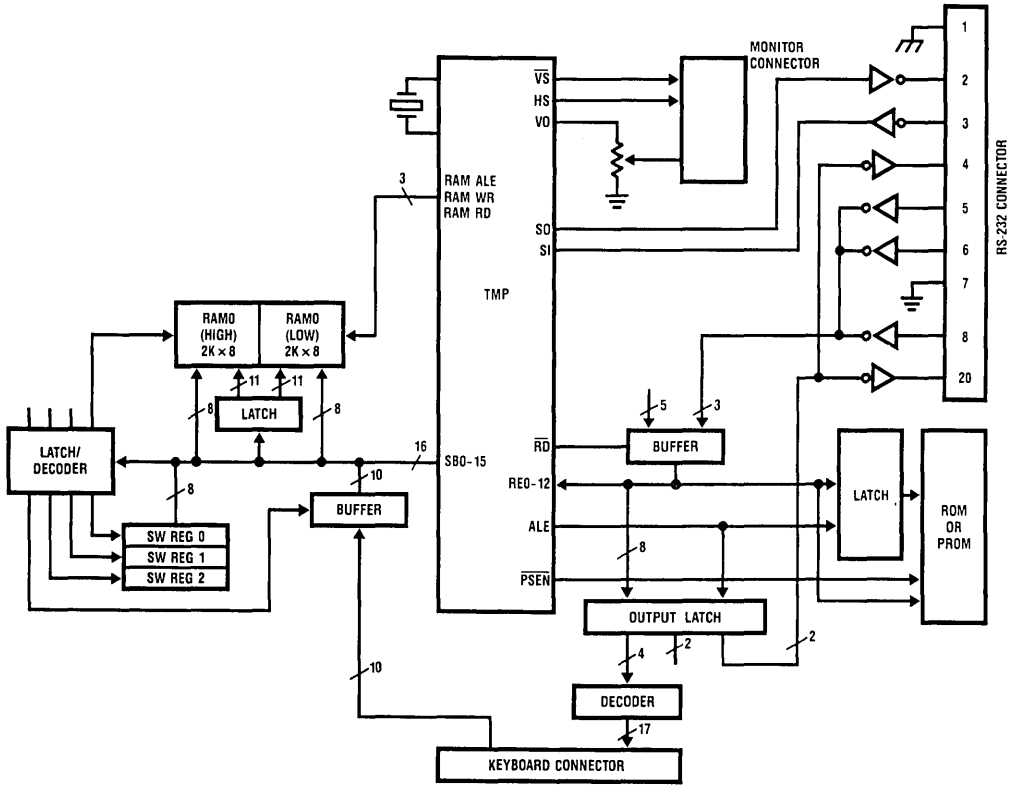
During a display refresh cycle the color attribute memory is selected by the low bank select (the same select signal that enables the data and attribute memories). However, the color attribute bits drive the external FIFO's, whereas the output from the other two memories is routed through the TMP. The data path from the color attribute memory to the TMP is buffered by an octal transceiver which is disabled when the low bank is selected. During a CPU access to color attribute memory, the high bank select enables the color attribute memory and the octal transceiver. The direction control of the transceiver is controlled by the RAM RD signal from the TMP.

2. EXTERNAL FIFO SYNCHRONIZATION. The TMP provides FI CLK (FIFO Input Clock) and FO CLK (FIFO Output clock) signals which may be used to clock an external FIFO. The FI CLK signal is identical in timing and duty cycle to the RAM RD signal except that FI CLK is disabled (stays high) when the CPU accesses display memory. When the 74LS224 is used as an external FIFO, FI CLK must be inverted. The rising edge of FO CLK occurs when output of the internal FIFO is loaded into the internal dot shifter. The FO CLK is used to empty a word from the external FIFO and clock it into an octal latch.
3. COMBINING COLOR ATTRIBUTES WITH VIDEO. When using foreground and background color attributes, a 74LS157 multiplexer works nicely to switch between the two. The eight color attribute bits from the latch are separated into groups of four. The video output signal is used to switch the multiplexer. When the video output is high, foreground attributes are selected: and when the video output is low, background attributes are selected. The outputs of the multiplexer (red, green, blue and intensity) directly drive the color monitor inputs. A minor problem arises because the video output from the TMP already includes the blanking signal. That makes it impossible to differentiate between a series of spaces in the middle of the screen and the horizontal blanking interval. In either case, the video output is low. The easiest solution is handled in software. Let's assume that we want an 80 column display and are using three 2K x 8 memory chips for the data, attribute and color memories. We set up the TMP for 81 columns and then configure the program so that the 81st column always contains a space code with all attribute bits off (including color). That way the background color will always be black during both horizontal and vertical retrace. The cost is 25 locations in each of the memories, but we can afford that many because an 80 x 25 display requires 2000 locations, leaving 48 free.

A Practical Example

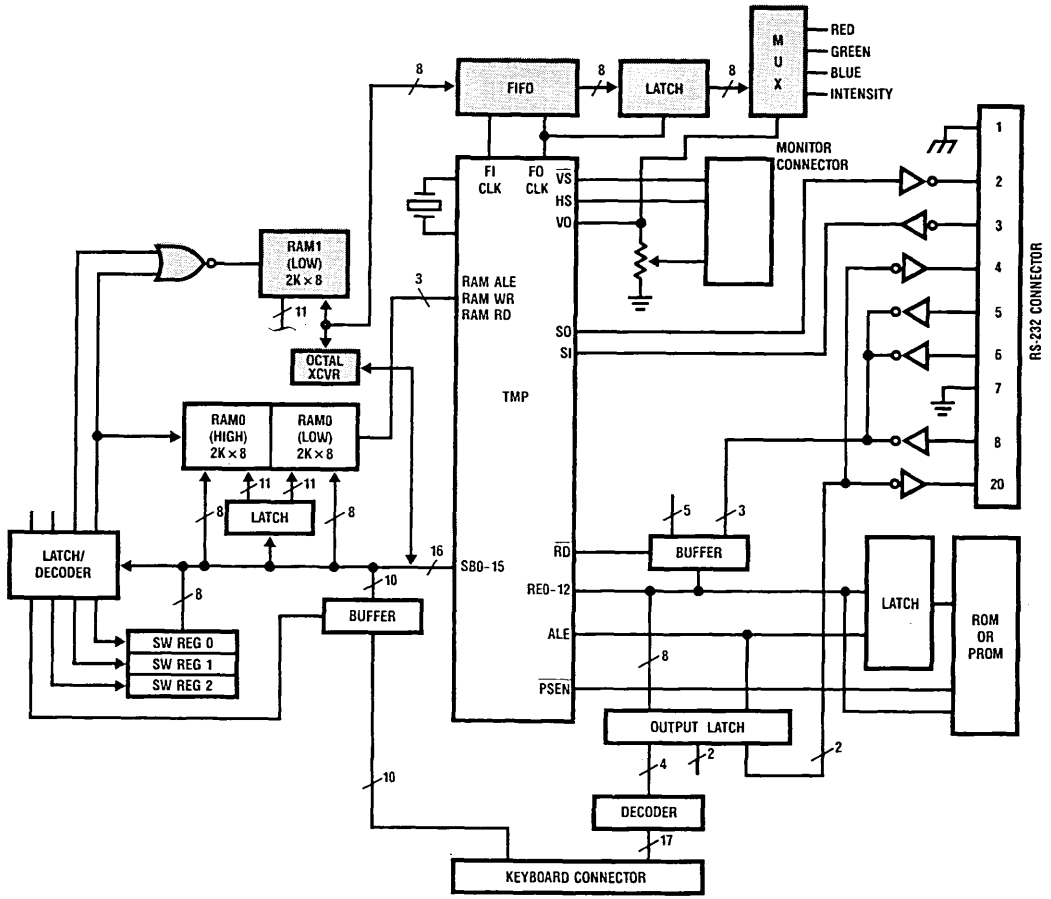
Here we will present a color terminal circuit with the associated program as an example of what you may want to do. We started with the terminal design of the TMP development board. See the block diagram in *Figure 5*.

The block diagram of the color terminal (with the old portions of the original monochrome terminal unshaded and the new color circuits shaded) appears in *Figure 6*. The new circuitry was added in the prototyping area of the development board.



TL/DD/7923-5

FIGURE 5. TMP Development Board Block Diagram



TL/DD/7923-6

FIGURE 6. TMP Development Board Color Circuitry Block Diagram

COLOR ATTRIBUTE BIT ASSIGNMENTS

The bit assignments are:

Bit 0 — Blue foreground
Bit 1 — Green foreground
Bit 2 — Red foreground
Bit 3 — Blue background
Bit 4 — Green background
Bit 5 — Red background
Bit 6 — Foreground intensity
Bit 7 — Background intensity

Without using the intensity control bits you get 8 foreground colors: red, green, blue, magenta, cyan, yellow, white, and black (beam off). The same 8 colors may be independently selected for the background. There are several RGB monitors available in the moderate price range with sufficient bandwidth to work with a 12 MHz TMP. Some of them include a separate intensity (or luminence) input. Others include internal decoding circuitry which provides the ability to handle 4 bits of color input and provide as many as 16 different colors.

The demonstration program which runs on the development board allows limited color support. The Escape, V sequence from the keyboard or the receiver prompts the program to treat the next character received as an eight bit color attribute byte with the bit assignments as listed above. That byte is written to the color attribute memory as each succeeding character is received, until another escape, V sequence is encountered. The table which follows includes the foreground and background color combinations for characters which can be entered from the keyboard, but it ignores the effect of the 2 high-order bits (foreground and background intensity).

COLOR COMBINATIONS FOR RGB MONITORS

Table I gives the Foreground/Background color combinations that occur when using the '<ESC> Vv' Escape sequence. To set the current color attribute, all that you need to do is select the color combination from the Table below, and send it to the NS405 as part of the <ESC> Vx sequence. For example, '<ESC> V"' causes the Foreground color to be green and the Background color to be red . . . not all that pleasing, to my tastes, but choose what you will.

TABLE I
Foreground/Background Color Combinations

Char	Fore/Back	Char	Fore/Back	Char	Fore/Back
sp	Black/Red	6	Yellow/Yellow	K	Cyan/Blue
!	Blue/Red	7	White/Yellow	L	Red/Blue
"	Green/Red	8	Black/White	M	Magenta/Blue
#	Cyan/Red	9	Blue/White	N	Yellow/Blue
$	Red/Red	:	Green/White	O	White/Blue
%	Magenta/Red	;	Cyan/White	P	Black/Green
&	Yellow/Red	<	Red/White	Q	Blue/Green
'	White/Red	=	Magenta/White	R	Green/Green
(	Black/Magenta	>	Yellow/White	S	Cyan/Green
)	Blue/Magenta	?	White/White	T	Red/Green
*	Green/Magenta	@	Black/Black	U	Magenta/Green
+	Cyan/Magenta	A	Blue/Black	V	Yellow/Green
,	Red/Magenta	B	Green/Black	W	White/Green
-	Magenta/Magenta	C	Cyan/Black	X	Black/Cyan
.	Yellow/Magenta	D	Red/Black	Y	Blue/Cyan
/	White/Magenta	E	Magenta/Black	Z	Green/Cyan
0	Black/Yellow	F	Yellow/Black	[	Cyan/Cyan
1	Blue/Yellow	G	White/Blue	\	Red/Cyan
2	Green/Yellow	H	Black/Blue	]	Magenta/Cyan
3	Cyan/Yellow	I	Blue/Blue	^	Yellow/Cyan
4	Red/Yellow	J	Green/Blue	-	White/Cyan
5	Magenta/Yellow				

TMP Extended Program Memory Application Note

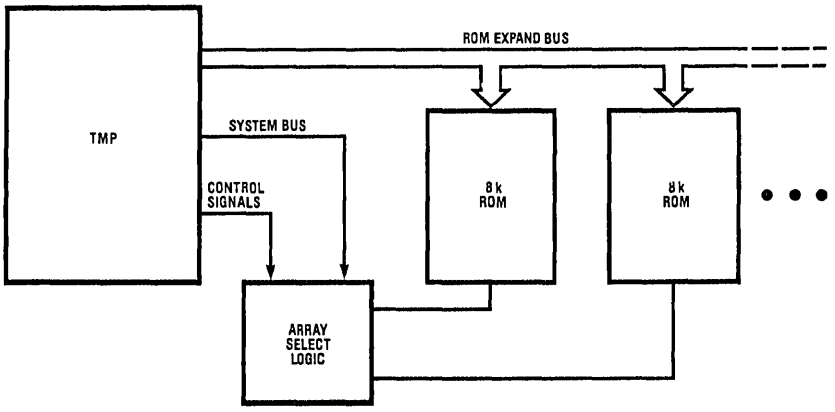
National Semiconductor Corp.
Application Note 399
Richard Lazovick

OVERVIEW/INTRODUCTION

The purpose of this application note is to describe methods for expanding the program memory of the NS405 series TERMINAL MANAGEMENT PROCESSOR (TMP) and to provide direction in software techniques for utilizing the expanded memory efficiently. The chip has a built-in capability of addressing up to 8k of external program memory (ROM), via the ROM Expand Bus, and 64k of video display memory (RAM), via the System Bus. Although 8k of program memory is sufficient for most applications there are many applications, such as emulating multiple terminals or using many look-up tables, that require still more memory. However, it is very rare that the entire 64k of video RAM is used since that is more than enough memory to store two screens of data in the pixel mode or thirty-two screens of data in the alphanumeric/block graphics mode. Therefore it is practical to use a video memory address to switch between two or more 8k memory arrays.

The idea behind using a bank select switch to change from one memory array to another is not new, nor is it difficult, and when implemented properly it can be a very useful tool. The TMP has all the necessary control signals to make both the software and hardware straight-forward.

Block Diagram



TL/DD/8430-1

SOFTWARE

For purposes of demonstration it will be easier to look at the software aspects of using an array select switch first, then designing the hardware to implement it.

The easiest case occurs if we use less than 16k of display memory. Then we have two system bus address lines available to select either of our two arrays. To switch arrays all we have to do is read from (or write to) an address that uses the address line you wish to toggle. It is safer to read from the address since we do not want to change data in memory at the location addressed by the lower order address lines.

Suppose we choose SB14 to select the low order array and SB15 to select the high order array. The program steps we would go through to switch from the low array to the high array could be:

```
MOV   A,#080        ;Load HACC w/ 80H to set SB15 HI
MOV   HACC,A        ;and set SB14 LO. We do not care
MOVL  R0,A          ;about the other address bits.
MOVX  A,@R0         ;SB15 goes HI.
```

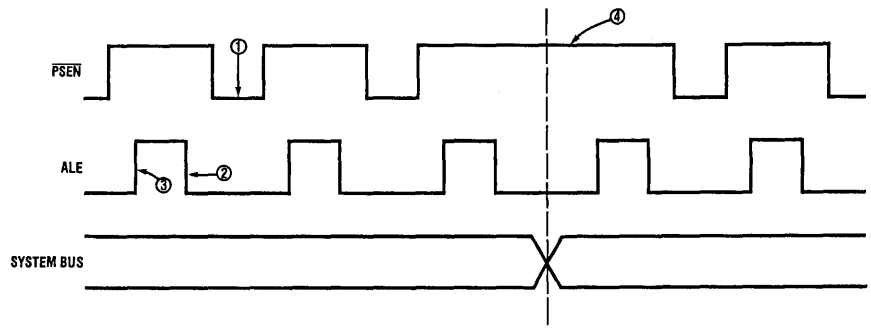
In general we will want to switch arrays several times, and we will want to be able to conveniently control the destination address in the new array.

Since it is very cumbersome to rewrite the whole sequence everytime, let's mimic the internal select memory bank command (SEL MBx) by using a subroutine and a CALL followed by a JMP to conveniently control our array switching.

```
        CALL  SELHA       ;Select HI order array.
        JMP   HERE        ;Jump to HERE in new array.

SELHA:  MOV   A,#080      ;Load HACC w/ 80H to set SB15 HI
        MOV   HACC,A      ;and set SB14 LO. We do not care
        MOVL  R0,A        ;about the other address bits.
        MOVX  A,@R0       ;SB15 goes HI.
        RET               ;Return to execute the jump.
```

Now each time we switch to the high order array all we have to do is execute a CALL and a JMP.

System Signals Timing Diagram



Note 1: Enable ROM output drivers.

TL/DD/8430-2

Note 2: ROM address available.

Note 3: RE bus addresses changes during rising edge of ALE and are stable by falling edge.

Note 4: No $\overline{\text{PSEN}}$ signal present during last cycle of MOVX instruction, however $\overline{\text{PSEN}}$ is active during both RET cycles.

HARDWARE

Now that we have made the software simple and straightforward we have to look at what hardware is necessary to implement it.

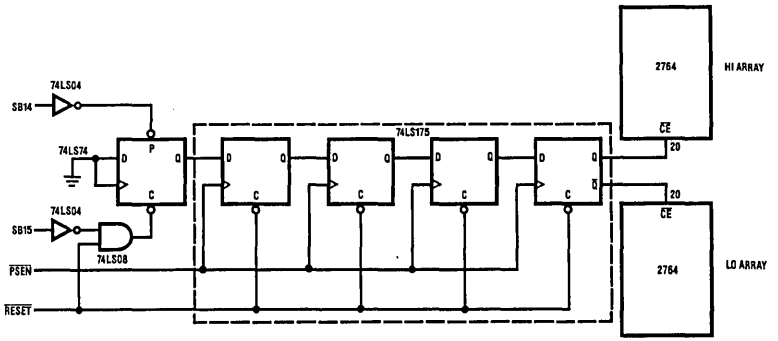
We want to: 1) create two mutually exclusive enable signals—one for each array,
2) be able to easily use and latch the address line signals, and
3) delay the actual switching of arrays until after the jump instruction, with the new address, is read into the TMP from the old array.

Looking at the program we see that the system bus changes after the MOVX instruction with the RET and JMP instructions still to be read in from memory before we actually want to switch arrays. Each instruction takes two cycles, therefore we want to delay our array switching signal by four cycles. Looking at all the output signals on the TMP there are two possible signals to use as a clock to delay the array switching signal. These signals are $\overline{PSEN}$ and ALE (see System Signals Timing Diagram).

The main disadvantage of using ALE is that whereas we want a rising edge to clock the flip-flops used for the delay, the ROM addresses are not stable until the falling edge of ALE. Therefore, we save one inverter by using $\overline{PSEN}$.

One possible circuit implementation is shown below:

Circuit Diagram



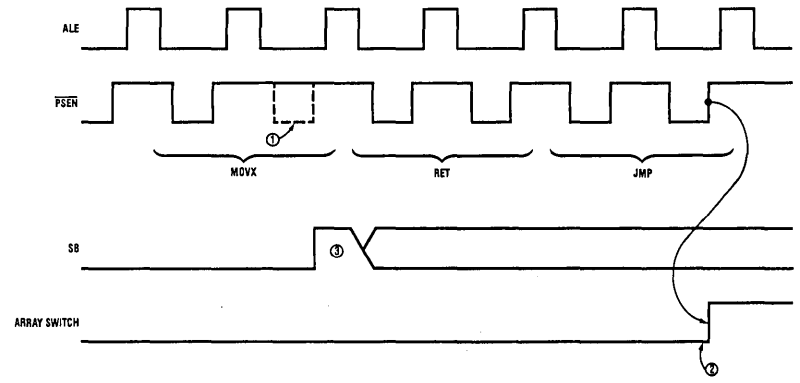
TL/DD/8430-3

The first flip-flop latches one of the two system bus signals and the next four delay the array switching signal by four $\overline{PSEN}$ cycles. The two inverters are there so that we trigger off a ONE on the address. If the system bus was configured as a 16 bit address/data bus (bit 4 of SCR set) then the latched address lines would have to be used. Since it is always desirable to have the flip-flops in a known state at power up, some sort of reset circuitry should be used (e.g., by tying the power up reset circuit to the clear inputs on the flip-flops), or both arrays should have identical reset sequences that include setting the flip-flops to a known state.

LOOKING IN DEPTH

Now that we have the basic software format we are going to use and the hardware to implement it, let's look at what is actually happening (see Array Switching Timing Diagram 1). The system bus line switches after the first cycle of the MOVX instruction, but there is no $\overline{PSEN}$ during the second cycle. Then there are two $\overline{PSEN}$ signals during the RET instruction and two $\overline{PSEN}$ signals during the JMP instruction. Just after the second byte of the JMP is read into the TMP, the arrays are switched, the PC gets loaded with the new address, and the program continues execution as normal in the new array from the address indicated in the JMP instruction. Be sure you understand the Array Switching Timing Diagram 1 before continuing.

Array Switching Timing Diagram 1



TL/DD/8430-4

Note 1: No $\overline{PSEN}$ signal.
Note 2: Arrays switch here.
Note 3: Valid approximately 360 ns.

ADDENDUMS

Although it can be a problem when trying to execute a call across array boundaries, the problem can be easily overcome as can the confusion that arises when many array switchings occur. All that one needs to do is to organize the program memory efficiently. One such scheme would be to set aside a block of memory in each bank, such as the last page to use for memory mapping. For example if we wanted to jump from location HOME in the LO array to location HERE in the HI array and then back to HOME we could map our memory as shown below:

Lo Array

MAIN PROGRAM:

```
                .
                .
                .
0500  HOME:  JMP   HERE
                .
                .
                .
```

ARRAY SWITCHING SUBROUTINE:

```
0700  SELHA: MOV   A,#080
0702         MOV   HACC,A
0703         MOVL, R0,A
0704         MOVX  A,@R0
0705         NOP
0706         NOP
0707         RET
```

MEMORY MAP:

```
070A  HERE:  CALL  SELHA
070C         JMP   HOME
070E        (CALL SELHA)
0710        (JMP addr)
                .
                .
                .
```

Hi Array

```
                .
                .
                .
0680  HERE:  NOP
0681         JMP   HOME
                .
                .
                .
```

```
0700  SELLA: MOV  A,#040
0702         MOV  HACC,A
0703         MOVL R0,A
0704         MOVX A,R0
0705         NOP
0706         NOP
0707         RET
```

```
070A  HOME:  CALL SELLA
070C         JMP  HERE
070E        (CALL SELHA)
0710        (JMP addr)
                .
                .
                .
```

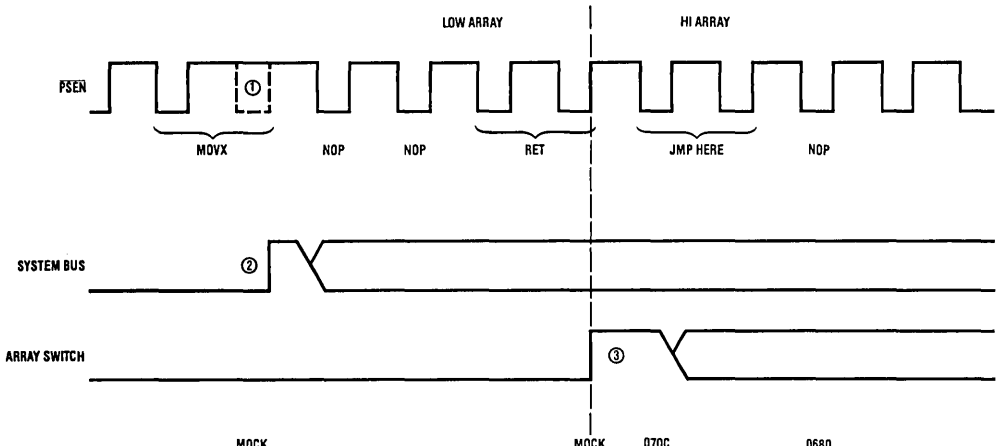
TL/DD/8430-6

Note: The arrows show which JMP corresponds to which subroutine call. For example the JMP HOME at location 070C in the LO ARRAY corresponds to the CALL at location 070A in the HI ARRAY. The ()'s show how the CALL's and JMP's can be strung together in a neat pattern.

Notice that there are two NOP's in the subroutine. Since the JMP after the CALL was moved to the new array the two NOP's were added to the subroutine so that the actual array switching occurs just after the completion of the RET instruction. The PC is then loaded with the new jump value, loaded in from the new array, and we continue execution as expected. Be sure to understand the timing before going any further (see Array Switching Timing Diagram 2). The way the memory map is set up it is easy to organize and keep track of jumps using the pattern indicated in the example. This method also eliminates any problems with the assembler searching for undefined labels.

ADDENDUMS (Continued)

Array Switching Timing Diagram 2



TL/DD/8430–5

Note 1: Missing $\overline{\text{PSEN}}$ signal.
Note 2: System bus changes.
Note 3: Arrays switch.
Note 4: Now in new array.

Since there are two extra NOP's in the switching array subroutine the hardware can now be simplified and the system speed increased by removing two of the flip-flops from the chain. Through the use of the SEL MBx commands the memory map can be located in any page of any memory bank. For example if we wanted to jump to location HERE in memory bank 2 of the HI array from memory bank 1 of the LO array (after having removed two flip-flops) we could map our memory as shown below:

Lo Array

ARRAY SWITCHING SUBROUTINE:

```
0700  SELHA:  MOV   A,#080
0702          MOV   HACC,A
0703          MOVL  R0,A
0704          MOVX  A,@R0
0705          RET
```

MEMORY MAP:

```
070A  HERE:   CALL  SELHA
070C
070E
```

MAIN PROGRAM:

```
0800  HOME:   SEL   MB0
0801          JMP   HERE
```

Hi Array

```
0700  SELLA:  MOV   A,#040
0702          MOV   HACC,A
0703          MOVL  R0,A
0704          MOVX  A,@R0
0705          RET
```

```
070A
070C          SEL   MB2
070E          JMP   HERE
```

```
1000  HERE:   NOP
```

If a call into the other array is necessary a similar pattern to that above could be used. Start by replacing the JMP's with CALL's to the desired subroutine and appropriately placing returns. For example (here it comes) if we wanted to CALL HOME from HERE we could memory map as shown below.

Lo Array	Hi Array

MAIN PROGRAM:

```
HOME:  NOP
       RET
```

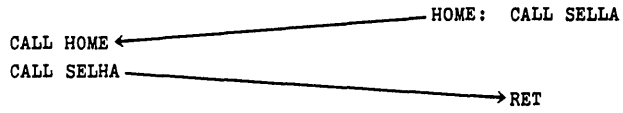
```
HERE:  NOP
       CALL HOME
```

ARRAY SWITCHING SUBROUTINE:

```
SELHA:  .
        .
        .
        RET
```

```
SELLA:  .
        .
        .
        RET
```

MEMORY MAP:



TL/DD/8430-7

Since calling between different memory banks is not straight forward it is advisable to be very careful when doing it, or to limit calls between arrays only to those that reside in the same memory bank.

HELPFUL HINTS

These schemes can all be modified to multiple arrays and easier or fancier mappings, however there are a few things to keep in mind.

1) If using a system bus address line to toggle the array, don't use that line as part of an actual display memory address.
2) The MOVX instruction can require more than two cycles depending on system bus contention, however we are only concerned with the last two cycles and the $\overline{\text{PSEN}}$ signals that occur after the system bus line changes.
3) If using interrupts—disable them while switching arrays and keep all time critical routines in the same array.
4) A demux or decoder can be used to select memory arrays or decode address lines when more than two 8k arrays are implemented or more than 16k of video RAM is being used.
5) If extra memory is needed, but a good deal of the program memory is data storage, the data could be stored in the video memory space instead of implementing a new array.
6) If the TMP is going to be used in a noisy environment or the system bus is configured as a 16 bit bidirectional bus a synchronous latch should be used to assure stable levels on SB14 and SB15.
7) The given array switching circuit can be implemented with the demo board by wiring it into an extension board that can be plugged into the prom socket U9. Wire the two new proms in parallel with each other and with a cable that can plug directly into the prom socket. However, instead of using pin 20 from the demo board, use the two array enable lines as the chip enables for the 2764 proms.

   Also use SB12 and SB13 instead of SB14 and SB15.

Section 8

Microcontroller Development Tools

Section 8 Contents

MOLE™

The MOLE (Microcontroller On-Line Emulator) is a low cost development system and emulator for all National microcontroller products.

The MOLE system was developed for the microcontroller design engineer and provides a method of writing, debugging and emulating microcontroller software.

This low-cost and powerful development system consists of three components—a computer, a Brain Board and a Personality Board. This partitioning gives you added flexibility in emulating microcontrollers.

The host provides editing and program assembly with cost-effective bulk storage and high-speed processing capabilities. Object code results are downloaded to the Brain/Personality Boards via RS232C interface for real-time in-system emulation.

Resident firmware allows the user to display and alter memory in both hex and mnemonic format; initiate breakpoints, traces and timing on addresses or external events; and examine and modify the internal resources of the microcontroller being emulated.

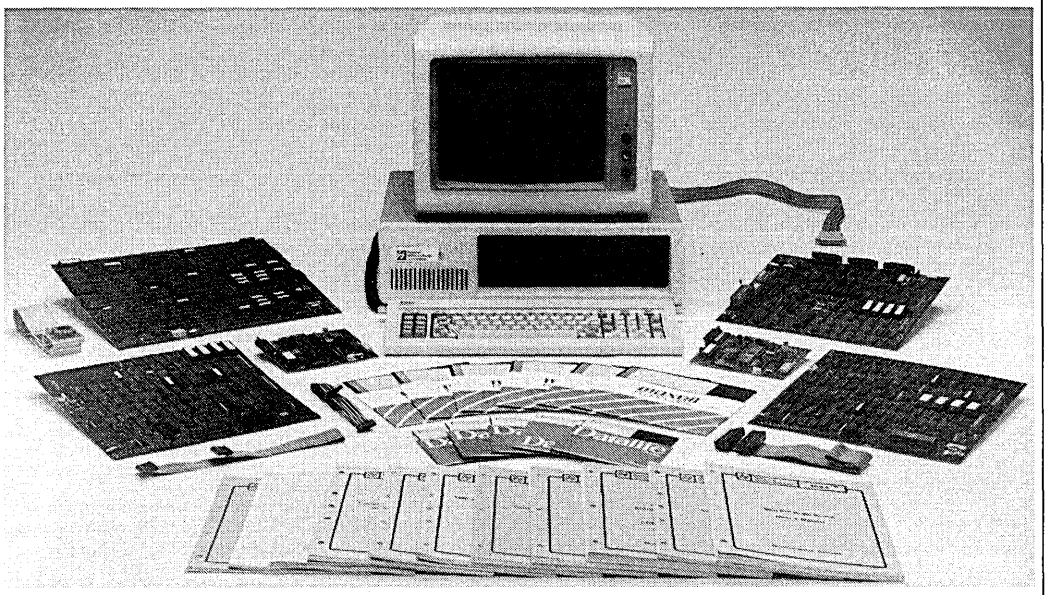
The Brain Board, common to all versions of the MOLE, contains an NSC800, 64k ROM, 32k ROM, 3 RS-232C channels and a PROM programmer.

The Personality Board contains the necessary hardware and firmware to tailor the MOLE system to the specific microcontroller. When connected to a CRT, the two boards provide all the resources for stand-alone emulation and debugging.

Connecting a MOLE to Intel's MDS, or other CP/M MS-DOS-based computers completes the development system.

Microcontroller Development Tools

National Semiconductor Corp.
Application Note 456
Microcontroller Marketing



TL/DD/8830–14

MOLE™ DEVELOPMENT TOOLS

The MOLE (Microcontroller On Line Emulator) system was designed to support the development of all NSC Microcontroller products. These include COPS™ family, TMP, and the HPC family of products. The MOLE provides effective support for the development of both software and hardware in the user's application.

The purpose of MOLE is to provide the user with the tools to write and assemble code for the target microcontroller and assist in the debugging of both the hardware and software.

A MOLE system consists of three components: a MOLE Brain Board, a MOLE Personality Board, and software for the user's host computer. This partitioning provides the Microcontroller design engineer with a new concept in flexibility. As an example of the flexibility consider the latitude the user has in the choice of a host CPU. The host may be IBM®-PC, Intel's MDS800, or Intellec® series II, or Kaypro, or one of a number of inexpensive personal computers. The software provided by National Semiconductor will run under control of the host computer CP/M or MS-DOS operating system.

Further flexibility is provided by the Personality board. This component tailors the system to emulate a single microcontroller family or device. For instance, one Personality Board supports the COPS CMOS and NMOS family. This Personality Board provides support for 42 Microcontroller device types.

Additionally, Personality boards are available for the HPC family of products and the NS405 Terminal Management Processor (TMP).

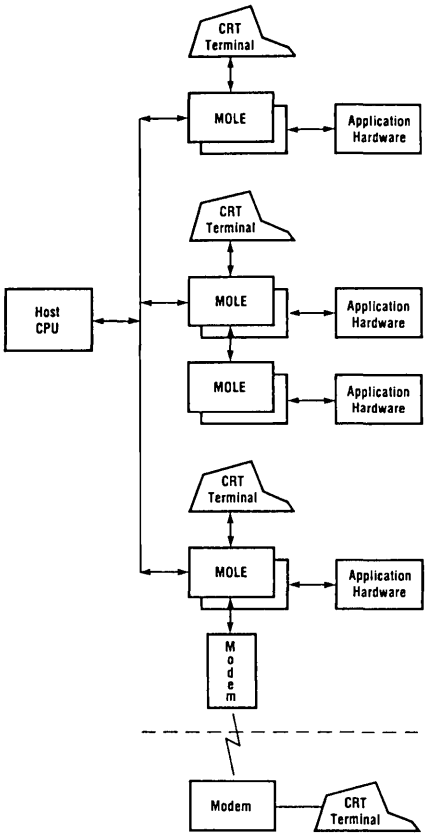
The MOLE components have been designed to provide maximum utility. The host CPU contributes cost effective bulk storage and high speed processing. Disk editing and assembly operations are controlled by the host CPU. The results are down loaded to the Brain Board over the RS-232 link. The Brain Board/Personality Board combination provides FULL emulation capability.

The resident firmware allows the user to down load from the host computer, display and alter code in both hex and mnemonic format, initiate Breakpoints, Traces, and timing on addresses and external events, examine and modify the internal resources of the Microcontroller being emulated. Hardware and Firmware are provided for programming EPROMs or EEPROMs.

Once debugged, the code is transmitted to National Semiconductor for use in creating the tooling necessary for manufacturing the masked Microcontroller part.

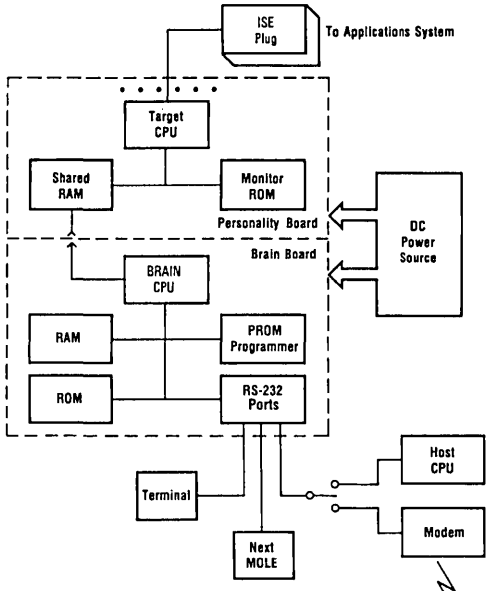
The MOLE concept provides maximum flexibility and maximum utility for development of a Microcontroller based system.

MOLE System Configuration

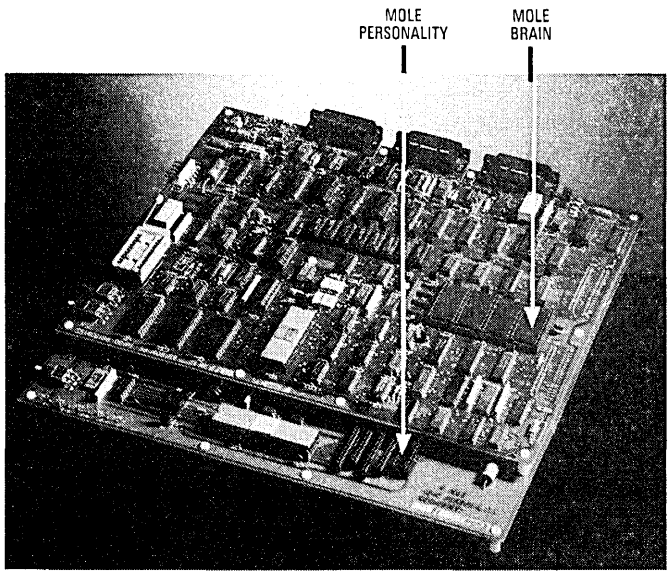


TL/DD/8830-1

MOLE System Block Diagram



TL/DD/8830-2



TL/DD/8830-3

MOLE BRAIN BOARD

The Brain Board is the pivotal component of the MOLE concept. In conjunction with a terminal and Personality Board it provides the user with a freestanding workstation for Microcontroller emulation. It ties the system together by communicating with the Personality Board, printers, modems, optional host computer, and other Brain Boards. Multiple Brain Boards, tied to a common host, can function as emulators for individual projects where each Brain Board is a separate workstation. They can also function as individual Microcontroller emulators within a multicontroller system.

The MOLE Brain Board utilizes a NSC800™ Microprocessor with 64k RAM and 32k ROM. It has an EPROM /EEPROM programmer for on-line changes. There are three RS-232 ports and a bus to connect the Brain to the Personality Board for actual emulation of code in the user's application system.

The RS-232 ports are used via the communication routines in firmware to interface with a host computer, terminal, modem, printer, or other MOLEs, for greater flexibility during system development.

The MOLE firmware is controlled by an EXEC. There are three major sets of EXEC commands. The first set of commands are calls to other main programs. These are:

COMM	Invoke Communications Program
DIAG	Invoke Diagnostics Program
MONITOR	Invoke Personality Emulation Monitor
PROG	Invoke PROM Programming Program

The second set of EXEC commands are:

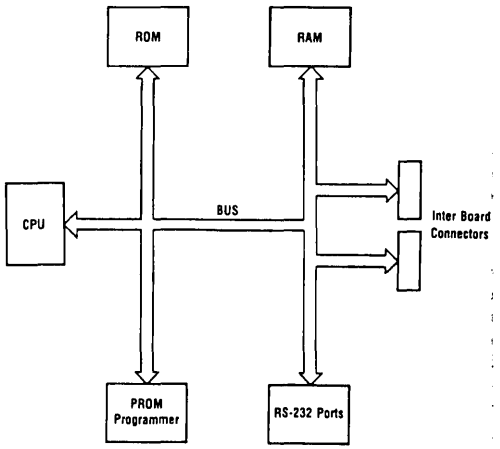
CALC	Adds/Subtracts decimal and hex numbers
COMPARE	Compares one buffer with another
ERASE	Used to erase all or part of a buffer
HELP	Prints a summary of EXEX commands
MOVE	Moves data from one buffer to another
STATUS	Display status of buffers, display and alter RS-232 parameters

The third set of commands are used exclusively for multiple MOLE configurations and they are:

CONNECT	Connect the user with the requested system
DISCONNECT	Disconnects the MOLE
IDENT	Identifies a MOLE system

The MOLE Brain Board supports NSC's entire family of MOLE Personality boards.

MOLE Brain Board Block Diagram



TL/DD/8830–5

- Single 5V operation
- Ability to interface to a wide variety of host computers
- Full communication control with host computer, a modem of other MOLEs
- Three RS-232 ports
- Auto baud selection (110, 300, 600, 1200, 2400, 4800, 9600, 19200 baud)
- Mask data submission via modem
- Self diagnostics
- Program EPROMS
  - — MM2716, NMC27C16
  - — MM2732, NMC27C32
  - — NMC2764
- Program EEPROMs
  - — 2816A

PHYSICAL SIZE
10″ x 12″

POWER REQUIREMENTS
+5V DC @ 3.5A
+12.5/+21V or +25V @ 50 mA
(Optional—required only for PROM programming)

WHAT P/N TO ORDER
MOLE-BRAIN

MOLE-BRAIN PACKAGE CONTAINS
MOLE Brain Board
MOLE Brain User's Manual P/N 420408188-001
2 RS-232 Cables
Power Cable
Miscellaneous Hardware

MOLE COPS FAMILY PERSONALITY BOARD

The MOLE COPS Family Personality Board supports the emulation of COPS family of Microcontrollers. The Personality Board allows the user to emulate the appropriate Microcontroller in the user's end system for fast development of application code and hardware. The Personality Board consists of: a Monitor, the hardware to control the operation of the Microcontroller in the emulation system, and an emulation cable to connect the emulator to the application system. The cable has the same pin configuration as the final masked part.

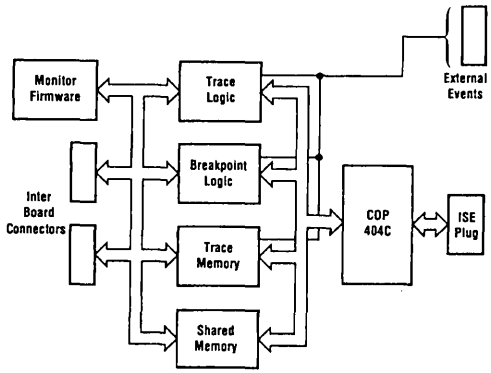
The Personality Board Monitor is contained in 32k of ROM, contains an assembler and disassembler and is directly executable by the NSC800 on the Brain Board. The Monitor commands will allow the user to execute the application code, examine and modify internal registers and I/O, examine and alter object code in hex or mnemonic format, execute Time measurements, and set Trace and Breakpoints.

The Personality Board also contains 2k bytes of shared memory (RAM) for application code and the necessary hardware for Trace and Breakpoint operation.

When the system is thoroughly tested, the code is sent to National Semiconductor for use in creating the required tooling for manufacturing the masked COPS production part.

- Supports entire COPS CMOS and NMOS family
- Single 5V operation
- Firmware monitor directly executed by Brain CPU
- Firmware diagnostics directly executed by Brain CPU
- Firmware Line Assembler
- Firmware Unassembler
- 2k bytes of shared memory
- 256 deep trace memory
- Eight external event inputs
- Trace on multiple addresses
- Trace on multiple address ranges
- Trace on external events
- Breakpoint on multiple addresses
- Breakpoint on multiple address ranges
- Breakpoint on external events
- List and alter shared memory
- Print and modify internal registers
- Singlestep
- Next—singlestep around subroutine calls
- Trigger output for logic analyzer
- Real time emulation

MOLE CMOS COPS Family
Personality Board Block Diagram



TL/DD/8830–7

PHYSICAL SIZE
12″ x 12″

POWER REQUIREMENTS
+5V @ 3.5A

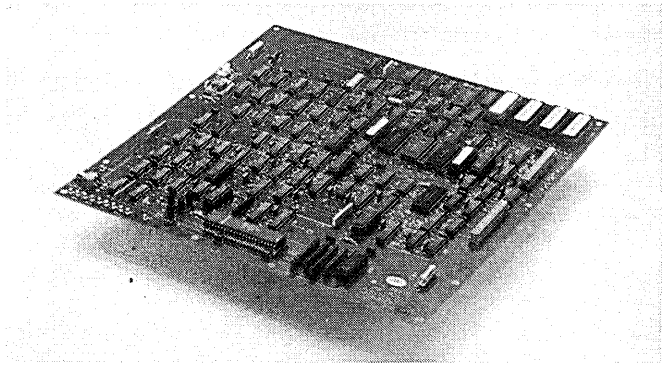
WHAT P/N TO ORDER
MOLE-COPS-PB1

MOLE-COPS PB1 PACKAGE CONTAINS
MOLE CMOS COPS Personality Board
MOLE CMOS COPS PB Manual
P/N 420408189-001
3 Emulator Cables
Power Cable
Miscellaneous Hardware

SOFTWARE ORDERED SEPARATELY
See Mole Software

COPS PROGRAMMERS MANUAL
Order P/N 424410284-001

COPS Personality Board



TL/DD/8830–6

MOLE TMP PERSONALITY BOARD

The TMP Personality Board allows the MOLE system to emulate the NS405 Terminal Management Processor (TMP). The Personality Board consists of a Monitor, 8k bytes of shared memory, 2k bytes of video memory, 2k bytes of character font memory, emulation hardware, and an In-System-Emulation, (ISE™), cable. The ISE cable has the same pinout as the chip and allows the Personality Board to function within the application system.

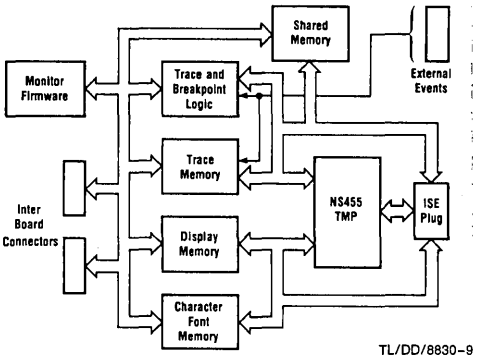
The NSC800 CMOS Microprocessor, located on the Brain Board, directly executes the 32k Personality Board monitor firmware. The Monitor allows execution of application code, examination and alteration of internal registers, examination and alteration of program memory contents, setting of trace and breakpoints, and control of the character generator memory. The ISE cable connects these capabilities to the application system. Up to eight external events may be traced in the 2k deep trace memory. Multiple trace, breakpoints are available to the user along with assemble and unassemble of shared memory.

Application programs of up to 8k bytes may be executed from Personality Board RAM or user system memory. Video display RAM also may be accessed from the Personality Board (2k bytes) or user system memory (64k bytes). Unique character sets may be displayed by accessing a high speed character font memory.

Once debugged, the code is transmitted to National Semiconductor for use in creating the tooling required to manufacture the part.

- Supports NS455 (TMP) microcontroller
- Single 5V operation
- Firmware monitor directly executed by Brain CPU
- Firmware diagnostics directly executed by Brain CPU
- Firmware Line Assembler
- Firmware Unassembler
- 8k bytes of shared program memory
- 2k bytes of video display memory
- 2k bytes of character font memory
- 2048 deep trace memory
- Eight external event inputs
- Trace on multiple addresses
- Trace on multiple address ranges
- Trace on external events
- Breakpoint on multiple addresses
- Breakpoint on multiple address ranges
- Breakpoint on external events
- List and alter shared memory
- List and alter display memory
- List and alter character font memory
- Print and modify internal registers
- Singlestep
- Next—singlestep around subroutine calls
- Trigger output for logic analyzer
- Real time emulation

MOLE NS455 (TMP) Personality Board Block Diagram



TL/DD/8830-9

PHYSICAL SIZE
12″ x 12″

POWER REQUIREMENTS
+5V @ 4A

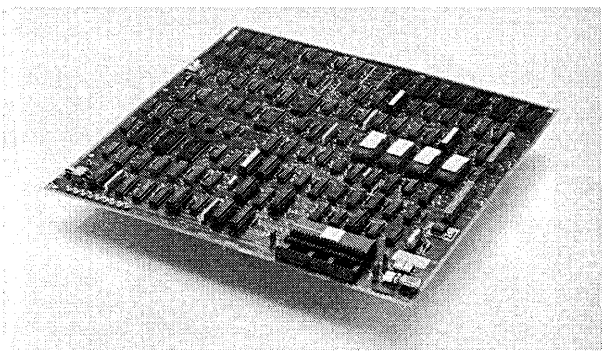
WHAT P/N TO ORDER
MOLE-TMP-PB1

MOLE-TMP-PB1 PACKAGE CONTAINS
MOLE TMP Personality Board
MOLE TMP PB User's Manual
P/N 420406203-001
1 Emulator Cable
Power Cable
Miscellaneous Hardware

SOFTWARE ORDERED SEPARATELY
See Mole Software

TMP APPLICATIONS MANUAL
Order P/N 420040401-001

TMP Personality Board



TL/DD/8830-8

MOLE COP800 FAMILY PERSONALITY BOARD

The COP800 Family Personality Board allows the MOLE system to emulate the COP800 family. The Personality Board consists of a firmware Monitor, 16k bytes of shared memory, 2000 deep Trace memory, Port recreation logic to recapture the pins used for emulation, emulation hardware, and an In System Emulator (ISE) cable. The ISE cable has the same pinout as a socketed masked part and allows the Personality Board to function within the application system.

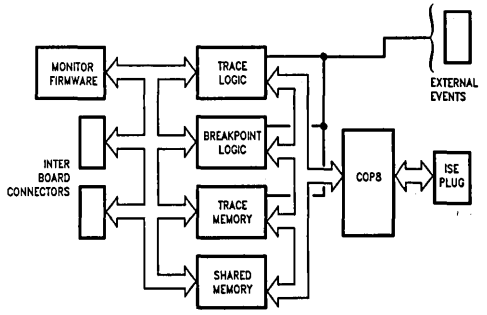
The NSC800 CMOS Microprocessor, located on the Brain Board, directly executes the 64k of Personality Board Monitor firmware. The Monitor allows execution of application code, examination and alteration of internal registers, examination and alteration of shared memory, and the setting of trace and breakpoints. The ISE cable connects these capabilities to the application system. Up to eight external events as well as 15-bit address and 8-bit data busses can be traced in the 2000 deep trace memory. Multiple breakpoints, plus assemble and unassemble commands are at the user's disposal.

Applications programs of up to 16k bytes from Personality Board RAM may be emulated.

Once debugged, the code is transmitted to National Semiconductor for use in creating the tooling required to manufacture the masked part.

- Supports COP800 microcontroller family
- Single 5V operation
- Firmware monitor directly executed by Brain CPU
- Firmware diagnostics directly executed by Brain CPU
- Firmware Line Assembler
- Firmware Unassembler
- 16k bytes of shared program memory
- 2000 deep trace memory
- Eight external event inputs
- Trace on multiple addresses
- Trace on multiple address ranges
- Trace on external events
- Breakpoint on multiple addresses
- Breakpoint on multiple address ranges
- Breakpoint on external events
- List and alter shared memory
- Print and modify internal registers
- Singlestep
- Next—singlestep around subroutine calls
- Trigger output for logic analyzer
- Real time emulation

MOLE CMOS COP800 Family Personality Board Block Diagram



TL/DD/8830–15

PHYSICAL SIZE
12" x 12"

POWER REQUIREMENTS
+5V @ 3.5A

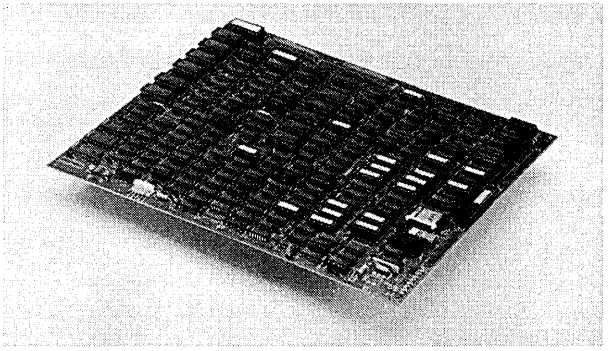
WHAT P/N TO ORDER
MOLE-COP8-PB1

MOLE-COP8-PB1 PACKAGE CONTAINS
MOLE CMOS COP8 Personality Board
MOLE CMOS COP8 PB Manual
P/N 420410528-001
Emulator Cables
Power Cable
Miscellaneous Hardware

SOFTWARE ORDERED SEPARATELY
See Mole Software

COP800 PROGRAMMERS MANUAL
Order P/N 420410703-001

COP800 Personality Board



TL/DD/8830–10

MOLE HPC FAMILY PERSONALITY BOARD

The HPC Family Personality Board allows the MOLE system to emulate the High Performance Controller (HPC) family. The Personality Board consists of a firmware Monitor, 16k bytes of shared memory, 2k x 48 Trace memory, Port recreation logic to recapture the pins used for emulation, emulation hardware, and an In System Emulator, ISE, cable. The ISE cable has the same pinout as a socketed masked part and allows the Personality Board to function within the application system.

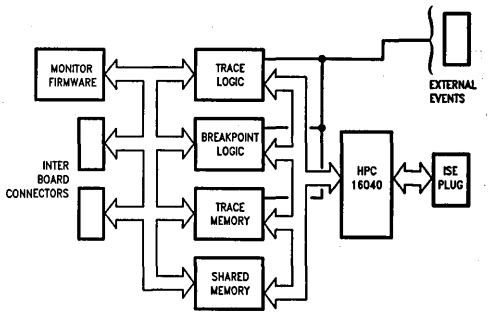
The NSC800 CMOS Microprocessor, located on the Brain Board, directly executes the 32k of Personality Board Monitor firmware. The Monitor allows execution of application code, examination and alteration of internal registers, examination and alteration of shared memory, and the setting of trace and breakpoints. The ISE cable connects these capabilities to the application system. Up to eight external events as well as 16-bit address and 16-bit data busses can be traced in the 2k deep trace memory. Multiple breakpoints, and chip error conditions plus assemble and unassemble commands are at the user's disposal.

Applications programs of up to 16k bytes from Personality Board RAM or 64k bytes from user system RAM may be emulated.

Once debugged, the code is transmitted to National Semiconductor for use in creating the tooling required to manufacture the masked part.

- Supports HPC microcontroller family
- Single 5V operation
- Firmware monitor directly executed by Brain CPU
- Firmware diagnostics directly executed by Brain CPU
- Firmware Line Assembler
- Firmware Unassembler
- 16k bytes of shared program memory
- 2000 deep trace memory
- Eight external event inputs
- Trace on multiple addresses
- Trace on multiple address ranges
- Trace on external events
- Breakpoint on multiple addresses
- Breakpoint on multiple address ranges
- Breakpoint on external events
- List and alter shared memory
- List and alter display memory
- Print and modify internal registers
- Singlestep
- Next—singlestep around subroutine calls
- Trigger output for logic analyzer
- Real time emulation

MOLE HPC Family Personality Board Block Diagram



TL/DD/8830-11

PHYSICAL SIZE
12" x 12"

POWER REQUIREMENTS
+5V @ 8A

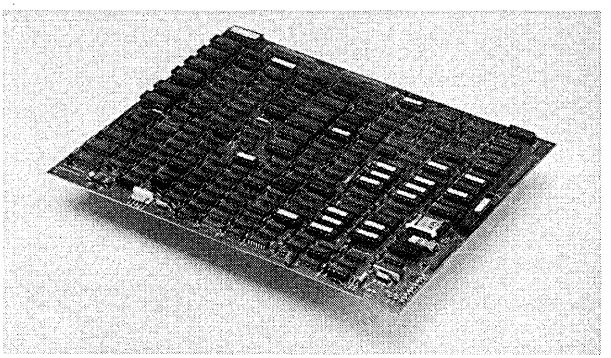
WHAT P/N TO ORDER
MOLE-HPC-PB1

MOLE-HPC-PB1 PACKAGE CONTAINS
MOLE HPC Personality Board
MOLE HPC PB User's Manual
P/N 420410477-001
1 Emulator Cable
Power Cable
Miscellaneous Hardware

SOFTWARE ORDERED SEPARATELY
See Mole Software

HPC PROGRAMMERS MANUAL
Order P/N 420410704-001

HPC Personality Board



TL/DD/8830-10

HOW TO ORDER

MOLE systems are available for a variety of host systems. To order a complete development package, select the section for the microcontroller to be developed and order the parts listed. MOLE software is available for several systems. For software other than for IBM systems, consult the alternate software selection table and substitute for the assembler software for the IBM.

Development Tools Selection Table

Icrocontroller	Order Part Number	Description	Includes	Manual Number
HPC	MOLE-BRAIN	Brain Board	Brain Board Users Manual	420408188-001
	MOLE-HPC-PB1	Personality Board	HPC Personality Board Users Manual	420410477-001
	MOLE-HPC-IBM*	Assembler Software for IBM	HPC Software Users Manual and Software Disk PC-DOS Communications Software Users Manual	424410446-001 420040416-001
	MOLE-HPC-IBM-C	C Compiler for IBM	HPC C Compiler Users Manual and Software Disk Assembler Software for IBM MOLE-HPC-IBM	424410559-001
	420410704-001	Programmers Manual		420410704-001
	420410705-001	Hardware Manual		420410705-001
COP800	MOLE-BRAIN	Brain Board	Brain Board Users Manual	420408188-001
	MOLE-COP8-PB1	Personality Board	COP800 Personality Board Users Manual	420410528-001
	MOLE-COP8-IBM*	Assembler Software for IBM	COP800 Software Users Manual and Software Disk PC-DOS Communications Software Users Manual	424410527-001 420040416-001
	MOLE-COP8-IBM-C	C Compiler for IBM	COP800 C Compiler Users Manual and Software Disk Assembler Software for IBM MOLE-COP8-IBM	Available 1988
	420410703-001	Programmers Manual		420410703-001
COP400	MOLE-BRAIN	Brain Board	Brain Board Users Manual	420408188-001
	MOLE-COPS-PB1	Personality Board	COP400 Personality Board Users Manual	420408189-001
	MOLE-COPS-IBM*	Assembler Software for IBM	COP400 Software Users Manual and Software Disk PC-DOS Communications Software Users Manual	424409479-002 420040416-001
	424410284-001	Programmers Manual		424410284-001
TMP	MOLE-BRAIN	Brain Board	Brain Board Users Manual	420408188-001
	MOLE-TMP-PB1	Personality Board	TMP Personality Board Users Manual	420408203-001
	MOLE-TMP-IBM*	Assembler Software for IBM	TMP Software Users Manual and Software Disk PC-DOS Communications Software Users Manual	424410087-002 420040416-001
	420040401-001	Applications Manual		420040401-001
	TMP-DEMO-12	12 MHz Demo Board (5 x 7 Font)	Demo Board, Data Sheet, and Operating Manual	420409478-001
	TMP-DEMO-18	18 MHz Demo Board (7 x 9 Font)	Demo Board, Data Sheet, and Operating Manual	420409478-001

For CP/M software see alternate assembler software selection table.

MOLE SOFTWARE

MOLE software is available for a variety of host computers. The software package provides cross assembly for the specific Microcontroller on the specific type of host computer. Included, along with the cross assembler, in the software package are two file conversion routines to convert the assembler output (LM) to HEX and to convert HEX to LM. Also included in the software package is a COMM program which facilitates the downloading and uploading between the host and the MOLE, and adds the capability to make the host act as a terminal. In addition, for the TMP, a program is provided to create custom fonts (FONT).

If using a host computer other than the IBM the select the Assembler software package from the following table.

Alternate Assembler Software Selection Table

HPC	COP800	COP400	TMP	Description
MOLE-HPC-CPM	MOLE-COP8-CPM	MOLE-COPS-CPM	MOLE-TMP-CPM	Assembler Software for CP/M Systems (8" SD Disk)
MOLE-HPC-INT-S	MOLE-COP8-INT-S	MOLE-COPS-INT-S	MOLE-TMP-INT-S	Assembler Software for Intel Single Density System (8" SD Disk)
MOLE-HPC-INT-D	MOLE-COP8-INT-D	MOLE-COPS-INT-D	MOLE-TMP-INT-D	Assembler Software for Intel Double Density Syster (8" DD Disk)
MOLE-HPC-KAY	MOLE-COP8-KAY	MOLE-COPS-KAY	MOLE-TMP-KAY	Assembler Software for KAYPRO2 Systems (5.25" DD Disk)
CP/M	Communications Software Users Manual (Included with the above Software)			420040415-0

DIAL-A-HELPER

Dial-A-Helper is a service provided by the MOLE (Microcontroller On Line Emulator) applications group. It consists of both an electronic bulletin board information system and a method by which applications can take control of a MOLE Development System at a remote site via modem in order to resolve any problems.

Information System

The Dial-A-Helper system provides access to an automated information storage and retrieval system that may be accessed over standard dial-up telephone lines 24 hours a day. The system capabilities include a MESSAGE SECTION (electronic mail) for communications to and from the Microcontroller Applications Group and a FILE SECTION mode that can be used to search out and retrieve application data about NSC Microcontrollers. The user needs as a minimum, a Dumb terminal, 300 or 1200 baud Modem, and a telephone.

If the user has a PC with a communications package then files from the FILE SECTION can be down loaded to disk for later use.

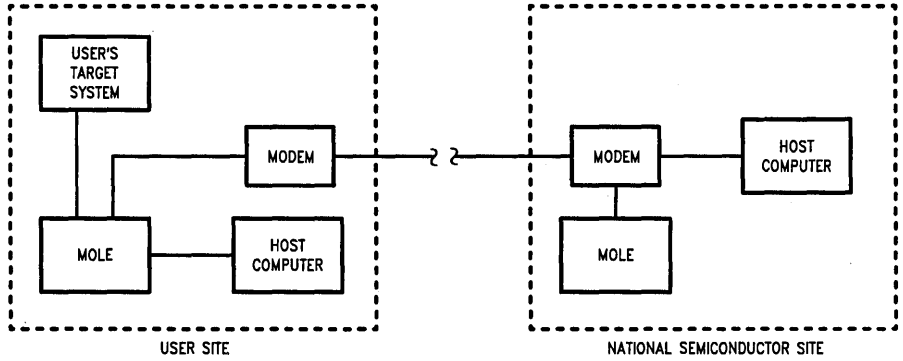
Factory Applications Support

Dial-A-Helper also provides immediate factory applications support. If a user is having difficulty in getting a MOLE to operate in a particular mode or something peculiar is occuring, he can contact us via his system and modem. He can leave messages on our electronic bulletin board, which we will respond to, or he can arrange for us to actually take control of his system via modem for debugging purposes.

The applications group can then cause his system to execute various commands and try to resolve the customers problem by actually getting customers system to respond. Both parties see exactly what is occurring, as it is happening.

This allows us to respond in minutes when applications help is needed.

DIAL-A-HELPER



TL/DD/8830-16

TMP DEMO BOARD

The TMP demo board is a minimum system configuration, CRT terminal. It utilizes the NS405 Terminal Management Processor (TMP) Single Chip CRT Controller microchip. TMP is suitable for low cost terminal, imbedded or display functions.

The Demo board is an excellent vehicle for evaluating the performance and capabilities of TMP or for OEM applications .

The addition of a keyboard, and a display is all that is required to provide ADM 3A terminal compatibility. The user has the flexibility of executing the on-board ROM functions or creating their own versions utilizing an external EPROM. The manual contains a complete, well documented listing, of the source code.

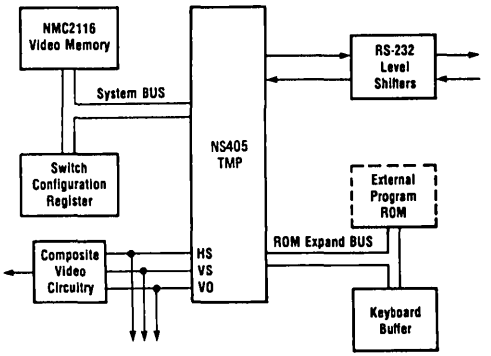
In addition, the user has complete control over temrinal operation through the use of the many escape sequences programmed into the microchip. Many cursor and character attribute display options available can easily be evaluated.

The heart of the Demo board is the NS405 Terminal Management Processor. Normally the TMP will be running its internal masked program, but will go external for program memory if instructed.

The ROM Expand bus is multiplexed with external program EPROM and 8-bit I/O port for the keyboard input. The System Bus similarly multiplexes the Video RAM and switch configuration register.

The video memory requirement of 80 columns by 25 rows is stored in a 2048 x 8 NMC2116 static RAM. Also mapped into the video RAM space is the switch configuration register. The switch register is used on power up initialization to set up the UART operation. The video outputs are either buffered or mixed together for composite video. The UART serial IN and OUT are level shifted to RS232 levels by the dual op-amp (LM1458).

Block Diagram

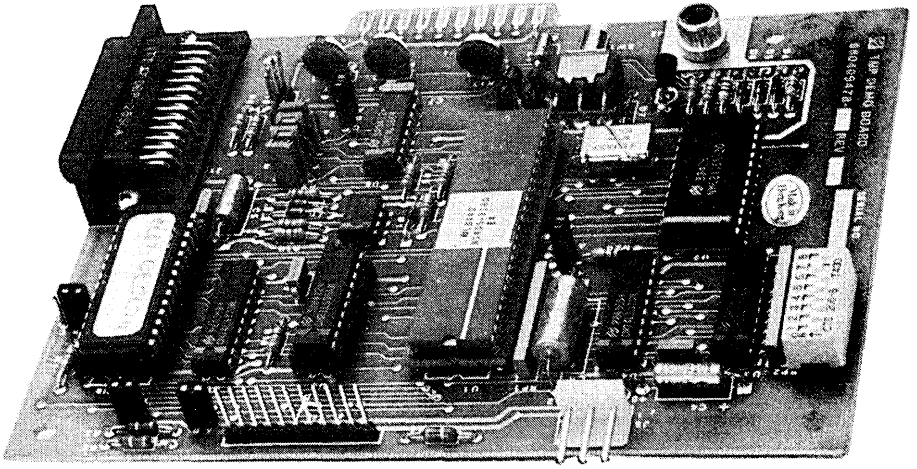


TL/DD/8830–13

Features

- 80 column x 25 row display
- Double wide font
- 12 or 18 MHz bandwidth with 5 x 7 or 7 x 9 fonts respectively
- Ball or composite video output
- RS-232C serial interface with full duplex 110–19.2k baud
- 50/60 Hz operation
- Status line display
- 24 escape sequences
- 15 control sequences
- Internal masked ROM program or self-programmed EPROM operation

TMP Demo Board



TL/DD/8830–17

HARDWARE DESCRIPTION

Keyboard Input

Seven-bit parallel ASCII data from an encoded keyboard is input through a row of wire wrap pins that make up the J2 connector. In addition to data input, +5V to −12V power to operate the keyboard is available. Characters are strobed in on each logic low level. The strobe may be any duration greater than 1.25 μs; the TMP checks to see that the strobe line has returned to its quiescent logic high state before accepting new characters. For strobes longer than 750 ms, an auto repeat function outputs 15 characters/s. In auto repeat, data lines are sampled 15 times/s. In general, data must be read within 10 μs after the falling strobe edge and 20 μs after the falling strobe. Data lines should change only when a new character is available.

Serial Interface

The demo board can be interfaced to a host computer with a RS-232C data link by using the J1 Plastic D connector. The parameters associated with serial data communication are user selectable.

Video Interface

The video input can be either a composite video or Ball (separate H-Sync, V-Sync, Video Input) monitor input. The monitor should have at least a 12 MHz or 18 MHz bandwidth to operate properly with, respectively, 12 MHz or 18 MHz TMP Demo Boards.

Program Memory

The program to run the demo board is shipped with the board in a 2716 EPROM. Both boards allow the user to customize their own program by adding or changing the EPROM.

Light Pen Input

The TMP chip has the capability of using a light pen. The TMP Demo Board is able to take advantage of this by connecting the light pen to jumper W8A.

TMP Terminal Operation

With everything powered up and running, a blinking block cursor will be in the upper left hand corner and the system status line on bottom of screen. At powerup, the screen format is an 80-column by 25 row display with the following character and cell dimensions.

TABLE II

	Character Size	Cell Size
12 MHz, 50 Hz	5 x 7	7 x 12
12 MHz, 60 Hz	5 x 7	7 x 10
18 MHz, 50 Hz	7 x 9	9 x 12
18 MHz, 60 Hz	7 x 9	9 x 12

The user has the option of displaying the status line in the 25th row or providing a full 25-row data field. Operation at this point is like any other CRT data terminal. A number of special Control and Escape sequences programmed into the TMP give a user a higher degree of flexibility in terminal operation. The Control and Escape sequences may be activated by sending the proper codes from either the ASCII keyboard or host computer through the RS-232 data link.

Escape sequences are 2 or more character transmissions with ESC being the first character sent.

PART NUMBERS

The TMP (Terminal Management Processor) boards can be ordered by requesting the following part numbers:

TMP-DEMO-12

(for 5 x 7 character font—12 MHz bandwidth)

TMP-DEMO-18

(for 7 x 9 character font—18 MHz bandwidth)

CONTAINS:

Demo Board

NS405 Series Data Sheet

Demo Board Operating Manual

P/N 420409478-001

TMP APPLICATIONS MANUAL:

Order P/N 420040401-001

Section 9

Appendices/ Physical Dimensions

Section 9 Contents

National Semiconductor Corporation

Plastic Chip Carrier (PCC) Packaging

General Description

The Plastic Chip Carrier (PCC) is a miniaturized low cost semiconductor package designed to replace the Plastic Dual-In-Line Package (P-DIP) in high density applications. The PCC utilizes a smaller lead-to-lead spacing—0.050″ versus 0.100″ - and leads on all four sides to achieve a significant footprint reduction over the P-DIP. The rolled under J-bend leadform separates this package style from other plastic quad packages with flat or gull wing lead forms. As with virtually all packages of 0.050″ or less lead spacing, the PCC requires surface mounting to printed circuit boards as opposed to the more conventional thru-hole mounting of the P-DIP.

History

The Plastic Chip Carrier with J-bend leadform was first introduced in 1976 as a premolded plastic package. The premolded version has yet to become popular but the quad format with J-Bend leads has been adapted to traditional post molded packaging technology (the same technology used to manufacture the P-DIP). In 1980 National Semiconductor developed a post molded version of the PCC. The J-bend leadform allowed them to adopt the footprint connection pattern already registered with JEDEC for the leadless chip carrier (LCC). In 1981 a task force was organized within JEDEC to develop a PCC registration for package I/O counts of 20, 28, 44, 52, 68, 84, 100, and 124. A registered outline was completed in 1984 (JEDEC Outline MO-047) after many changes and improvements over the original proposals. This first PCC registration covers square packages with an equal number of leads on all sides. A second registration, MO-052, was completed in 1985 for rectangular packages with I/O counts of 18, 22, 28 and 32.

Since 1980 many additional semiconductor manufacturers and packaging subcontractors have developed PCC capability. There are now well over 20 sources with the number growing steadily.

Surface Mounting

Surface mounting refers to component attachment whereby the component leads or pads rest on the surface of the PCB instead of the traditional approach of inserting the leads into through-holes which go through the board. With surface mounting there are solder pads on the PCB which align with the leads or pads on the component. The resulting solder joint forms both the mechanical and electrical connection.

ADVANTAGES

The primary reason for surface mounting is to allow leads to be placed closer together than the 0.100″ standard for DIPs with through-hole mounting. Through-hole mounting on smaller than 0.100″ spacing is difficult to achieve in production and generally avoided. The move to 0.050″ lead spacing offered with the current generation of surface mounted components, along with a switch from a dual-in-line format to a quad format, has achieved a threefold increase in component mounting density. A need to achieve greater density is a major driving force in today's marketplace.

MANUFACTURING TECHNIQUES

Learning how to surface mount components to printed circuit boards requires the user to become educated in new assembly processes not typically associated with through-hole insertion/wave soldering assembly methods.

Surface mounting involves three basic process steps:

1) Application of solder or solder paste to the printed circuit board.
2) Positioning of the component onto the printed circuit board
3) Reflowing of the solder or solder paste.

As with any process, there are many details involved to achieve acceptable throughput and acceptable quality. National Semiconductor offers a surface mounting guide which deals with the specifics of successful surface mounting. We encourage the user to review this document and to contact us if further information on surface mounting is desired.

Benefits of the PCC

There are four principle advantages offered the user by switching from P-DIP to PCC. These four advantages are outlined below as follows:

1. Increased Density—
   - — Typically 3-to-1 size reduction of printed circuit boards. See *Figure 1* for a footprint comparison between PCC and P-DIP. This can be as high as 6-to-1 in certain applications.
   - — Surface mounting allows components to be placed on both sides of the board.
   - — Surface mount and thru-hole mount components can be placed on the same board.
   - — The large diameter thru-holes can be reduced in number, entirely eliminated, or reduced in size (if needed for via connection).
2. Increased Performance—
   - — Shorter traces on printed circuit boards.
   - — Better high frequency operation.
   - — Shorter leads in package. *Figure 2* and Table I compare PCC and P-DIP mechanical and electrical characteristics.
3. Increased Reliability—
   - — Leads are well protected.
   - — Fewer connectors.
   - — Simplified rework.
   - — Vibration and shock resistant.
4. Reduced Cost—
   - — Fewer or smaller printed circuit boards.
   - — Less hardware.
   - — Same low cost printed circuit board material.
   - — Plastic packaging material.
   - — Reduced number of costly plated-through-holes.
   - — Fewer circuit layers.

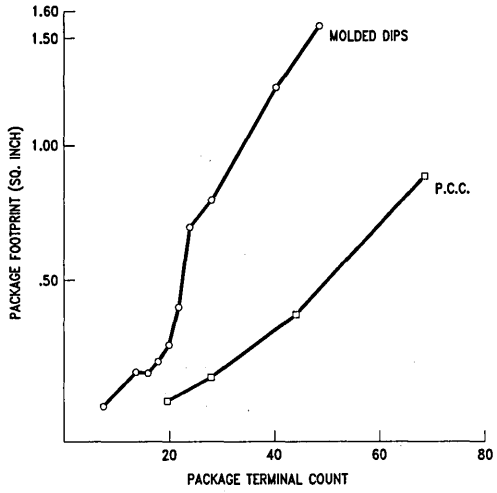


FIGURE 1. Footprint Area of PCC vs. P-DIP

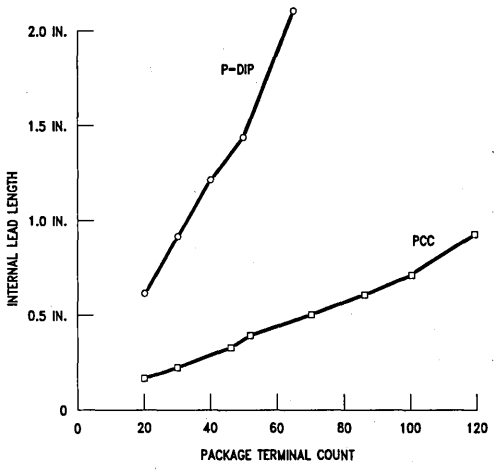


FIGURE 2. Longest Internal Lead PCC vs. P-DIP

TABLE I. Electrical Performance of PCC vs. P-DIP (44 I/O PCC vs. 40 I/O P-DIP, both with Copper Leads)

Criteria	Shortest Lead		Longest Lead	
	PCC	P-DIP	PCC	P-DIP
Lead Resistance (Measured)	3Ω	4Ω	6Ω	7Ω
Lead-to-Lead Capacitance (Measured on Adjacent Leads)	0.1 pF	0.1 pF	0.3 pF	3.0 pF
Lead Self-Inductance (Calculated)	3.2 nH	1.4 nH	3.5 nH	19.1 nH

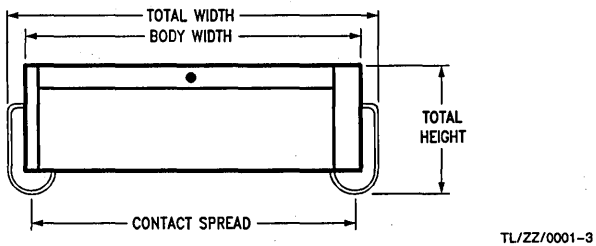


FIGURE 3. Package Outline

TABLE II. Principle Dimensions Inches/(Millimeters) (Refer to *Figure 3*)

Lead Count	Total Width		Total Height		Body Width		Contact Spread	
	Min	Max	Min	Max	Min	Max	Min	Max
20	0.385 sq. (9.779)	0.395 sq. (10.03)	0.165 sq. (4.191)	0.180 sq. (4.572)	0.345 sq. (8.763)	0.355 sq. (9.017)	0.310 sq. (7.874)	0.330 sq. (8.382)
28	0.485 sq. (12.32)	0.495 sq. (12.57)	0.165 sq. (4.191)	0.180 sq. (4.572)	0.445 sq. (11.30)	0.455 sq. (11.56)	0.410 sq. (10.41)	0.430 sq. (10.92)
44	0.685 sq. (17.40)	0.695 sq. (17.65)	0.165 sq. (4.191)	0.180 sq. (4.572)	0.645 sq. (16.38)	0.655 sq. (16.64)	0.610 sq. (15.49)	0.630 sq. (16.00)

TABLE II. Principle Dimensions Inches/(Millimeters) (Refer to *Figure 3*) (Continued)

Lead Count	Total Width		Total Height		Body Width		Contact Spread	
	Min	Max	Min	Max	Min	Max	Min	Max
68	0.985 sq. (25.02)	0.995 sq. (25.27)	0.165 sq. (4.191)	0.180 sq. (4.572)	0.945 sq. (24.00)	0.955 sq. (24.26)	0.910 sq. (23.11)	0.930 sq. (23.62)
84	1.185 sq. (30.10)	1.195 sq. (30.36)	0.165 sq. (4.191)	0.180 sq. (4.572)	1.150 sq. (29.21)	1.158 sq. (29.41)	1.110 sq. (28.20)	1.130 sq. (28.70)
124	1.685 sq. (49.13)	1.695 sq. (49.39)	0.180 sq. (4.572)	0.200 sq. (5.080)	1.650 sq. (41.91)	1.658 sq. (42.11)	1.610 sq. (40.90)	1.630 sq. (41.40)

TABLE III. Package Thermal Resistance
(Deg. C/Watt, Junction-to-Ambient, Board Mount)

Lead Count	Device Size		
	1,000 Mil^2	10,000 Mil^2	100,000 Mil^2
20	102	85	67
28	95	73	55
44	54	47	40
68	44	40	38
84*	40	35	30
124*	40	35	30

timated values

Package Design Criteria

Experience has taught us there are certain criteria to the PCC design which must be followed to provide the user with the proper mechanical and thermal performance. These requirements should be carefully reviewed by the user when selecting suppliers for devices in PCC. Some of these are covered by the JEDEC registration and some are not. These important requirements are listed in Table IV.

Reliability

National Semiconductor utilizes an assembly process for the PCC which is similar to our P-DIP assembly process. We also utilize identical materials. This is a very important point when considering reliability. Many years of research and development have gone into steadily improving our P-DIP quality and maintaining a leadership position in plastic package reliability. All of this technology can be directly applied to the PCC. Table V shows the results of applying this technology to the PCC. As we make further advances in plastic package reliability, these will also be applied to the PCC.

Sockets

There are several manufacturers currently offering sockets for the plastic chip carrier. Following is a listing of those manufacturers. The listing is divided into test/burn-in and production categories. There may be some individual sockets that will cover both requirements.

TABLE IV. Package Design Criteria

Criteria	Required to Comply with JEDEC Registration
Minimum Inside Bend Radius of Lead at Shoulder Equal or Greater than Lead Thickness—to Prevent Lead Cracking/Fatigue	Not Required
Minimum One Mil Clearance Between Lead and Plastic Body at all Points—to Provide Lead Compliancy and Prevent Shoulder Joint Cracking/Fatigue	Not Required
Copper Leads for Low Thermal Resistance	Not Required
Minimum 10 Mil Lead Thickness for Low Thermal Resistance and Good Handling Properties	Not Required
Minimum 26 Mil Lead Shoulder Width to Prevent Interlocking of Devices During Handling	Yes
Maximum 4 Mils coplanarity Across Seating Plane of all Leads	Yes

TABLE V. Reliability Test Data
(Expressed as Failures per Units Tested)

Device/Package	OPL	TMCL	TMSK	BHTL	ACLV
LM324/20 Lead	0/96	0/199	0/50	0/97	0/300
LF353/20 Lead	0/50	0/50	—	0/45	0/100
DS75451/20 Lead	0/47	—	0/50	0/93	0/179
DM875191/28 Lead	0/154	0/154	0/154	0/154	0/154
DM875181/28 Lead	0/77	0/77	0/77	0/77	0/77

OPL = Dynamic high temperature operating life at 125°C or 150°C, 1,000 hours.
TMCL = Temperature cycle, Air-to-Air, −40°C to +125°C or −65°C to +150°C, 2,000 cycles.
TMSK = Thermal shock, Liquid-to-Liquid, −65°C to +150°C, 100 cycles.
BHTL = Biased humidity temperature life, 85°C, 85% humidity, 1,000 hours.
ACLV = Autoclave, 15 psi, 121°C, 100% humidity, 1,000 hours.

Production Sockets

AMP
Harrisburg, PA
(715) 564-0100

Augat
Attleboro, MA
(617) 222-2202

Burndy
Norwalk, CT
(203) 838-4444

Methode
Rolling Meadows, IL
(312) 392-3500

Textool
Irving, TX
(214) 259-2676

Thomas & Betts
Raritan, NJ
(201) 469-4000

Test/Burn-In Sockets

Plastronics
Irving, TX
(214) 258-1906

Textool
Irving, TX
(214) 259-2676

Yamaichi
c/o Nepenthe Dist.
(415) 856-9332

ADDITIONAL INFORMATION AND SERVICES

National Semiconductor offers additional Databooks which cover surface mount technology in much greater detail. We also have a surface mount laboratory to provide demonstrations and customer support, as well as technology development. Feel free to contact us about these additional resources.

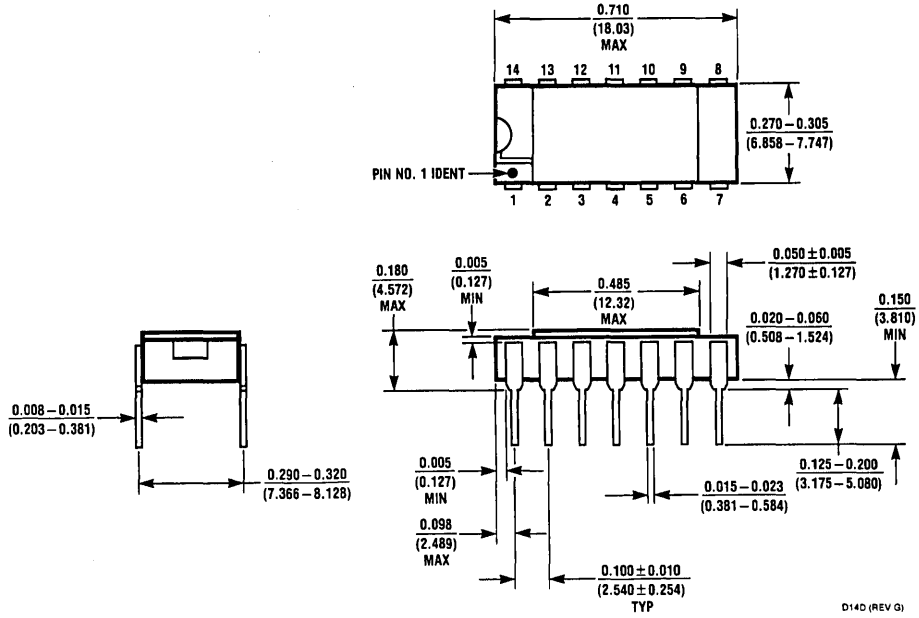
Technical Manuals

Technical Manuals are available from National's Sales offices

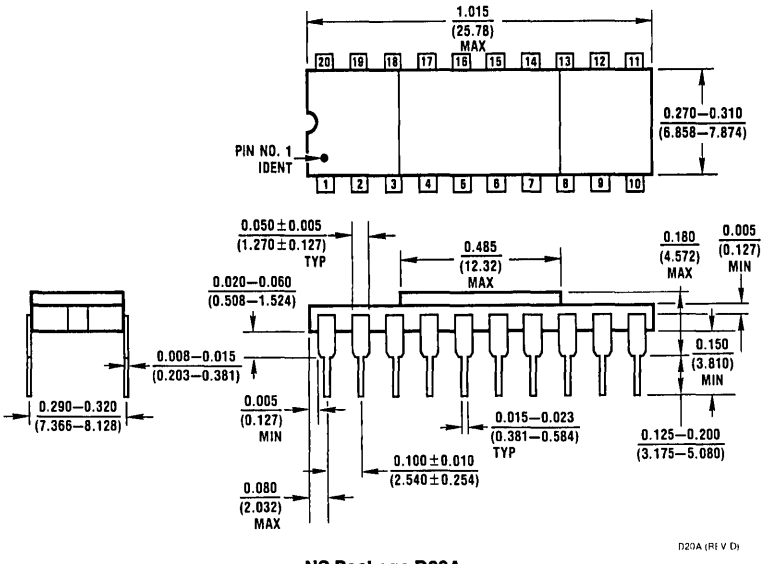
MOLE Brain Board Users Manual	COPS, HPC, TMP 8050	420408188-001
Software Users Manual	COPS	424409479-002
Personality Board Manual	COPS	420408189-001
	TMP	420408203-001
PC-DOS Software Users Manual	COPS, TMP, HPC, 8050	420040416-001
TMP-DEMO-12/18 Board Users Manual		420409478-001
NS405 Terminal Management Processor (TMP) Application Manual		420040401-001
COPS Programming Manual		424410284-001
CPM Software Users Manual		420040415-001

National Semiconductor Corporation

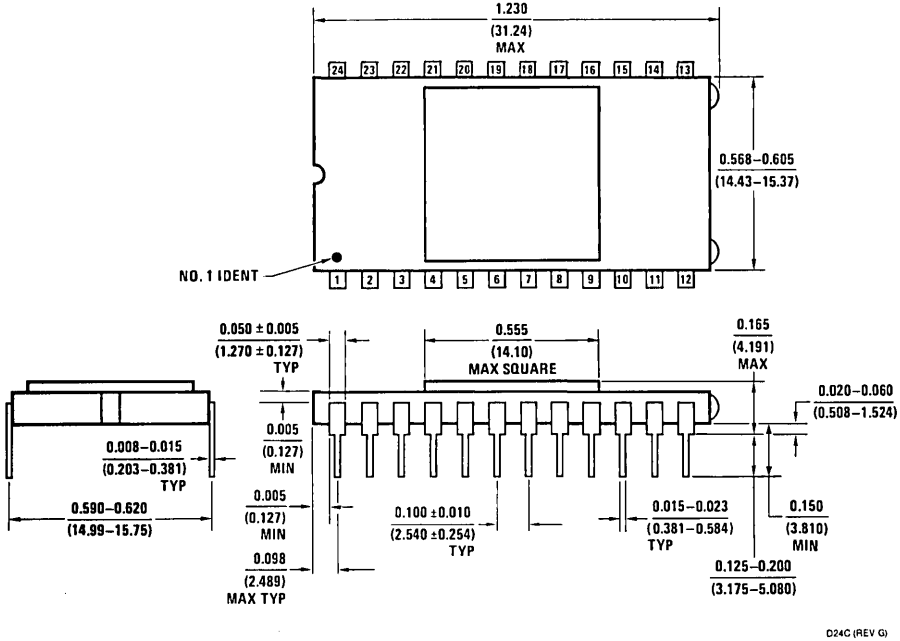
All dimensions are in inches (millimeters)



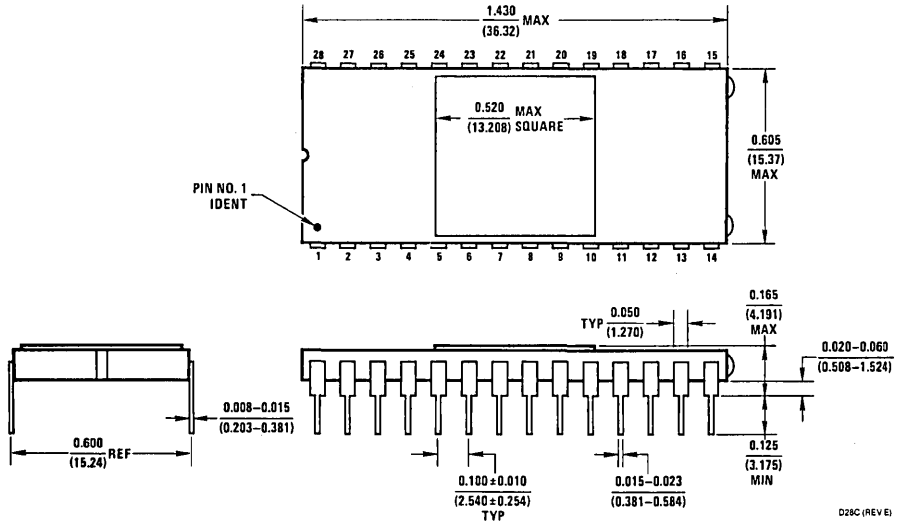
NS Package D14D



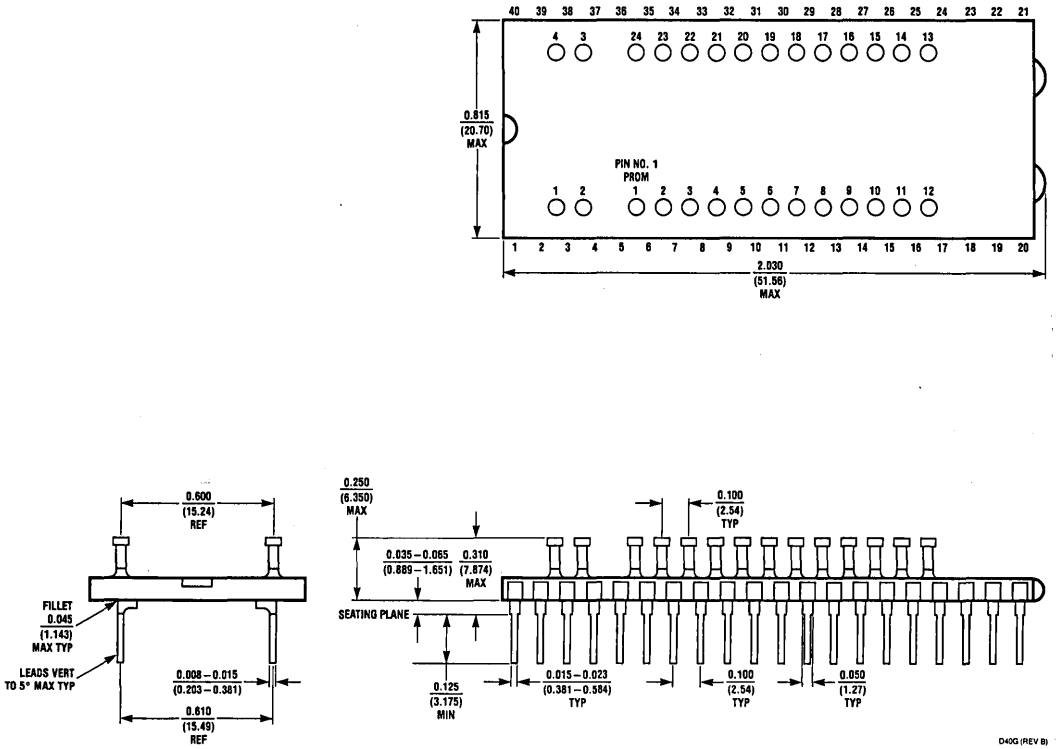
NS Package D20A



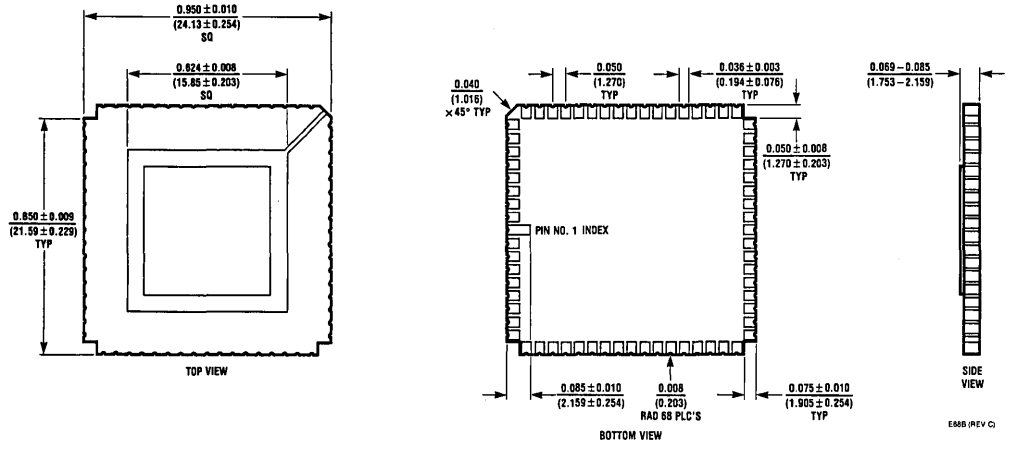
NS Package D24C



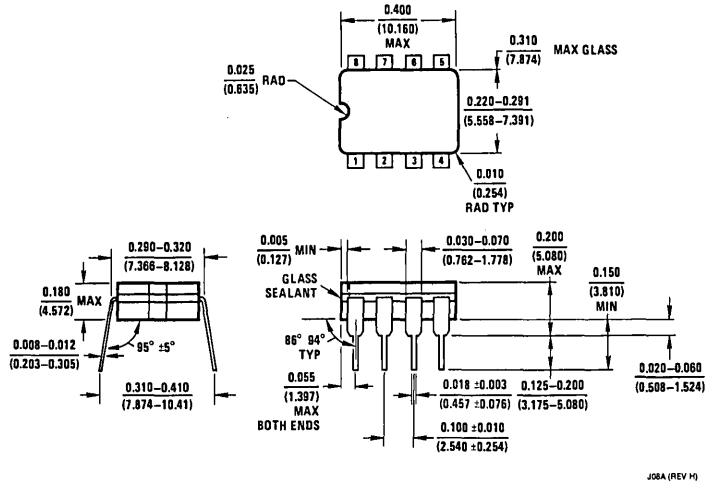
NS Package D28C



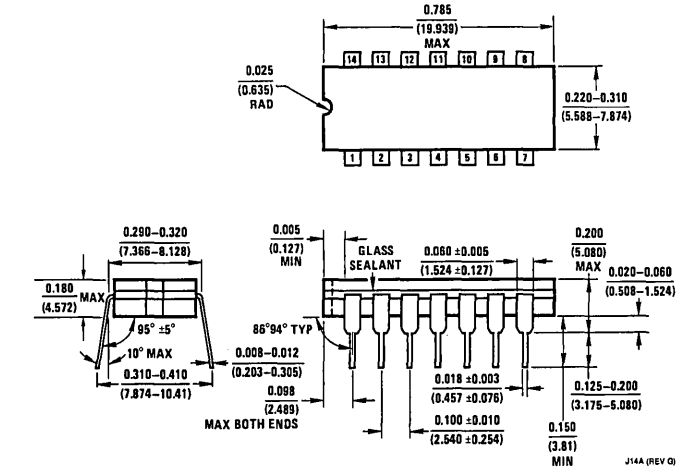
NS Package D40G



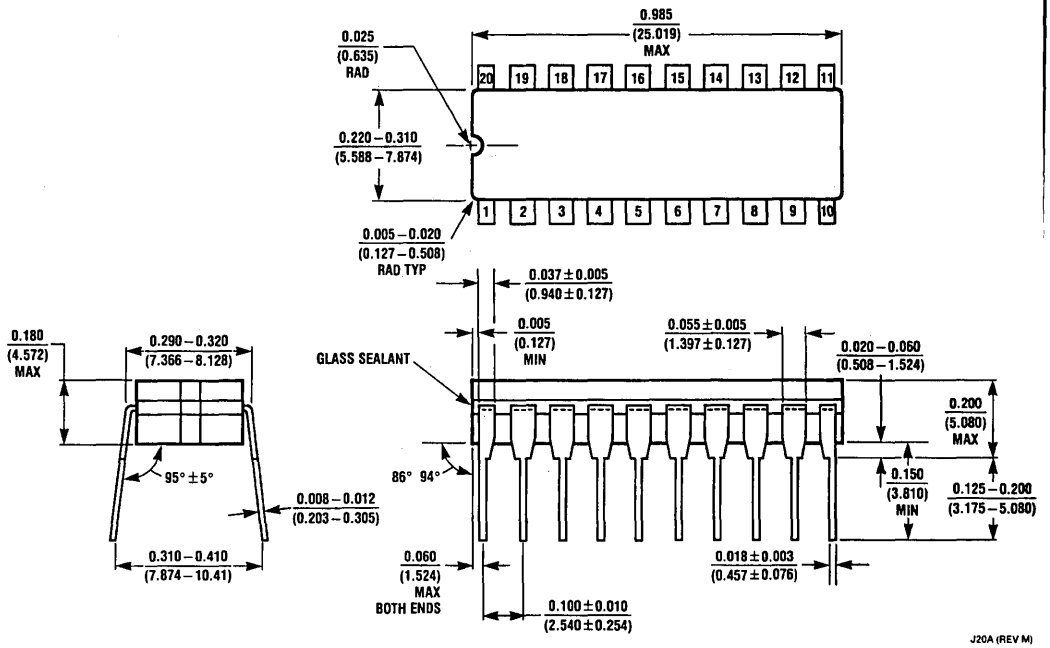
NS Package E68B



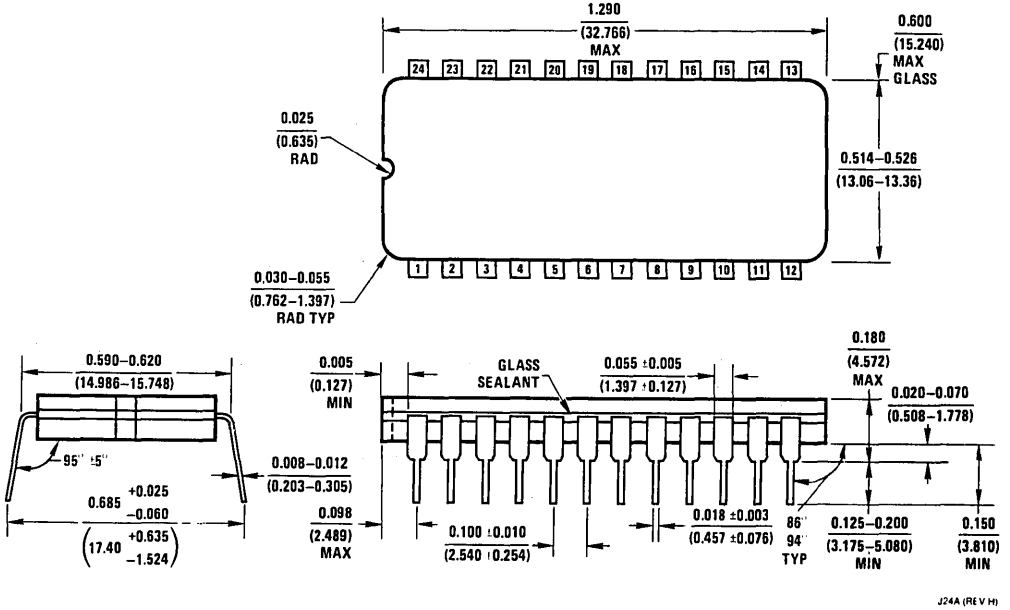
NS Package J08A



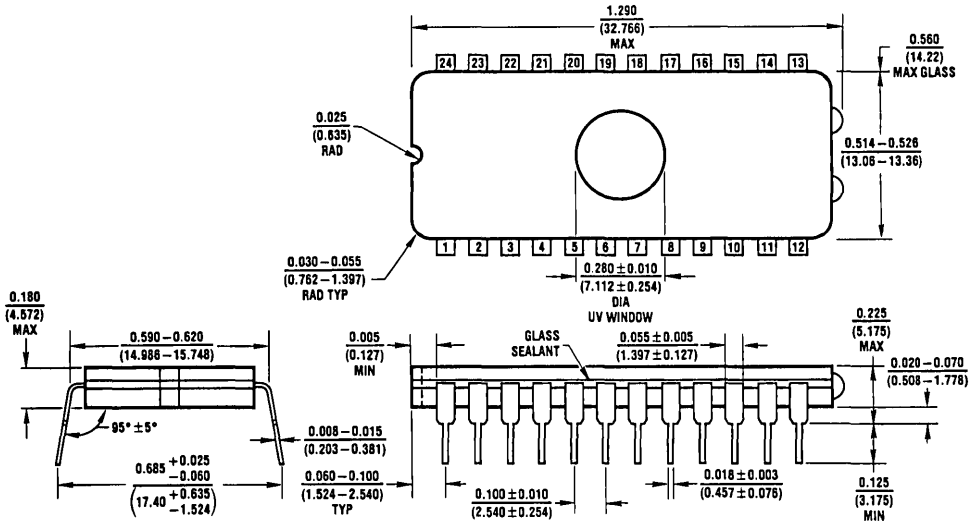
NS Package J14A



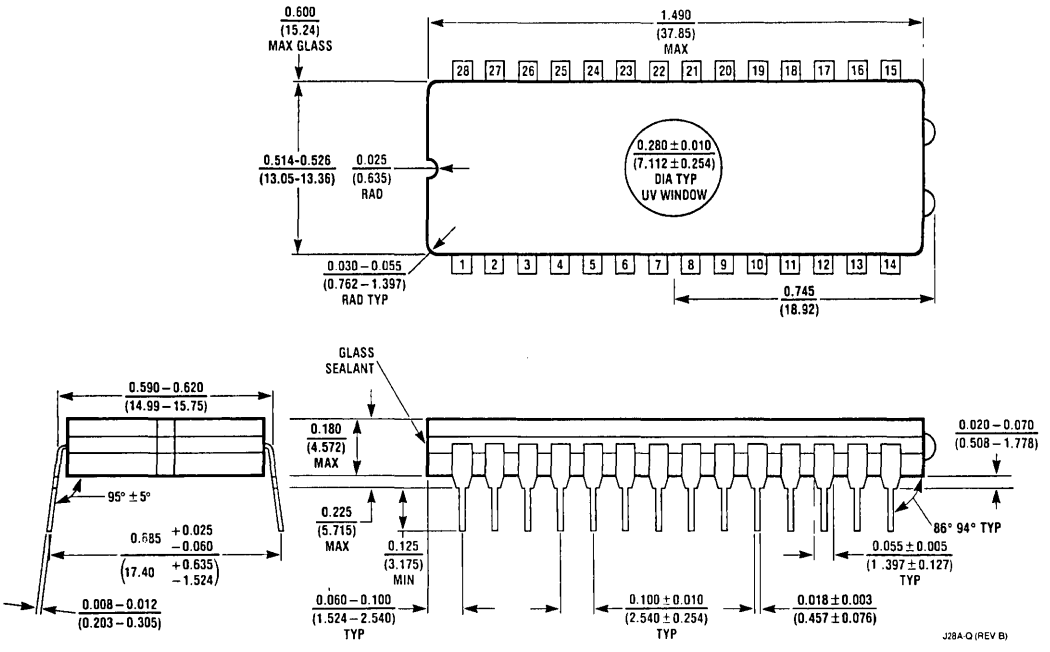
NS Package J20A



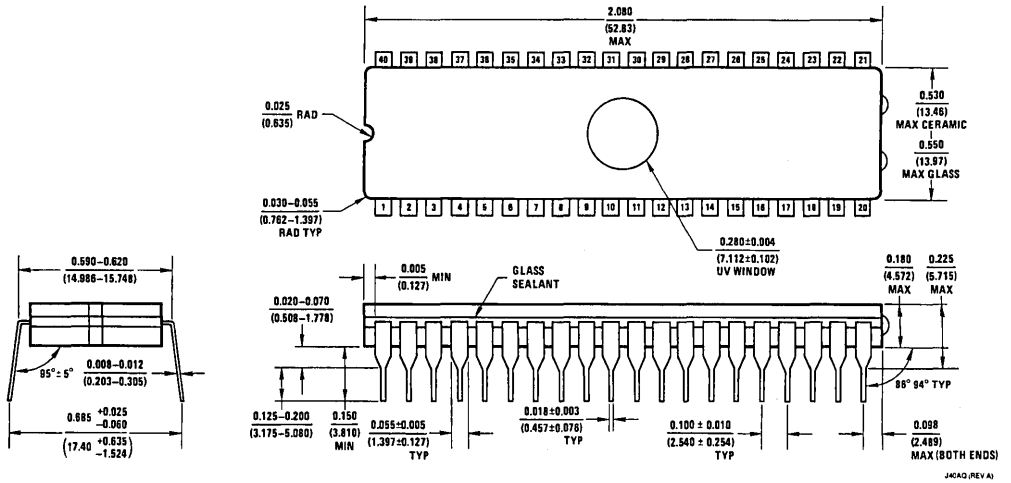
NS Package J24A



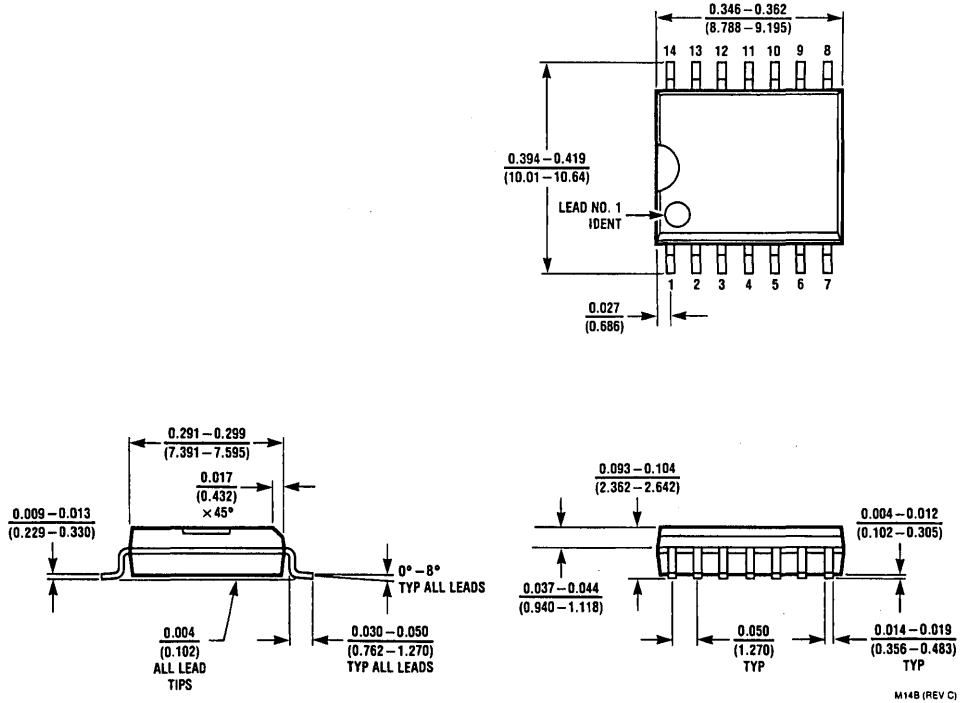
NS Package J24AQ



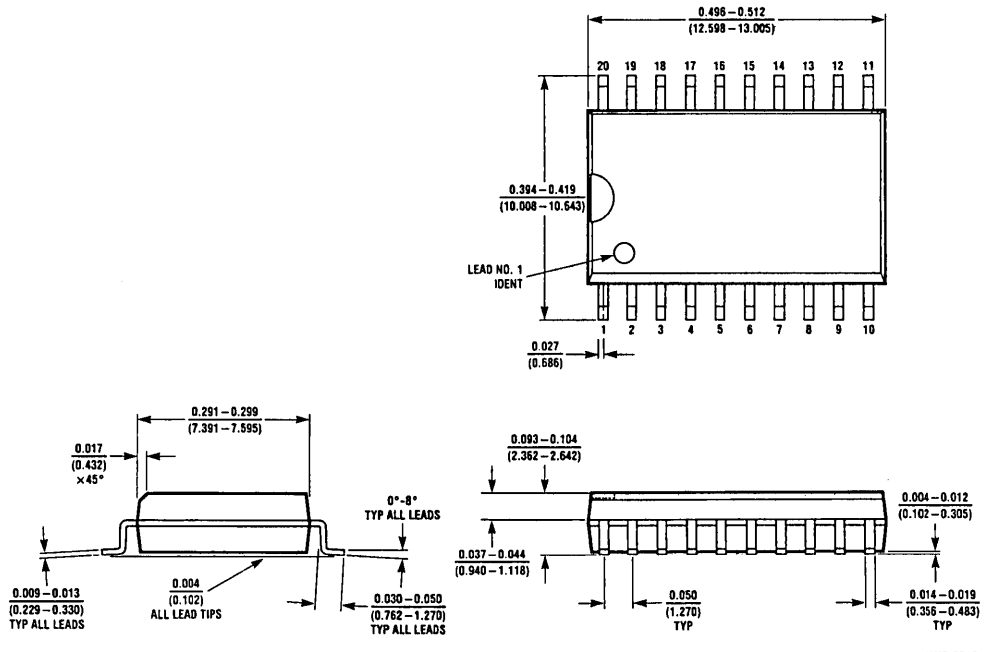
NS Package J28AQ



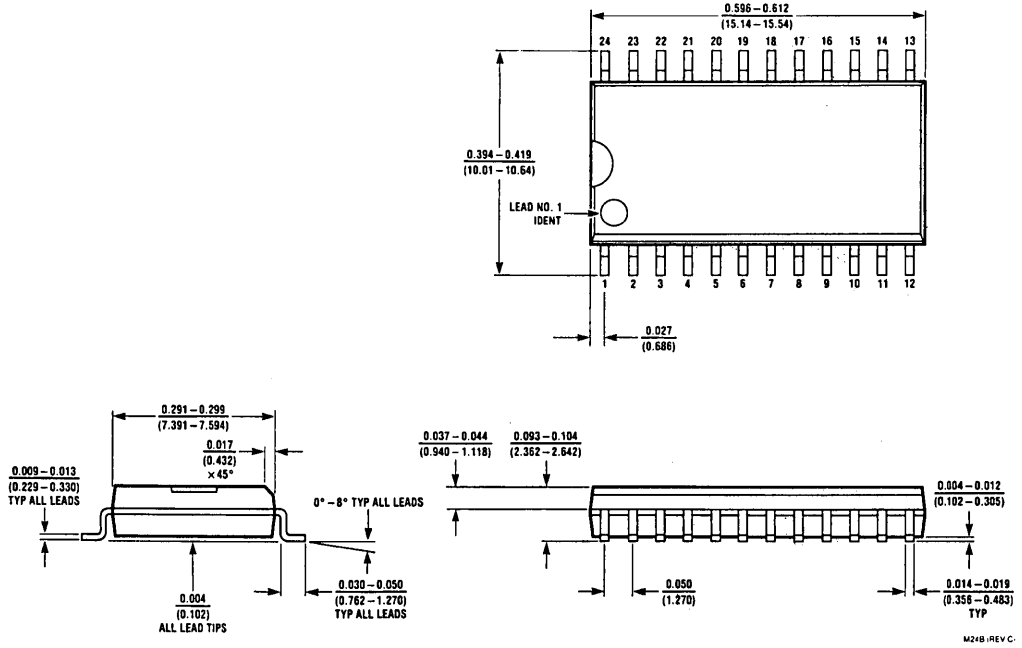
NS Package J40AQ



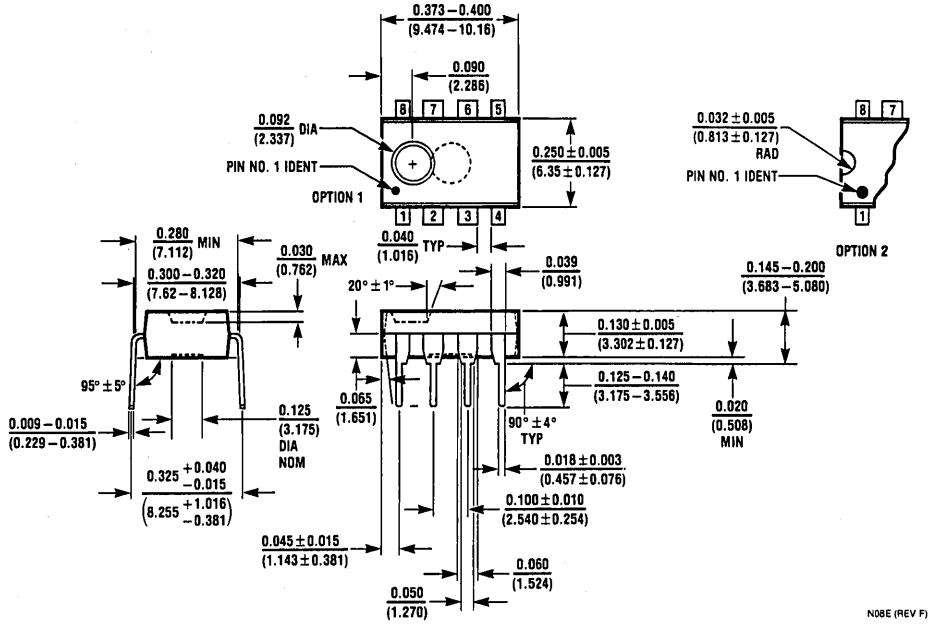
NS Package M14B



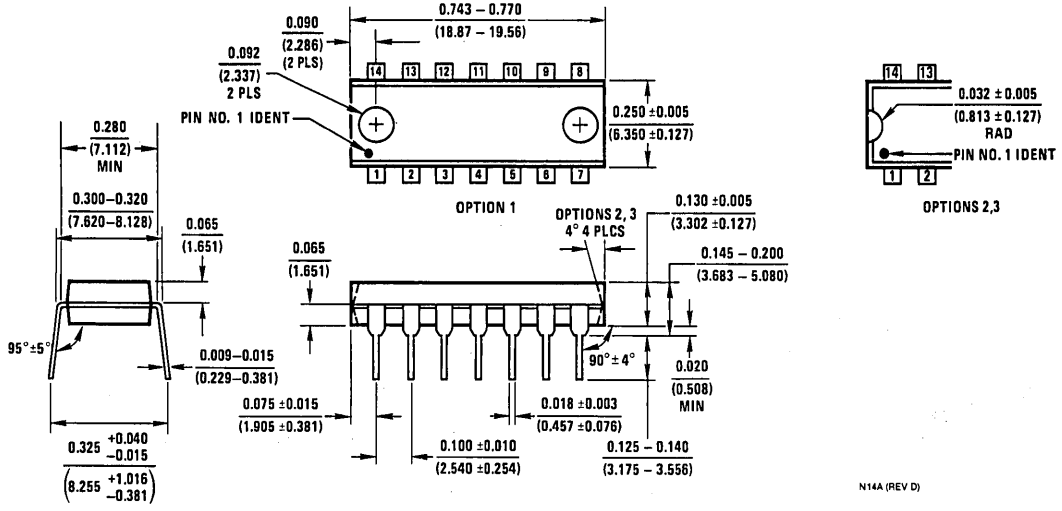
NS Package M20B



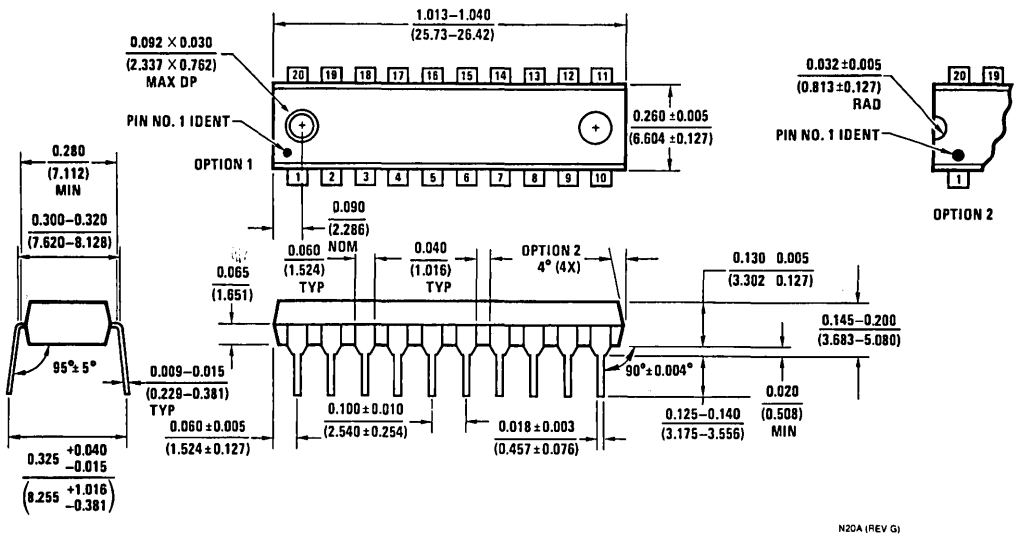
NS Package M24B



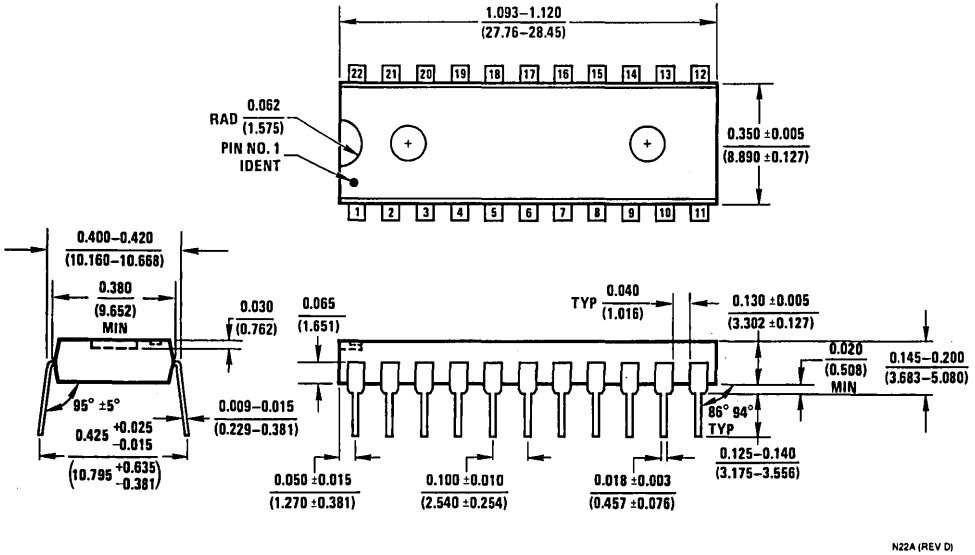
NS Package N08E



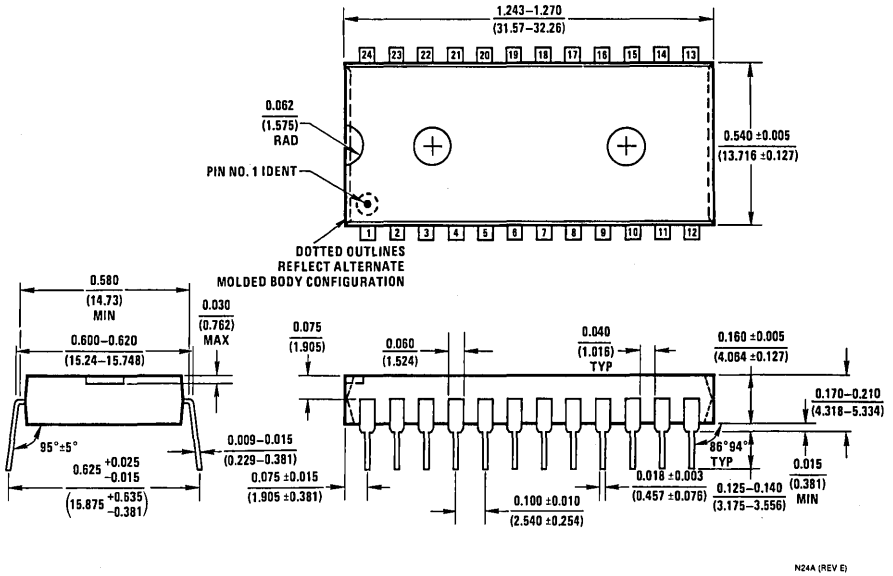
NS Package N14A



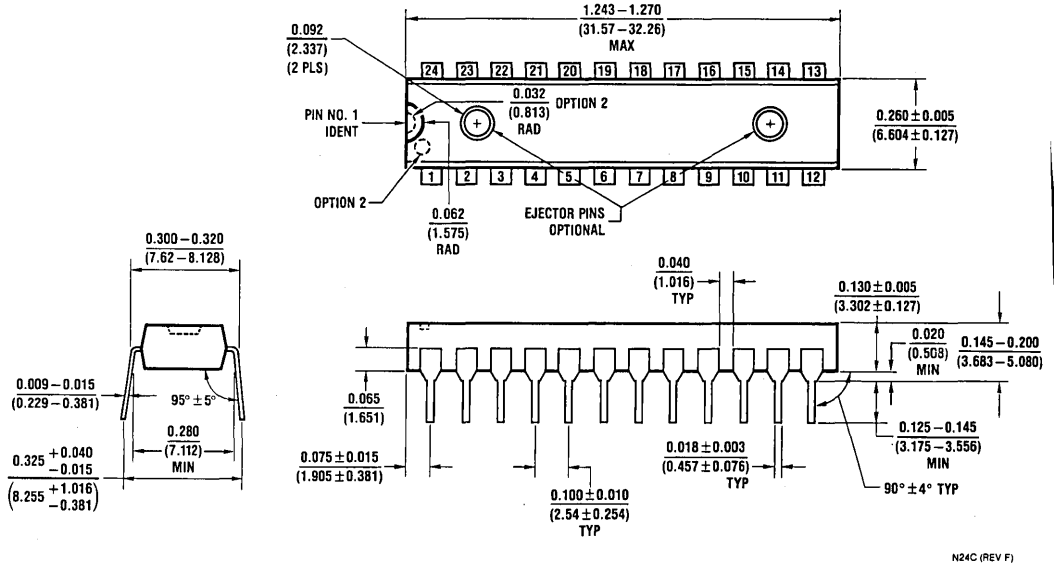
NS Package N20A



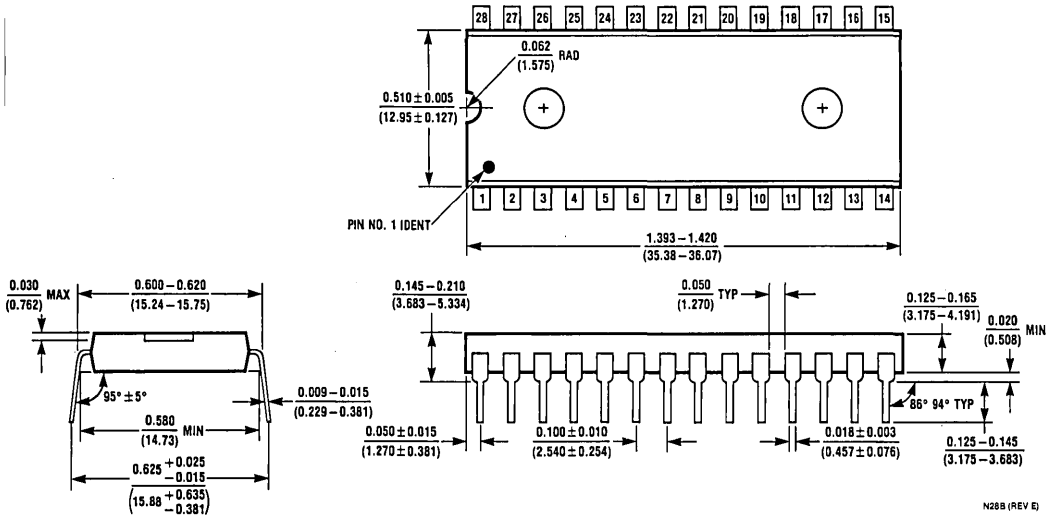
NS Package N22A



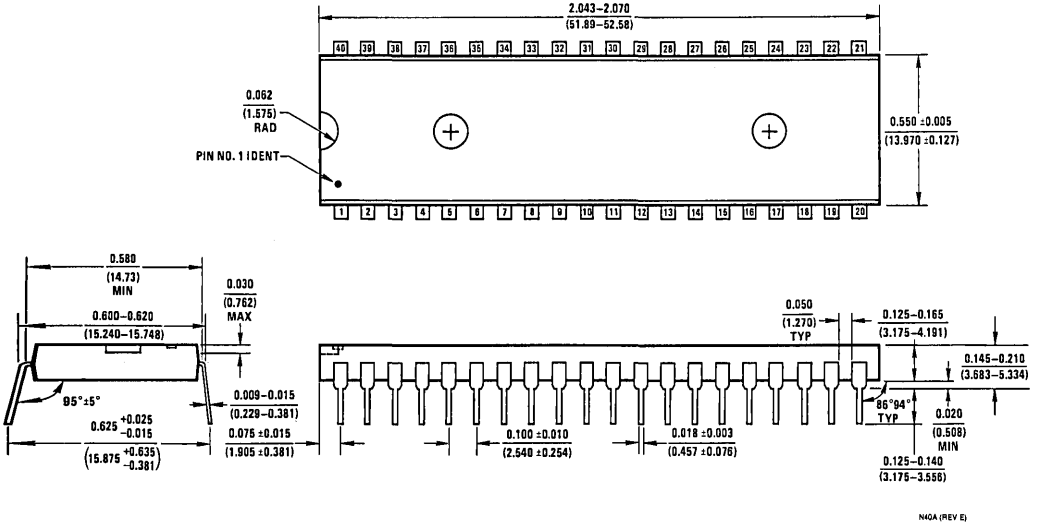
NS Package N24C



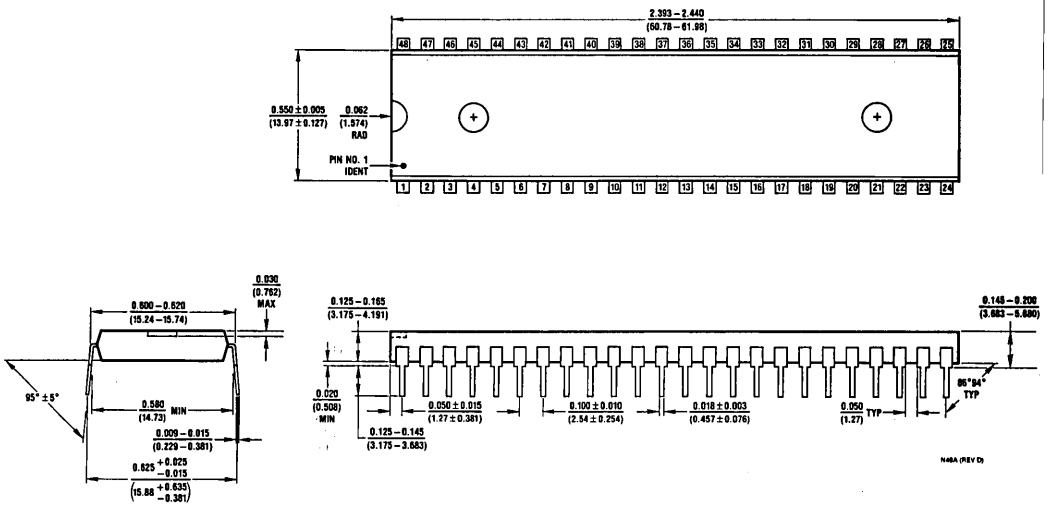
NS Package N24C



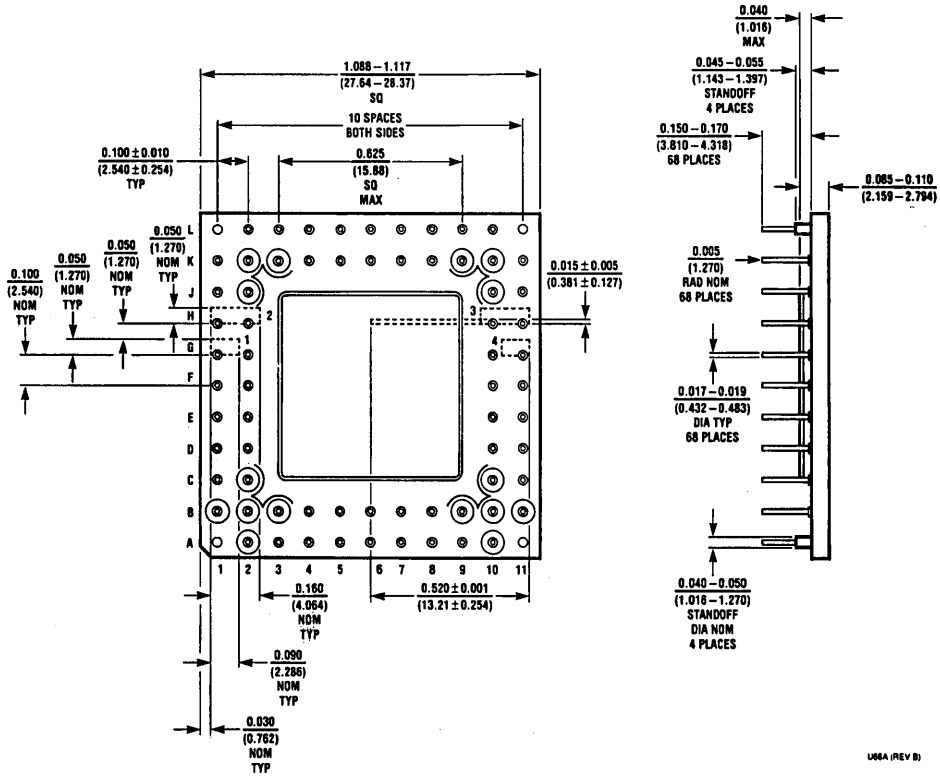
NS Package N28B



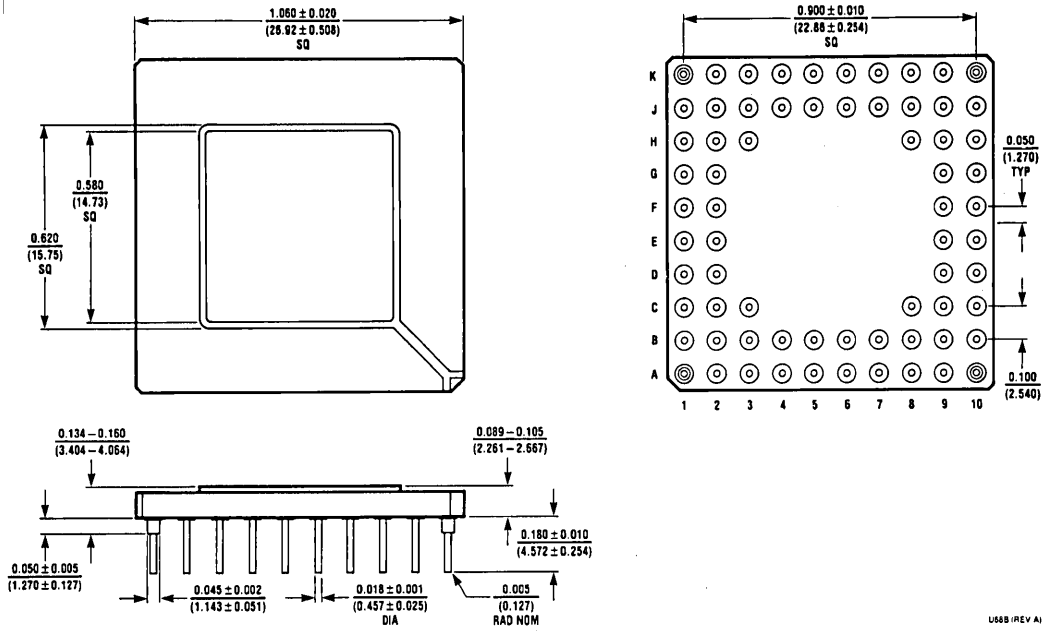
NS Package N40A



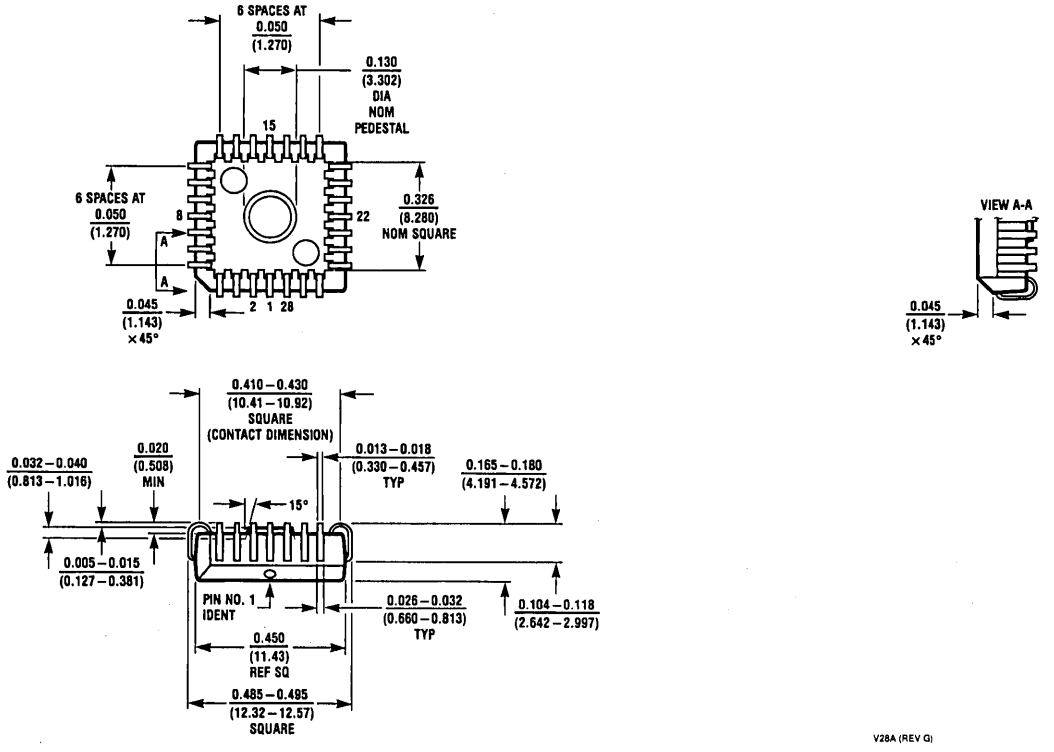
NS Package N48A



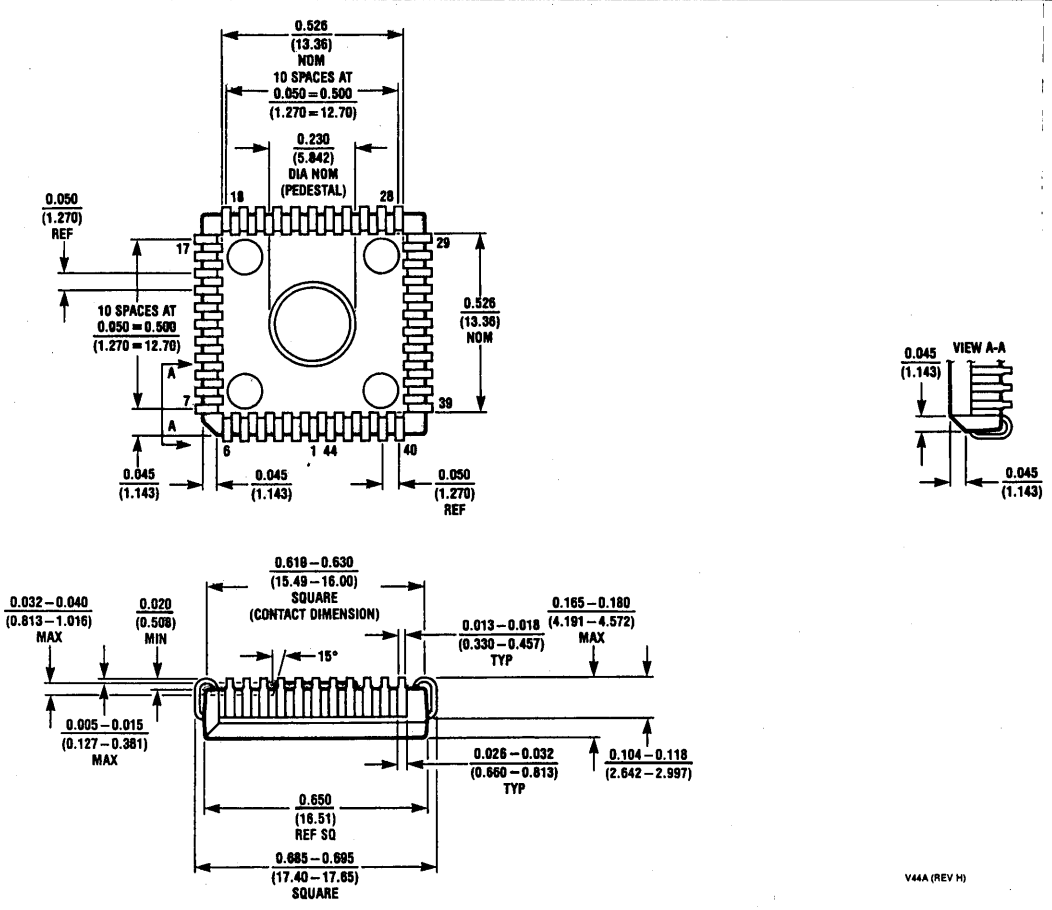
NS Package U68A



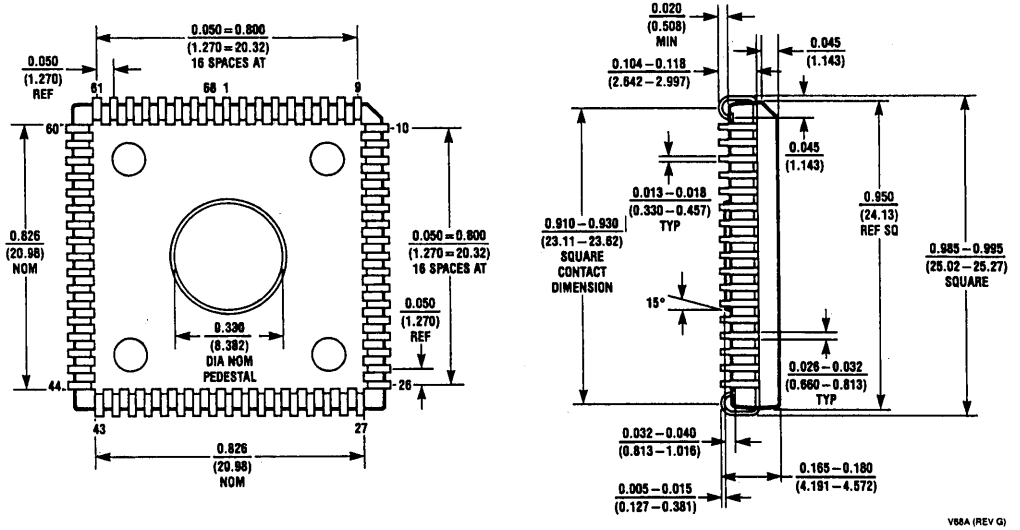
NS Package U68A



NS Package V28A



NS Package V44A



NS Package V68A

National Semiconductor Corporation

Bookshelf of Technical Support Information

National Semiconductor Corporation recognizes the need to keep you informed about the availability of current technical literature.

This bookshelf is a compilation of books that are currently available. The listing that follows shows the publication year and section contents for each book.

Please contact your local National sales office for possible complimentary copies. A listing of sales offices follows this bookshelf.

We are interested in your comments on our technical literature and your suggestions for improvement.

Please send them to:

Technical Communications Dept. M/S 23-200
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090

For a recorded update of this listing plus ordering information for these books from National's Literature Distribution operation, please call (408) 749-7378.

DATA CONVERSION/ACQUISITION DATABOOK—1984

Selection Guides • Active Filters • Amplifiers • Analog Switches • Analog-to-Digital Converters
Analog-to-Digital Display (DVM) • Digital-to-Analog Converters • Sample and Hold • Sensors/Transducers
Successive Approximation Registers/Comparators • Voltage References

HYBRID PRODUCTS DATABOOK—1982

Operational Amplifiers • Buffers • Instrumentation Amplifiers • Sample & Hold Amplifiers • Comparators
Non-Linear Functions • Precision Voltage Regulators and References • Analog Switches
MOS Clock Drivers • Digital Drivers • A-D Converters • D-A Converters • Fiber-Optic Products
Active Filters & Telecommunication Products • Precision Networks • 883/RETS

INTERFACE DATABOOK—1986

Transmission Line Drivers/Receivers • Bus Transceivers • Peripheral/Power Drivers • Display Controllers/Drivers
Memory Support • Microprocessor Support • Level Translators/Buffers • Frequency Synthesis

INTERFACE/BIPOLAR LSI/BIPOLAR MEMORY/PROGRAMMABLE LOGIC DATABOOK—1983

Transmission Line Drivers/Receivers • Bus Transceivers • Peripheral/Power Drivers
Level Translators/Buffers • Display Controllers/Drivers • Memory Support • Dynamic Memory Support
Microprocessor Support • Data Communications Support • Disk Support • Frequency Synthesis
Interface Appendices • Bipolar PROMs • Bipolar and ECL RAMs • 2900 Family/Bipolar Microprocessor
Programmable Logic

INTUITIVE IC CMOS EVOLUTION—1984

Thomas M. Frederiksen's new book targets some of the most significant transitions in semiconductor technology since the change from germanium to silicon. *Intuitive IC CMOS Evolution* highlights the transition in the reduction in defect densities and the development of new circuit topologies. The author's latest book is a vital aid to engineers, and industry observers who need to stay abreast of the semiconductor industry.

INTUITIVE IC OP AMPS—1984

Thomas M. Frederiksen's new book, *Intuitive IC Op Amps,* explores the many uses and applications of different IC op amps. Frederiksen's detailed book differs from others in the way he focuses on the intuitive groundwork in the basic functioning concepts of the op amp. Mr. Frederiksen's latest book is a vital aid to engineers, designers, and industry observers who need to stay abreast of the computer industry.

LINEAR APPLICATIONS HANDBOOK—1986

The purpose of this handbook is to provide a fully indexed and cross-referenced collection of linear integrated circuit applications using both monolithic and hybrid circuits from National Semiconductor.

Individual application notes are normally written to explain the operation and use of one particular device or to detail various methods of accomplishing a given function. The organization of this handbook takes advantage of this innate coherence by keeping each application note intact, arranging them in numerical order, and providing a detailed Subject Index.

LINEAR SUPPLEMENT DATABOOK—1984

Amplifiers • Comparators • Voltage Regulators • Voltage References • Converters • Analog Switches
Sample and Hold • Sensors • Filters • Building Blocks • Motor Controllers • Consumer Circuits
Telecommunications Circuits • Speech • Special Analog Functions

LOGIC DATABOOK VOLUME I—1984

CMOS AC Switching Test Circuits and Timing Waveforms • CMOS Application Notes • MM54HC/MM74HC
MM54HCT/MM74HCT • CD4XXX • MM54CXXX/MM74CXXX • LSI/VLSI

LOGIC DATABOOK VOLUME II—1984

Introduction to Bipolar Logic • Advanced Low Power Schottky • Advanced Schottky • Low Power Schottky
Schottky • TTL • Low Power

MASS STORAGE HANDBOOK—1986

Disk Interface Design Guide and User Manual • Winchester Disk Support • Winchester Disk Data Controller
Floppy Disk Support • Drive Interface Support Circuits

MEMORY SUPPORT HANDBOOK—1986

Dynamic Memory Control • Error Checking and Correction • Microprocessor Interface and Applications
Memory Drivers and Support

MOS MEMORY DATABOOK—1984

Standard Terminology • MOS Memory Cross Reference Guide • Dynamic RAMs • NMOS Static RAMs
CMOS Static RAMs • EPROMs • EEPROMs • Military/Aerospace • Reliability

THE NSC800 MICROPROCESSOR FAMILY DATABOOK—1985

CPU • Peripherals • Evaluation Board • Logic Devices • MA2000 Macrocomponent Family

SERIES 32000 DATABOOK—1986

Introduction • CPU-Central Processing Unit • Slave Processors • Peripherals • Data Communications and LAN's
Disk Control and Interface • DRAM Interface • Development Tools • Software Support • Application Notes

RELIABILITY HANDBOOK—1986

Reliability and the Die • Internal Construction • Finished Package • MIL-STD-883 • MIL-M-38510
The Specification Development Process • Reliability and the Hybrid Device • VLSI/VHSIC Devices
Radiation Environment • Electrostatic Discharge • Discrete Device • Standardization
Quality Assurance and Reliability Engineering • Reliability and Documentation • Commercial Grade Device
European Reliability Programs • Reliability and the Cost of Semiconductor Ownership
Reliability Testing at National Semiconductor • The Total Military/Aerospace Standardization Program
883B/RETS™ Products • MILS/RETS™ Products • 883/RETS™ Hybrids • MIL-M-38510 Class B Products
Radiation Hardened Technology • Wafer Fabrication • Semiconductor Assembly and Packaging
Semiconductor Packages • Glossary of Terms • Key Government Agencies • AN/ Numbers and Acronyms
Bibliography • MIL-M-38510 and DESC Drawing Cross Listing

THE SWITCHED-CAPACITOR FILTER HANDBOOK—1985

Introduction to Filters • National's Switched-Capacitor Filters • Designing with Switched-Capacitor Filters
Application Circuits • Filter Design Program • Nomographs and Tables

TRANSISTOR DATABOOK—1982

NPN Transistors • PNP Transistors • Junction Field Effect Transistors • Selection Guides • Pro Electron Series
Consumer Series • NA/NB/NR Series • Process Characteristics Double-Diffused Epitaxial Transistors
Process Characteristics Power Transistors • Process Characteristics JFETs • JFET Applications Notes

VOLTAGE REGULATOR HANDBOOK—1982

Product Selection Procedures • Heat Flow & Thermal Resistance • Selection of Commercial Heat Sink
Custom Heat Sink Design • Applications Circuits and Descriptive Information • Power Supply Design
Data Sheets

48-SERIES MICROPROCESSOR HANDBOOK—1980

The 48-Series Microcomputers • The 48-Series Single-Chip System • The 48-Series Instruction Set
Expanding the 48-Series Microcomputers • Applications for the 48-Series • Development Support
Analog I/O Components • Communications Components • Digital I/O Components • Memory Components
Peripheral Control Components

NATIONAL SEMICONDUCTOR CORPORATION AUTHORIZED DISTRIBUTORS

ALABAMA
Huntsville
Hall-Mark
(205) 837-8700
Hamilton/Avnet
(205) 837-7210
Pioneer
(205) 837-9300
Schweber
(205) 882-2200

ARIZONA
Phoenix
Schweber
(602) 997-4874
Tempe
Anthem Electronics
(602) 966-6600
Hamilton/Avnet
(602) 231-5100

CALIFORNIA
Canoga Park
Hall-Mark
(818) 716-7300
Schweber
(818) 999-4702
Chatsworth
Anthem Electronics
(818) 700-1000
Avnet Electronics
(818) 700-8668
Hamilton Electro Sales
(818) 700-6050
Citrus Heights
Hall-Mark
(916) 722-8600
Costa Mesa
Avnet Electronics
(714) 754-6050
Hamilton Electro Sales
(714) 641-4159
Culver City
Hamilton Electro Sales
(213) 558-2121
Garden Grove
Bell Industries
(714) 895-7801
Gardena
Bell Industries
(213) 515-1800
Irvine
Anthem Electronics
(714) 768-4444
Schweber
(714) 863-0200
Roseville
Bell Industries
(916) 969-3100
Sacramento
Hamilton/Avnet
(916) 925-2216
Schweber
(916) 929-9732
San Diego
Anthem Electronics
(619) 279-5200
Hamilton/Avnet
(619) 571-7510
San Jose
Anthem Electronics
(408) 946-8000
Hall-Mark
(408) 946-0900
Schweber
(408) 946-7171
Sunnyvale
Bell Industries
(408) 734-8570
Hamilton/Avnet
(408) 743-3355
Thousand Oaks
Bell Industries
(805) 499-6821
Tustin
Hall-Mark
(714) 669-4700

COLORADO
Englewood
Anthem Electronics
(303) 790-4500
Hamilton/Avnet
(303) 779-9998
Schweber
(303) 799-0258
Wheatridge
Bell Industries
(303) 424-1985

CONNECTICUT
Danbury
Hamilton/Avnet
(203) 797-2800
Schweber
(203) 748-7080
Meridian
Lionex Inc.
(203) 237-2282
Norwalk
Pioneer Northeast
(203) 853-1515

FLORIDA
Altamonte Springs
Pioneer
(305) 834-9090
Schweber
(305) 331-7555
Clearwater
Hall-Mark
(813) 530-4543
Deerfield
Pioneer
(305) 428-8877
Ft. Lauderdale
Hamilton/Avnet
(305) 971-2900
Hollywood
Schweber
(305) 927-0511
Orlando
Hall-Mark
(305) 855-4020
Pompano Beach
Hall-Mark
(305) 971-9280
Winter Park
Hamilton-Avnet
(305) 628-3888

GEORGIA
Norcross
Hall-Mark
(404) 447-8000
Hamilton-Avnet
(404) 447-7500
Pioneer
(404) 448-1711
Schweber
(404) 449-9170

ILLINOIS
Bensenville
Hamilton/Avnet
(312) 860-7780
Chicago
Bell Industries
(312) 982-9210
Elk Grove Village
Pioneer
(312) 952-8440
Schweber
(312) 364-3750
Wood Dale
Hall-Mark
(312) 860-3800

INDIANA
Carmel
Hamilton/Avnet
(317) 844-9333
Indianapolis
Advent
(317) 872-4910
Pioneer
(317) 849-7300

IOWA
Cedar Rapids
Advent Electronics
(319) 363-0221
Bell Industries
(319) 395-0730
Hamilton/Avnet
(319) 362-4757
Schweber
(319) 373-1417

KANSAS
Lenexa
Hall-Mark
(913) 888-4747
Overland Park
Hamilton/Avnet
(913) 888-8900
Schweber
(913) 492-2921

MARYLAND
Columbia
Hall-Mark
(301) 988-9800
Hamilton/Avnet
(301) 995-3500
Gaithersburg
Pioneer Washington
(301) 921-0660
Schweber
(301) 840-5900

MASSACHUSETTS
Bedford
Schweber
(617) 275-5100
Billerica
Hall-Mark
(617) 935-9777
Lexington
Pioneer Northeast
(617) 861-9200
Wilmington
Lionex
(617) 657-5170
Woburn
Hamilton/Avnet
(617) 273-7500

MICHIGAN
Grand Rapids
Hamilton/Avnet
(616) 243-8805
R-M Michigan
(616) 531-9300
Livonia
Hamilton/Avnet
(313) 522-4700
Pioneer
(313) 525-1800
Schweber
(313) 525-8100

MINNESOTA
Bloomington
Hall-Mark
(612) 854-3223
Edina
Schweber
(612) 941-5280
Minnetonka
Hamilton/Avnet
(612) 932-0600
Pioneer
(612) 935-5444

MISSOURI
Earth City
Hamilton/Avnet
(314) 344-1200
Schweber
(314) 739-0526
Maryland Heights
Hall-Mark
(314) 291-5350

NEW HAMPSHIRE
Manchester
Hamilton/Avnet
(603) 624-9400
Schweber
(603) 625-2250

NEW JERSEY
Cherry Hill
Hall-Mark
(609) 424-7300
Hamilton/Avnet
(609) 424-0100
Fairfield
Hall-Mark
(201) 575-4415
Hamilton/Avnet
(201) 575-3390
Lionex
(201) 227-7960
Schweber
(201) 227-7880
Pine Brook
Pioneer Northeast
(201) 575-3510

NEW MEXICO
Albuquerque
Alliance Electronics
(505) 292-3360
Bell/Century
(505) 292-2700
Hamilton/Avnet
(505) 765-1500

NATIONAL SEMICONDUCTOR CORPORATION AUTHORIZED DISTRIBUTORS (Continued)

NEW YORK
Buffalo
Summit Distributors
(716) 887-2800
East Syracuse
Hamilton/Avnet
(315) 437-2642
Fairport
Pioneer
(716) 381-7070
Hauppauge
Hamilton/Avnet
(516) 434-7413
Lionex
(516) 273-1660
Rochester
Hamilton/Avnet
(716) 475-9130
Schweber
(716) 424-2222
Summit Electronics
(716) 334-8110
Ronkonkoma
Hall-Mark
(516) 737-0600
Vestal
Pioneer
(607) 748-8211
Westbury
Hamilton/Avnet
(516) 997-6868
Schweber
(516) 334-7474
Woodbury
Pioneer
(516) 921-8700

NORTH CAROLINA
Charlotte
Pioneer
(704) 527-8188
Raleigh
Hall-Mark
(919) 872-0712
Hamilton/Avnet
(919) 878-0810
Schweber
(919) 876-0000

OHIO
Beechwood
Schweber
(216) 464-2970
Cleveland
Hamilton/Avnet
(216) 831-3500
Pioneer
(216) 587-3600
Dayton
Bell Industries
(513) 435-8660
Hamilton/Avnet
(513) 439-6700
Pioneer
(513) 236-9900
Highland Heights
CAM/OHIO
(216) 461-4700
Solon
Hall-Mark
(216) 349-4632
Westerville
Hall-Mark
(614) 891-4555

OKLAHOMA
Tulsa
Hall-Mark
(918) 665-3200
Quality Components
(918) 664-8812
Radio Inc.
(918) 587-9123
Schweber
(918) 622-8000

OREGON
Beaverton
Almac Electronics
(503) 641-9070
Lake Oswego
Anthem Electronics
(503) 684-2661
Bell Industries
(503) 241-4115
Hamilton/Avnet
(503) 635-7850

PENNSYLVANIA
Horsham
Lionex
(215) 443-5150
Pioneer
(215) 674-4000
Schweber
(215) 441-0600
Pittsburgh
CAM/RPC
(412) 782-3770
Pioneer
(412) 782-2300
Schweber
(412) 782-1600

TEXAS
Addison
Quality Components
(214) 733-4300
Austin
Hall-Mark
(512) 258-8848
Hamilton/Avnet
(512) 837-8911
Pioneer
(512) 835-4000
Quality Components
(512) 835-0220
Schweber
(512) 458-8253
Dallas
Hall-Mark
(214) 553-4300
Pioneer
(214) 386-7300
Schweber
(214) 661-5010
Houston
Hall-Mark
(713) 781-6100
Hamilton/Avnet
(713) 780-1771
Pioneer
(713) 988-5555
Schweber
(713) 784-3600
Irving
Hamilton/Avnet
(214) 659-4151
Sugarland
Quality Components
(713) 491-2255

UTAH
Salt Lake City
Anthem Electronics
(801) 973-8555
Bell Industries
(801) 972-6969
Hamilton/Avnet
(801) 972-4300

WASHINGTON
Bellevue
Hamilton/Avnet
(206) 453-5844
Redmond
Anthem Electronics
(208) 881-0850
Seattle
Almac Electronics
(206) 643-9992

WISCONSIN
Brookfield
Schweber
(414) 784-9020
Milwaukee
Taylor Electric Co.
(414) 241-4321
New Berlin
Hall-Mark
(414) 761-3000
Hamilton/Avnet
(414) 784-4516
Waukesha
Bell Industries
(414) 547-8879

CANADA
Western Provinces
Calgary
Hamilton/Avnet
(403) 230-3586
Zentronics
(403) 272-1021
Edmonton
Zentronics
(403) 463-3014
Richmond
Zentronics
(604) 273-5575
Winnipeg
Zentronics
(204) 775-8661

Eastern Provinces
Brampton
Zentronics
(416) 451-9600
Doval
Semad
(514) 636-4614
Markham
Semad
(416) 475-8500
Missisauga
Hamilton/Avnet
(416) 677-7432
Nepean
Hamilton/Avnet
(613) 226-1700
Ottawa
Semad
(613) 729-6145
Zentronics
(613) 238-6411
St. Laurent
Zentronics
(514) 735-5361
Waterloo
Zentronics
(519) 884-5700

SALES OFFICES AND REPRESENTATIVES

ALABAMA
Huntsville
New Interep Associates
(205) 533-1730
Mobile
New Interep Associates
(205) 478-1036

ARIZONA
Temple
National Semiconductor
(602) 966-4563
Phoenix
Desert Technical Sales
(602) 244-0735

CALIFORNIA
Santa Clara
National Semiconductor
(408) 730-3054
Bay Tech Sales
(408) 730-3009
Roseville
Bay Tech Sales
(916) 969-5577
Woodland Hills
National Semiconductor
(818) 888-2602
Encino
Great American Rep. Co.
(818) 990-4870
Irvine
National Semiconductor
(714) 250-1440
Santa Ana
Great American Rep. Co.
(714) 954-0371
San Diego
National Semiconductor
(619) 279-0724

COLORADO
Englewood
National Semiconductor
(303) 790-8090
Aurora
Skyline Technical Sales
(303) 360-0955

CONNECTICUT
Ridgefield
National Semiconductor
(203) 431-8182
Fairfield
NRG Limited
(203) 384-1112

FLORIDA
No. Miami Beach
National Semiconductor
(305) 947-0031
St. Petersburg
QXI
(813) 821-2281
Boca Raton
QXI
(305) 392-2626

GEORGIA
Atlanta
National Semiconductor
(404) 393-2626
Action Component Sales
(404) 393-9494

ILLINOIS
Schaumburg
National Semiconductor
(312) 397-8777
Arlington Heights
Delta Technical Sales
(312) 253-9440

INDIANA
Carmel
National Semiconductor
(317) 843-7160
Indianapolis
Advanced Component Sales
(317) 545-6441
Fort Wayne
Advanced Component Sales
(219) 484-0722

IOWA
Cedar Rapids
Stan Clothier Company
(319) 395-0245

MARYLAND
Hanover
National Semiconductor
(301) 796-8900
Glen Burnie
TRIMARK, Inc.
(301) 761-6000

MASSACHUSETTS
Burlington
National Semiconductor
(617) 273-3170
Lexington
A/D Systems Sales, Inc.
(617) 861-6371

MICHIGAN
W. Bloomfield
National Semiconductor
(313) 855-0166

MINNESOTA
Bloomington
National Semiconductor
(612) 854-8200
Minneapolis
Stan Clothier Company
(612) 944-3456

MISSOURI
Kansas City
National Semiconductor
(816) 941-3537
Raytown
Cen Tech
(816) 358-8100
Bridgeton
Cen Tech
(314) 291-4230

NEW JERSEY
Paramus
National Semiconductor
(201) 599-0955

NEW MEXICO
Albuquerque
Reptronix, Inc.
(505) 292-1718

NEW YORK
Fairport
National Semiconductor
(716) 425-1358
Syracuse
Electra Sales Corp.
(315) 463-1248
Rochester
Electra Sales Corp.
(716) 461-5252
Poughkeepsie
National Semiconductor
(914) 473-8830
Melville
Parallax Sales
(516) 351-1000

NORTH CAROLINA
Durham
National Semiconductor
(919) 683-2676
Cary
Engineering Devices Corp.
(919) 469-9323

OHIO
Highland Heights
National Semiconductor
(216) 461-0191
Micro-Tec, Inc.
(216) 442-1555
Dayton
National Semiconductor
(513) 435-6886
Micro-Tec, Inc.
(513) 435-0644
Columbus
Micro-Tec, Inc.
(614) 451-2400

OREGON
Beaverton
Meritech, Inc.
(503) 644-0304

PENNSYLVANIA
Ft. Washington
National Semiconductor
(215) 643-4910
Huntingdon Valley
Omega Electronic Sales
(215) 947-4135

TENNESSEE
Knoxville
Action Component Sales
(615) 694-0140

TEXAS
Richardson
National Semiconductor
(214) 690-4552
Interactive Component
Sales
(214) 669-4031
Austin
National Semiconductor
(512) 339-7555
Interactive Component
Sales
(512) 451-0201
Houston
Interactive Component
Sales
(713) 270-6141

UTAH
Salt Lake City
Skyline Technical Sales
(801) 261-5402

WASHINGTON
Bellevue
National Semiconductor
(206) 453-9944
Meritech, Inc.
(206) 454-4600

WISCONSIN
Milwaukee
Delta Technical Sales
(414) 527-3800

CANADA
Mississauga
National Semiconductor
(416) 678-2920
Brampton
Canadian Micro Sales
(416) 453-9121
Lachine
Canadian Micro Sales
(514) 636-8525
Nepean
Canadian Micro Sales
(613) 596-0411

PUERTO RICO
Puerto Nuevo
National Semiconductor
(809) 792-9110

National Semiconductor Corporation

National Semiconductor Corporate Headquarters
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: (408) 721-5000
TWX: (910) 339-9240

SALES OFFICES AND REPRESENTATIVES (Continued)

INTERNATIONAL OFFICES

Electronica NSC de Mexico SA
Juventino Rosas No. 118-2
Col Guadalupe Inn
Mexico, 01020 D.F. Mexico
Tel: (905) 524-9402

National Semicondutores Do Brasil Ltda.
Av. Brig. Faria Lima, 830
8 Andar
01452 Sao Paulo, SP. Brasil
Tel: (55/11) 212-5066
Telex: 391-1131931 NSBR BR

National Semiconductor GmbH
Industriestraße 10
D-8080 Fürstenfeldbruck
West Germany
Tel: (0 8141) 103-0
Telex: 527 649

National Semiconductor (UK) Ltd.
301 Harpur Centre
Horne Lane
Bedford MK40 1TR
United Kingdom
Tel: 0234-47147
Telex: 826 209

National Semiconductor Benelux
Ave Charles Quint 545
B-1080 Bruxelles
Belgium
Tel: (02) 4661807
Telex: 61007

National Semiconductor (UK) Ltd.
1, Bianco Lunos Alle
DK-1868 Copenhagen V
Denmark
Tel: (01) 213211
Telex: 15179

National Semiconductor
Expansion 10000
28, Rue de la Redoute
F-92 260 Fontenay-aux-Roses
France
Tel: (01) 660-8140
Telex: 250956

National Semiconductor S.p.A.
Via Solferino 19
20121 Milano
Italy
Tel: (02) 345-2046/7/8/9
Telex: 332835

National Semiconductor AB
Box 2016
Stensätravägen 13
S-12702 Skärholmen
Sweden
Tel: (08) 97 01 90
Telex: 10 731

National Semiconductor GmbH
Calle Agustin de Foxa, 27 (9.D)
28036 Madrid
Spain
Tel: (01) 7 33 29 58
Telex: 46 133

National Semiconductor Switzerland
Alte Winterthurerstrasse 53
Postfach 567
CH-8304 Wallisellen-Zurich
Tel: (01) 830-2727
Telex: 59000

National Semiconductor
Pasilanraitio 6C
SF-00240 Helsinki 24
Finland
Tel: (90) 14 03 44
Telex: 124854

NS Japan Ltd.
4-403 Ikebukuro, Toshima-ku
Tokyo 171, Japan
Tel: (03) 988-2131
Fax: 011-81-3-988-1700

National Semiconductor Hong Kong Ltd.
Southeast Asia Marketing
Austin Tower, 4th Floor
22-26A Austin Avenue
Tsimshatsui, Kowloon, H.K.
Tel: 3-7231290
Cable: NSSEAMKTG
Telex: 52996 NSSEA HX

National Semiconductor Advanced Systems PTY LTD
11-17 Khartoum Road
North Ryde, N.S.W. 2113
Sydney, Australia
Tel: (03) 729-6333
Telex: AA32096

National Semiconductor (PTE), Ltd.
51 Goldhill Plaza, No. 10-01
Newton Road
Singapore 1130
Tel: 2506884
Telex: RS 33877

National Semiconductor (Far East) Ltd.
Taiwan Branch
P.O. Box 68-332 Taipei
7th Floor, Nan Shan Life Bldg.
302 Min Chuan East Road,
Taipei, Taiwan R.O.C.
Tel: (02) 501-7227
Telex: 22837 NSTW
Cable: NSTW TAIPEI

National Semiconductor (Far East) Ltd.
Korea Office
Third Floor, Hankyung Bldg.
4-25 Hannam-Dong
Yongsan-Ku, Seoul 140, Korea
Tel: 797-8001/3
Telex: K24942 NSRK

 TL1934 RRD-RRD75M017/Printed in U.S.A