TEXAS
INSTRUMENTS

First-Generation TMS320

User's Guide

1989

Digital Signal Processor Products

User's Guide TMS320C1x

1989

First-Generation TMS320 User's Guide

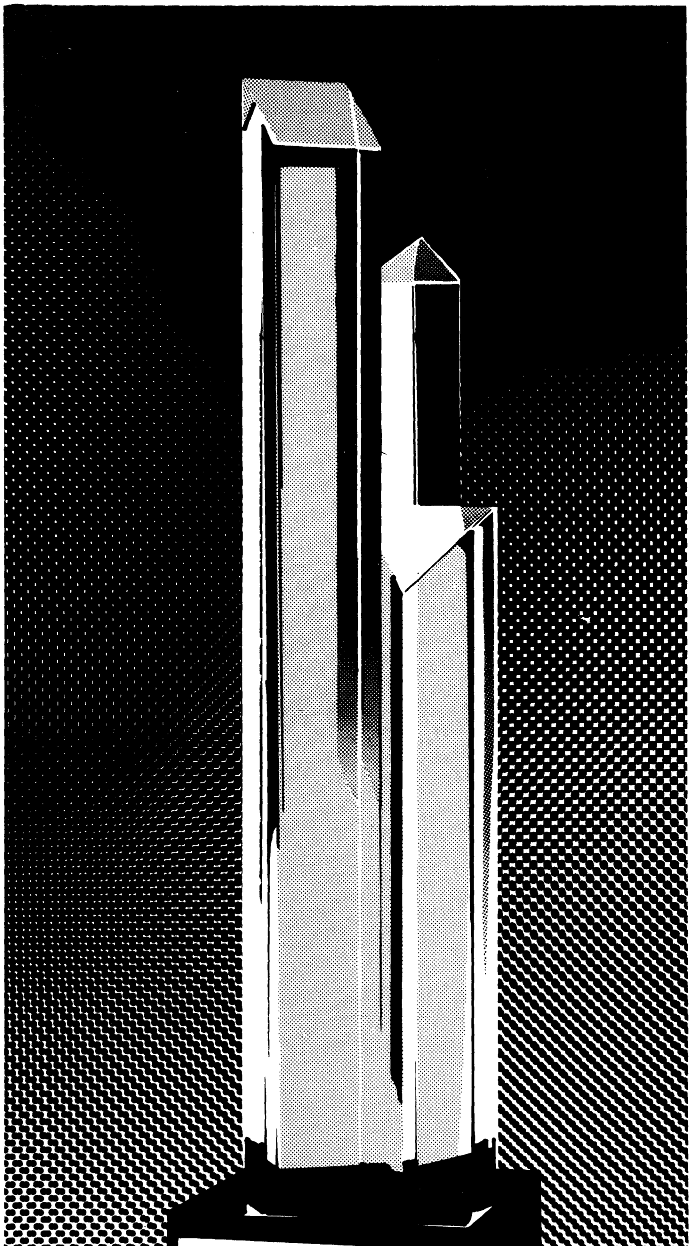
Texas Instruments

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to or to discontinue any semiconductor product or service identified in this publication without notice. TI advises its customers to obtain the latest version of the relevant information to verify, before placing orders, that the information being relied upon is current.

TI warrants performance of its semiconductor products to current specifications in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Unless mandated by government requirements, specific testing of all parameters of each device is not necessarily performed.

TI assumes no liability for TI applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.


ELECTRONIC PRODUCTS 1983
PRODUCT OF THE YEAR AWARD
TMS320 DIGITAL SIGNAL PROCESSOR
TEXAS INSTRUMENTS

Contents

Section *Page*

Illustrations

Figure *Page*

Tables

Table *Page*

Section 1

Introduction

The TMS320 family of 16/32-bit single-chip digital signal processors combines the flexibility of a high-speed controller with the numerical capability of an array processor, offering an inexpensive alternative to custom VLSI and multichip bit-slice processors.

The TMS32010, the first digital signal processor in the TMS320 family, was introduced in 1983. During that year, the TMS32010 was named "Product of the Year" by the magazine, *Electronic Products.* Its powerful instruction set, inherent flexibility, high-speed number-crunching capabilities, and innovative architecture have made this high-performance, cost-effective processor the ideal solution to many telecommunications, computer, commercial, industrial, and military applications.

The TMS320 family has now expanded into three generations of processors: TMS320C1x, TMS320C2x, and TMS320C3x (see Figure 1-1). Many features are common among these generations. Some specific features are added in each processor to provide different cost/performance tradeoffs. Software compatibility is maintained throughout the family to protect the user's investment in architecture. Each processor has software and hardware tools to facilitate rapid design.

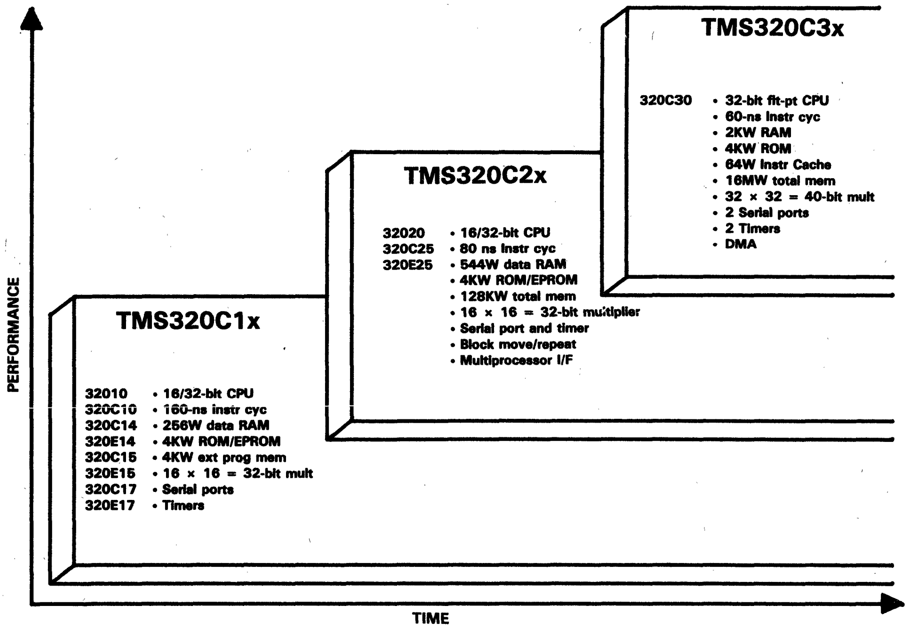


Figure 1-1. TMS320 Device Evolution

Throughout this document, the first-generation device group within the TMS320 family will be referred to as TMS320C1x. The specific members of the first-generation TMS320 include:

- TMS32010, the first 20-MHz digital signal processor
- TMS320C10, a CMOS 20-MHz version of the TMS32010
- TMS320C10-14, a 14-MHz version of the TMS320C10
- TMS320C10-25, a 25-MHz version of the TMS320C10
- TMS320C14, a TMS320C15 designed for a control system
- TMS320E14, an EPROM version of the TMS320C14
- TMS320C15, a TMS320C10 with expanded ROM and RAM
- TMS320C15-25, a 25-MHz version of the TMS320C15
- TMS320E15, an EPROM version of the TMS320C15
- TMS320E15-25, a 25-MHz version of the TMS320E15
- TMS320C17, a TMS320C15 with serial and coprocessor ports
- TMS320E17, an EPROM version of the TMS320C17

This document describes the core CPU, memory, and basic I/O port architecture of the first-generation devices (TMS320C1x) in the TMS320 family. The peripherals for the TMS320C17/E17 devices are also described. For descriptions of the TMS320C14/E14 as well as software and hardware applications/examples, refer to the *TMS320C14/TMS320E14 User's Guide* (literature number SPRU032).

The TMS320 family combines the high performance and specialized features necessary in digital signal processing (DSP) applications with an extensive program of development support, including hardware and software development tools, product documentation, textbooks, newsletters, DSP design workshops, and a variety of application reports. See Appendix E for a discussion of the wide range of development tools available.

Plans for expansion of the TMS320 family include more spinoffs of the existing generations as well as more powerful future generations of digital signal processors.

1.1 General Description

The combination of the TMS320's Harvard-type architecture (separate program and data buses) and its special digital signal processing (DSP) instruction set provides speed and flexibility to produce a microprocessor family capable of executing 6.25 MIPS (million instructions per second). While other processors implement functions through software or microcode, the TMS320 family optimizes performance by implementing functions within the hardware. This hardware-intensive approach provides the design engineer with power previously unavailable on a single chip.

Table 1-1 provides an overview of the TMS320C1x group of processors with comparisons of technology, memory, I/O, cycle timing, package type, and military support.

Table 1-1. TMS320C1x Processors Overview

DEVICE	TECH	MEMORY ON-CHIP			MEMORY OFF-CHIP	I/O»		CYCLE TIME	PACKAGE TYPE†		
		RAM	ROM	EPROM	PROG	SER	PAR	(ns)	DIP	PLCC	CER
TMS32010‡	NMOS	144	1.5K	–	4K	–	8x16	200	40	–	–
TMS320C10‡	CMOS	144	1.5K	–	4K	–	8x16	200	40	44	–
TMS320C10-14	CMOS	144	1.5K	–	4K	–	8x16	280	40	44	–
TMS320C10-25	CMOS	144	1.5K	–	4K	–	8x16	160	40	44	–
TMS320C14§	CMOS	256	4K	–	4K	1	7x16¶	160	–	68	–
TMS320E14§	CMOS	256	–	4K	4K	1	7x16¶	160	–	–	68
TMS320C15§	CMOS	256	4K	–	4K	–	8x16	200	40	44	–
TMS320C15-25	CMOS	256	4K	–	4K	–	8x16	160	40	44	–
TMS320E15§	CMOS	256	–	4K	4K	–	8x16	200	40	–	44
TMS320E15-25	CMOS	256	–	4K	4K	–	8x16	160	40	–	44
TMS320C17	CMOS	256	4K	–	–	2	6x16♪	200	40	44	–
TMS320E17	CMOS	256	–	4K	–	2	6x16♪	200	40	–	44

»SER = serial; PAR = parallel.
†DIP = dual in-line pin; PLCC = plastic leaded chip carrier; CER = surface mount ceramic leaded chip carrier (CER-QUAD).
‡Military version available.
§Military versions planned; contact nearest TI Field Sales Office for availability.
¶On-chip 16-bit I/O, four capture inputs, and six compare outputs are available.
♪On-chip 16-bit coprocessor interface is optional by pin selection.

The first generation of the TMS320 family includes both NMOS and CMOS products. The TMS32010 microprocessor is the only NMOS device. The other members are processed in CMOS technology: TMS320C10, TMS320C10-14, TMS320C10-25, TMS320C14/E14, TMS320C15/E15, TMS320C15-25/E15-25, and TMS320C17/E17.

The **TMS32010**, the first TMS320 family member, is a microprocessor capable of achieving a 16 x 16-bit multiply in a single 200-ns cycle. On-chip data memory of 144 words is available. Up to 4K words of off-chip program memory can be executed at full speed. The TMS32010 is also available in a microcomputer version, with 1.5K words of on-chip program ROM and up to 2.5K words of off-chip program memory for a total of 4K words. This ROM-code version can also operate entirely from off-chip ROM for ease of prototyping, code update, and field upgradeability.

The **TMS320C10** has a 200-ns instruction cycle time and is object-code and pin-for-pin compatible with the TMS32010. The TMS320C10 is processed in CMOS technology, achieving a power dissipation less than one-sixth that of the NMOS device. Because of its low-power dissipation (165 mW), the TMS320C10 is ideal for power-sensitive applications such as digital telephony and portable consumer products. A masked ROM option is available for the TMS320C10.

The **TMS320C10-14**, a 14-MHz version of the TMS320C10, provides a low-cost alternative for DSP applications not requiring the maximum operating frequency of the TMS320C10. The device can execute 3.5 million instructions per second and has a 280-ns instruction cycle time.

The **TMS320C10-25**, a 25-MHz version of the TMS320C10, has a 160-ns instruction cycle time. Its lower power and higher speed make it well suited for high-performance DSP applications.

The **TMS320C14** and **TMS320E14** are microcontrollers with an instruction cycle time of less than 160-ns, 256 words of on-chip RAM, and 4K words of on-chip program ROM (TMS320C14) or EPROM (TMS320E14). The TMS320C14/E14 feature an event manager with four capture inputs and six compare outputs, a bit-selectable I/O port, a serial port with programmable protocols and timer, a watchdog timer, and two general-purpose timers. These devices are object-code compatible with the TMS32010 and processed in CMOS technology.

The **TMS320C15** and **TMS320E15** are fully object-code and pin-for-pin compatible with the TMS32010. Each offers an expanded on-chip RAM of 256 words and an on-chip program ROM (TMS320C15) or EPROM (TMS320E15) of 4K words. The devices are processed in CMOS technology. Both are also available in the 160-ns versions, the **TMS320C15-25** and **TMS320E15-25**.

The **TMS320C17** and **TMS320E17** are dedicated microcomputers. Each offers 256 words of on-chip RAM and 4K words of on-chip program ROM (TMS320C17) or EPROM (TMS320E17). The TMS320C17/E17 features a dual-channel serial interface, on-chip companding hardware (µ-law/A-law), a serial port timer, and a latched 16-bit coprocessor port for direct microprocessor I/O interface. The devices are object-code compatible with the TMS32010 and processed in CMOS technology.

1.2 Key Features

Some of the key features of the TMS320C1x devices are listed below and on the following page. Specific devices for a particular feature are enclosed in parentheses.

- Instruction cycle timing:
  - 160 ns (TMS320C10-25/C14/E14/C15-25/E15-25)
  - 200 ns (TMS32010/C10/C15/E15/C17/E17)
  - 280 ns (TMS320C10-14)
- 144-/256-word on-chip data RAM
- 1.5K-/4K-word on-chip program ROM
- 4K-word on-chip program EPROM (TMS320E14/E15/E15-25/E17)
- EPROM code protection for copyright security
- 4K-word total external memory at full speed (TMS32010/C10/C10-14/C10-25/C14/E14/C15/C15-25/E15/E15-25)
- 16-bit bidirectional data bus at 50-Mbps transfer rate
- 32-bit ALU/accumulator
- 16- x 16-bit parallel multiplier with a 32-bit product
- 0- to 16-bit barrel shifter
- On-chip clock generator
- Eight input and eight output channels
- Dual-channel serial port with timer (TMS320C17/E17)
- Direct interface to combo-codecs (TMS320C17/E17)
- On-chip µ-law/A-law companding hardware (TMS320C17/E17)
- 16-bit coprocessor interface (TMS320C17/E17)
- 16-pin bit-selectable I/O ports (TMS320C14/E14)
- Serial port with programmable protocols (TMS320C14/E14)
- Event manager with capture inputs and compare outputs (TMS320C14/E14)

- Four independent timers (TMS320C14/E14)
  - General-purpose (2)
  - Serial port
  - Watchdog
- 15 external/internal interrupts (TMS320C14/E14)
- Single 5-V supply
- Device packaging:
  - 40-pin DIP (TMS32010/C10/C10-14/C10-25/C15/C15-25/E15/E15-25/C17/E17)
  - 44-lead PLCC (TMS320C10/C10-14/C10-25/C15/C15-25/C17)
  - 68-lead PLCC (TMS320C14)
  - 44-lead CER-QUAD (TMS320E15/E15-25/E17)
  - 68-lead CER-QUAD (TMS320E14)
- Technology:
  - NMOS (TMS32010)
  - CMOS (TMS320C10/C10-14/C10-25/C14/E14/C15/C15-25/E15/E15-25/C17/E17)
- Commercial and military versions available.

1.3 Typical Applications

The TMS320 family's unique versatility and realtime performance offer flexible design approaches in a variety of applications. In addition, TMS320 devices can simultaneously provide the multiple functions often required in those complex applications. Table 1-2 lists typical TMS320 family applications.

Table 1-2. Typical Applications of the TMS320 Family

GENERAL-PURPOSE DSP	GRAPHICS/IMAGING	INSTRUMENTATION
Digital Filtering Convolution Correlation Hilbert Transforms Fast Fourier Transforms Adaptive Filtering Windowing Waveform Generation	3-D Rotation Robot Vision Image Transmission/ Compression Pattern Recognition Image Enhancement Homomorphic Processing Workstations Animation/Digital Map	Spectrum Analysis Function Generation Pattern Matching Seismic Processing Transient Analysis Digital Filtering Phase-Locked Loops
VOICE/SPEECH	**CONTROL**	**MILITARY**
Voice Mail Speech Vocoding Speech Recognition Speaker Verification Speech Enhancement Speech Synthesis Text-to-Speech	Disk Control Servo Control Robot Control Laser Printer Control Engine Control Motor Control	Secure Communications Radar Processing Sonar Processing Image Processing Navigation Missile Guidance Radio Frequency Modems
TELECOMMUNICATIONS		**AUTOMOTIVE**
Echo Cancellation ADPCM Transcoders Digital PBXs Line Repeaters Channel Multiplexing 1200 to 19200-bps Modems Adaptive Equalizers DTMF Encoding/Decoding Data Encryption	FAX Cellular Telephones Speaker Phones Digital Speech Interpolation (DSI) X.25 Packet Switching Video Conferencing Spread Spectrum Communications	Engine Control Vibration Analysis Antiskid Brakes Adaptive Ride Control Global Positioning Navigation Voice Commands Digital Radio Cellular Telephones
CONSUMER	**INDUSTRIAL**	**MEDICAL**
Radar Detectors Power Tools Digital Audio/TV Music Synthesizer Educational Toys	Robotics Numeric Control Security Access Power Line Monitors	Hearing Aids Patient Monitoring Ultrasound Equipment Diagnostic Tools Prosthetics Fetal Monitors

1.4 How To Use This Manual

The purpose of this user's guide is to serve as a reference book for the first-generation TMS320 digital signal processors. Sections 2 through 6 provide specific information on architecture and operation of these devices. Appendix A furnishes electrical specifications and mechanical data information.

The following table lists each section and briefly describes the section contents.

Section 2. Pinouts and Signal Descriptions. Drawings of the DIP and PLCC packages for TMS320C1x devices. Functional listings of the signals, their pin locations, and descriptions.

Section 3. Architecture. TMS320C1x design description, hardware components, and device operation. Functional block diagrams and internal hardware summary table.

Section 4. Assembly Language Instructions. Addressing modes and format descriptions. Instruction set summary listed according to function. Alphabetized individual instruction descriptions with examples.

Section 5. Software Applications. Software application examples for the use of various TMS320C1x instruction set features.

Section 6. Hardware Applications. Hardware design techniques and application examples for interfacing to codecs, external memory, or common 4-/8-/16-/32-bit microcomputers and microprocessors.

Seven appendices are included to provide additional information.

Appendix A. First-Generation TMS320 Data Sheets. Electrical specifications, timing, and mechanical data for all TMS320C1x devices.

Appendix B. SMJ32010/C10 Data Sheets. Electrical specifications, timing, and mechanical data for these military devices.

Appendix C. ROM Codes. Discussion of ROM codes (mask options) and the procedure for implementation.

Appendix D. Quality and Reliability. Discussion of Texas Instruments quality and reliability criteria for evaluating performance.

Appendix E. Development Support/Part Order Information. Listings of the hardware and software available to support the TMS320C1x devices.

Appendix F. Memories, Analog Converters, Sockets, and Crystals. Listings of the TI memories, analog conversion devices, and sockets available to support the TMS320C1x devices in DSP applications. Crystal specifications and vendors.

Appendix G. Programming the TMS320E15/E17 EPROM Cell. Procedure for programming and verifying the EPROM cell using the 28-pin TMS27C64.

1.5 References

The following reference list contains useful information regarding functions, operations, and applications of digital signal processing. These books also provide other references to many useful technical papers. The reference list is organized into categories of general DSP, speech, image processing, and digital control theory; if known, each category is alphabetized by the author's last name.

General Digital Signal Processing:

Antoniou, Andreas, *Digital Filters: Analysis and Design*. New York, NY: McGraw-Hill Company, Inc., 1979.

Brigham, E. Oran, *The Fast Fourier Transform*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1974.

Burrus, C.S. and Parks, T.W., *DFT/FFT and Convolution Algorithms*. New York, NY: John Wiley and Sons, Inc., 1984.

Digital Signal Processing Applications with the TMS320 Family, Texas Instruments, 1986; Prentice-Hall, Inc., 1987.

Gold, Bernard and Rader, C.M., *Digital Processing of Signals*. New York, NY: McGraw-Hill Company, Inc., 1969.

Hamming, R.W., *Digital Filters*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1977.

IEEE ASSP DSP Committee (Editor), *Programs for Digital Signal Processing*. New York, NY: IEEE Press, 1979.

Jackson, Leland B., *Digital Filters and Signal Processing*. Hingham, MA: Kluwer Academic Publishers, 1986.

Jones, D.L. and Parks, T.W., *A Digital Signal Processing Laboratory Using the TMS32010*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1987.

Lim,Jae and Oppenheim, Alan V., *Advanced Topics in Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1988.

Morris, L. Robert, *Digital Signal Processing Software*. Ottawa, Canada: Carleton University, 1983.

Oppenheim, Alan V. (Editor), *Applications of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1978.

Oppenheim, Alan V. and Schafer, R.W., *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1975.

Oppenheim, Alan V. and Willsky, A.N. with Young, I.T., *Signals and Systems*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1983.

Parks, T.W. and Burrus, C.S., *Digital Filter Design*. New York, NY: John Wiley and Sons, Inc., 1987.

Rabiner, Lawrence R. and Gold, Bernard, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1975.

Treichler, J.R., Johnson, Jr., C.R., and Larimore, M.G., *Theory and Design of Adaptive Filters*. New York, NY: John Wiley and Sons, Inc., 1987.

Speech:

Gray, A.H. and Markel, J.D., *Linear Prediction of Speech*. New York, NY: Springer-Verlag, 1976.

Jayant, N.S. and Noll, Peter, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1984.

Papamichalis, Panos, *Practical Approaches to Speech Coding*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1987.

Rabiner, Lawrence R. and Schafer, R.W., *Digital Processing of Speech Signals*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1978.

Image Processing:

Andrews, H.C. and Hunt, B.R., *Digital Image Restoration*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1977.

Gonzales, Rafael C. and Wintz, Paul, *Digital Image Processing*. Reading, MA: Addison-Wesley Publishing Company, Inc., 1977.

Pratt, William K., *Digital Image Processing*. New York, NY: John Wiley and Sons, 1978.

Digital Control Theory:

Astrom, K. and Wittenmark, B., *Computer Controlled Systems*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1984.

Iserman, R., *Digital Control Systems*. New York, NY: Springer-Verlag, 1981.

Jacquot, R., *Modern Digital Control Systems*. New York, NY: Marcel Dekker, Inc., 1981.

Katz, P., *Digital Control Using Microprocessors*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1981.

Kuo, B.C., *Digital Control Systems*. New York, NY: Holt, Reinholt and Winston, Inc., 1980.

Moroney, P., *Issues in the Implementation of Digital Feedback Compensators*. Cambridge, MA: The MIT Press, 1983.

Phillips, C. and Nagle, H., *Digital Control System Analysis and Design*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1984.

Section 2

Pinouts and Signal Descriptions

The TMS320C1x (first-generation TMS320) digital signal processors, except TMS320C14 and TMS320E14, are available in a 40-pin dual-in-line (DIP) package. The TMS320C14 is only available in the 68-pin plastic-leaded chip carrier (PLCC) and the TMS320E14 is only available in a 68-pin CER-QUAD package. The TMS320C10 and TMS320C15/C17 are also packaged in a 44-pin plastic-leaded chip carrier (PLCC). The TMS320E15 and TMS320E17 are available in 44-pin CER-QUAD packages, too.

This section provides the pinouts and signal definitions in the following subsections:

- TMS320C1x Pinouts (Section 2.1 on page 2-2)
- TMS32010/C10/C15/E15 Signal Descriptions (Section 2.2 on page 2-3)
- TMS320C17/E17 Signal Descriptions (Section 2.3 on page 2-5)

Electrical specifications and mechanical data are given in Appendix A which contains the First-Generation TMS320 and the TMS320C14/E14 data sheets. For pinouts used in programming the TMS320E14/E15/E17 EPROMs, refer to Appendix G.

2.1 TMS320C1x Pinouts

Figure 2-1 shows pinouts of the DIP packages for the TMS320C1x devices and the PLCC packages for the TMS320C10/C15/C17. For pinouts of the TMS320C14/E14, see Appendix A or refer to the *TMS320C14/TMS320E14 User's Guide* (literature number SPRU032).

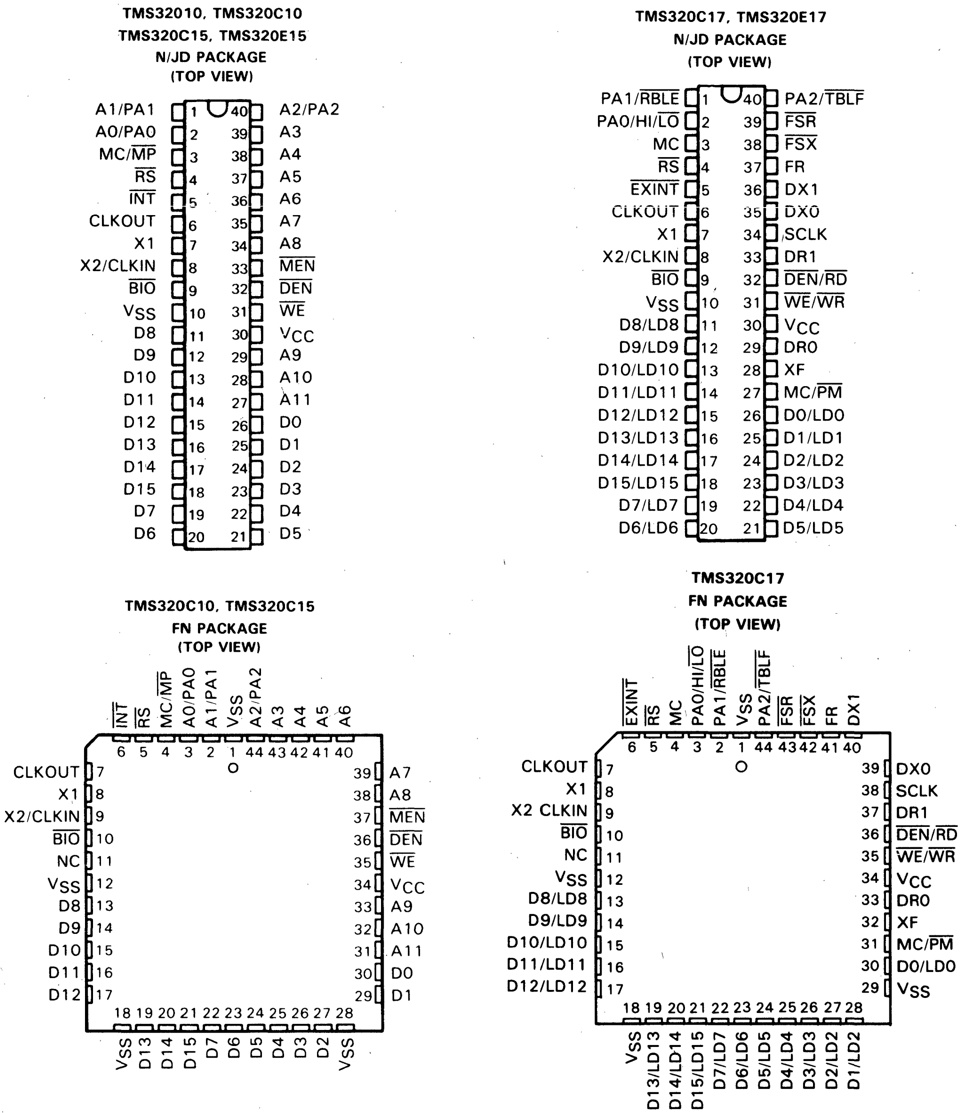


Figure 2-1. TMS320C1x Pin Assignments

2.2 TMS32010/C10/C15/E15 Signal Descriptions

The signal descriptions for the TMS32010/C10 and TMS320C15/E15 devices are provided in this section. Table 2-1 lists each signal, its pin location (DIP/PLCC), function, and operating mode(s), i.e., input, output, or high-impedance state as indicated by I, O, or Z. The signals in Table 2-1 are grouped according to function and alphabetized within that grouping.

Table 2-1. TMS32010/C10/C15/E15 Signal Descriptions

SIGNAL	PIN (DIP/PLCC)	I/O/Z†	DESCRIPTION
			ADDRESS/DATA BUSES
A11 MSB A10 A9 A8 A7 A6 A5 A4 A3 A2/PA2 A1/PA1 A0/PA0	27/31 28/32 29/33 34/38 35/39 36/40 37/41 38/42 39/43 40/44 1/2 2/3	O	Program memory address bus A11 (MSB) through A0 (LSB) and port addresses PA2 (MSB) through PA0 (LSB). Addresses A11 through A0 are always active and never go to high impedance. During execution of the IN and OUT instructions, pins A2 through A0 carry the port addresses. (Address pins A11 through A3 are always driven low on IN and OUT instruction)
D15 MSB D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 LSB	18/21 17/20 16/19 15/17 14/16 13/15 12/14 11/13 19/22 20/23 21/24 22/25 23/26 24/27 25/29 26/30	I/O/Z	Parallel data bus D15 (MSB) through D0 (LSB). The data bus is always in the high-impedance state except when $\overline{\text{WE}}$ is active (low).
			INTERRUPT AND MISCELLANEOUS SIGNALS
$\overline{\text{BIO}}$	9/10	I	External polling input. Polled by BIOZ instruction. If low, the device branches to the address specified by the instruction.
$\overline{\text{DEN}}$	32/36	O	Data enable for device input data. When active low, $\overline{\text{DEN}}$ indicates that the device will accept data from the data bus. $\overline{\text{DEN}}$ is only active during the first cycle of the IN instruction. When $\overline{\text{DEN}}$ is active, $\overline{\text{MEN}}$ and $\overline{\text{WE}}$ will always be inactive (high).

† Input/Output/High-impedance state

Table 2-1. TMS32010/C10/C15/E15 Signal Descriptions (Concluded)

SIGNAL	PIN (DIP/PLCC)	I/O/Z†	DESCRIPTION
$\overline{\text{INT}}$	5/6	I	External interrupt input. The interrupt signal is generated by applying a negative-going edge to the $\overline{\text{INT}}$ pin. The edge is used to latch the interrupt flag register (INTF) until an interrupt is granted by the device. An active low level will also be sensed.
MC/$\overline{\text{MP}}$	3/4	I	Memory mode select pin. High selects the microcomputer mode, in which 1.5K words (4K on the TMS320C15/E15) of on-chip program memory are available. This mode also allows an additional 2.5K words of program memory to reside off-chip on the TMS32010/C10. A low on MC/$\overline{\text{MP}}$ pin enables the microprocessor mode. In this mode, the entire memory space is external, i.e., addresses 0 through 4095.
$\overline{\text{MEN}}$	33/37	O	Memory enable. $\overline{\text{MEN}}$ will be active low on every machine cycle except when $\overline{\text{WE}}$ and $\overline{\text{DEN}}$ are active. $\overline{\text{MEN}}$ is a control signal generated by the device to enable instruction fetches from program memory. $\overline{\text{MEN}}$ will be active on instructions fetched from both internal and external memory.
$\overline{\text{RS}}$	4/5	I	Reset input for initializing the device. When held at an active low for a minimum of five clock cycles, $\overline{\text{DEN}}$, $\overline{\text{WE}}$, and $\overline{\text{MEN}}$ are forced high; and, the data bus (D15 through D0) is not driven. The program counter (PC) and the address bus (A11 through A0) are then synchronously cleared after the next complete clock cycle from the falling edge of $\overline{\text{RS}}$. Reset also disables the interrupt, clears the interrupt flag register, and leaves the overflow mode register unchanged. The device can be held in the reset state indefinitely.
$\overline{\text{WE}}$	31/35	O	Write enable for device output data. When active low, $\overline{\text{WE}}$ indicates that data will be output from the device on the data bus. $\overline{\text{WE}}$ is only active during the first cycle of the OUT instruction and the second cycle of the TBLW instruction. When $\overline{\text{WE}}$ is active, $\overline{\text{MEN}}$ and $\overline{\text{DEN}}$ will always be inactive (high).
SUPPLY/OSCILLATOR SIGNALS			
CLKOUT	6/7	O	System clock output (one-fourth crystal/CLKIN frequency). Duty cycle is fifty percent.
V_{CC}	30/34	I	5-V supply pin.
V_{SS}	10/12	I	Ground pin.
X1	7/8	O	Crystal output pin for internal oscillator. If the internal oscillator is not used, this pin should be left unconnected.
X2/CLKIN	8/9	I	Input pin to the internal oscillator (X2) from the crystal. Alternatively, an input pin for an external oscillator (CLKIN).

† Input/Output/High-impedance state

2.3 TMS320C17/E17 Signal Descriptions

Table 2-2 lists each signal provided on the TMS320C17/E17, its pin location, function, and operating mode(s), i.e., input, output, or high-impedance state as indicated by I, O, or Z. The signals in Table 2-2 are grouped according to function and alphabetized within that grouping. Note that the first signal and the signal following the slash are both used on the TMS320C17/E17.

Table 2-2. TMS320C17/E17 Signal Descriptions

SIGNAL	PIN (DIP/PLCC)	I/O/Z†	DESCRIPTION
			BIDIRECTIONAL DATA BUS
D15/LD15	18/21	I/O/Z	During the microcomputer mode, this represents a 16-bit parallel data bus (D15 through D0). The data bus is always in the high-impedance state, except when $\overline{\text{WE}}$ is active (low) or when an IN instruction is being executed from either port 0 or port 1. During the coprocessor mode, the 16-bit data lines (LD15 through LD0) is used for a coprocessor latch. The data bus is always held in a high-impedance state, except when $\overline{\text{RD}}$ is active (low).
D14/LD14	17/20		
D13/LD13	16/19		
D12/LD12	15/17		
D11/LD11	14/16		
D10/LD10	13/15		
D9/LD9	12/14		
D8/LD8	11/13		
D7/LD7	19/22		
D6/LD6	20/23		
D5/LD5	21/24		
D4/LD4	22/25		
D3/LD3	23/26		
D2/LD2	24/27		
D1/LD1	25/28		
D0/LD0	26/30		
			PORT ADDRESS BUS
PA2/$\overline{\text{TBLF}}$	40/44	O	I/O port address output/transmit buffer latch full flag.
PA1/$\overline{\text{RBLE}}$	1/2	O	I/O port address output/receive buffer latch empty flag.
PA0/HI/$\overline{\text{LO}}$	2/3	I/O/Z	I/O port address output/latch byte select pin. During the microcomputer mode, these pins carry the port address when using the IN and OUT instructions. When using other instruction cycles, these pins carry the three LSBs of the program counter. During the coprocessor mode, these pins signal the status of the receive and the transmit buffer latches.
			INTERRUPT AND MISCELLANEOUS SIGNALS
$\overline{\text{BIO}}$	9/10	I	External polling input. Polled by BIOZ instruction. If low, the device branches to the address specified by the instruction. When in the coprocessor mode, the $\overline{\text{BIO}}$ line is reserved for coprocessor interface and cannot be driven externally.

† Input/Output/High-impedance state

Table 2-2. TMS320C17/E17 Signal Descriptions (Continued)

SIGNAL	PIN (DIP/PLCC)	I/O/Z†	DESCRIPTION
$\overline{\text{DEN}}/\overline{\text{RD}}$	32/36	I/O/Z	Data enable for device input data/external read for the output latch. When active low, $\overline{\text{DEN}}$ indicates that the device will accept data from the data bus. $\overline{\text{DEN}}$ is only active during the first cycle of the IN instruction. $\overline{\text{WE}}$ will always be inactive (high) when $\overline{\text{DEN}}$ is active. In the coprocessor mode, the external processor reads from the coprocessor latch by driving the $\overline{\text{RD}}$ line active (low), thus enabling the output latch to drive the latched data. When the data has been read, the external device will bring the $\overline{\text{RD}}$ line high.
$\overline{\text{EXINT}}$	5/6	I	External interrupt input. The interrupt signal is generated by applying a logic low level to the $\overline{\text{EXINT}}$ pin. The edge is used to latch the system control register flag bit (CR0) until an interrupt is granted by the device. When in the coprocessor mode, the $\overline{\text{EXINT}}$ line is reserved for coprocessor interface and cannot be driven externally.
MC	3/4	I	Microcomputer mode select pin. The MC pin must be connected to the same state as the MC/$\overline{\text{PM}}$ pin. When these pins are low, the coprocessor port is enabled. When these pins are high, the microcomputer mode is enabled.
MC/$\overline{\text{PM}}$	27/31	I	Microcomputer or peripheral/coprocessor mode select pin. This pin must be connected to the same state as the MC pin. When these pins are low, the coprocessor port is enabled. When these pins are high, the microcomputer mode is enabled.
$\overline{\text{RS}}$	4/5	I	Reset input for initializing the device. When an active low is placed on the $\overline{\text{RS}}$ pin for a minimum of five clock cycles, both $\overline{\text{DEN}}$ and $\overline{\text{WE}}$ are forced high, and the data bus (D15 through D0) goes to a high-impedance state. The serial port clock and transmit outputs also go to the high-impedance state. The program counter (PC) and the port address bus (PA2 through PA0) are then synchronously cleared after the next complete clock cycle from the falling edge of $\overline{\text{RS}}$.
$\overline{\text{WE}}/\overline{\text{WR}}$	31/35	I/O	Write enable for device output data/external write enable for the input latch. When active low, $\overline{\text{WE}}$ indicates that data will be output from the device on the data bus. $\overline{\text{WE}}$ is only active during the first cycle of the OUT instruction and the second cycle of the TBLW instruction. $\overline{\text{DEN}}$ will always be inactive (high) when $\overline{\text{WE}}$ is active. In the coprocessor mode, the external processor lowers the $\overline{\text{WR}}$ line and places data on the bus. It next raises the $\overline{\text{WR}}$ line to clock the data into the on-chip latch.

† Input/Output/High-impedance state

Table 2-2. TMS320C17/E17 Signal Descriptions (Concluded)

SIGNAL	PIN (DIP/PLCC)	I/O/Z†	DESCRIPTION
XF	28/32	O	External logic output flag. Programmable via system control register bit 10 (CR10). This pin is the direct output of the CR10 latch.
SUPPLY/OSCILLATOR SIGNALS			
CLKOUT	6/7	O	System clock output (one-fourth crystal/CLKIN frequency).
V_{CC}	30/34	I	5-V supply pin.
V_{SS}	10/12	I	Ground pin.
X1	7/8	O	Crystal output pin for internal oscillator. If the internal oscillator is not used, this pin should be left unconnected.
X2/CLKIN	8/9	I	Input pin to the internal oscillator (X2) from the crystal. Alternatively, an input pin for an external oscillator (CLKIN).
SERIAL PORT SIGNALS			
DR1 DR0	33/37 29/33	I	Serial-port receive-channel inputs. Serial data is received in the receive registers via these pins.
DX1 DX0	36/40 35/39	O/Z	Serial-port transmit-channel outputs. Serial data is transmitted from the transmit registers on these pins. These outputs are in the high-impedance state when not transmitting.
FR	37/41	O	Internal serial-port framing output. If internal framing is enabled, serial-port transmit and receive operations occur simultaneously on an active (high) FR framing pulse. Both short and long FR pulses are selectable to provide fixed and variable data-rate framing pulses for combo-codec interface. The FR frequency is derived from the serial-port clock (SCLK) and system control register bits CR23-CR16.
$\overline{\text{FSR}}$	39/43	I	External serial-port receive-framing input. If external framing is enabled via the system control register, data is received via the receive pins (DR1 and DR0) on the active (low) $\overline{\text{FSR}}$ input. The falling edge of $\overline{\text{FSR}}$ initiates the receive process, and the rising edge sets the flag bit (CR1) in the system control register, causing an interrupt to occur if enabled.
$\overline{\text{FSX}}$	38/42	I	External serial-port transmit-framing input. If external framing is enabled, data is transmitted on the transmit pins (DX1,DX0) on the active (low) $\overline{\text{FSX}}$ input. The falling edge of $\overline{\text{FSX}}$ initiates the transmit process, and the rising edge sets the flag bit (CR2) in the system control register, causing an interrupt to occur if enabled.
SCLK	34/38	I/O/Z	Serial-port clock. Master clock for transmitting and receiving serial-port data. Configurable as an input or output. SCLK must always be present for serial-port operation. As an input, SCLK is the external clock that controls data transfers with the serial port. As an output, SCLK provides the serial clock for data transfers and framing-pulse synchronization. Its frequency is derived from the TMS320C17/E17 system clock, X2/CLKIN, and system control register bits CR27-CR24. Reset ($\overline{\text{RS}}$) forces SCLK to the high-impedance state.

† Input/Output/High-impedance state

Section 3

Architecture

The modified Harvard architecture of the TMS320C1x (first-generation TMS320) microprocessors increases throughput by allowing program fetch to overlap data operations. The hardware-intensive design of these devices provides performance previously unavailable on a single chip. Hardware is used to implement functions that other processors typically perform in software. For example, a TMS320C1x device contains a hardware multiplier to perform the multiplication process during one instruction cycle. Flexibility is further enhanced by the comprehensive instruction set which supports either general-purpose or digital signal processing applications.

Major topics discussed in this section are listed below and on the next page.

- Architectural Overview (Section 3.1 on page 3-3)
- Functional Block Diagrams (Section 3.2 on page 3-5)
- Internal Hardware Summary (Section 3.3 on page 3-7)
- Memory Organization (Section 3.4 on page 3-10)
  Data and program memory
  Data movement
  Memory maps
  Auxiliary registers
  Microcomputer/microprocessor modes
  Addressing modes
- Central Arithmetic Logic Unit (CALU) (Section 3.5 on page 3-17)
  Shifters, ALU, and accumulator
  Multiplier, T and P registers
- System Control (Section 3.6 on page 3-22)
  Program counter and stack
  Reset
  Status register
- I/O Functions (Section 3.7 on page 3-27)
  Input/output operation
  Table read/table write operation
  General-purpose I/O pins ($\overline{\text{BIO}}$ and XF)

- Interrupts (Section 3.8 on page 3-32)
- Serial Port (Section 3.9 on page 3-36)
  - Receive and transmit registers
  - Timing and framing control
- Companding Hardware (Section 3.10 on page 3-43)
  - Encoder and decoder
- Coprocessor Port (Section 3.11 on page 3-46)
- System Control Register (Section 3.12 on page 3-51)

3.1 Architectural Overview

The TMS320 family utilizes a modified Harvard architecture for speed and flexibility. In a strict Harvard architecture, program and data memory lie in two separate spaces, permitting a full overlap of instruction fetch and execution. The TMS320 family's modification of the Harvard architecture allows transfers between program and data spaces, thereby increasing the flexibility of the device. This modification permits coefficients stored in program memory to be read into RAM, eliminating the need for a separate coefficient ROM. It also makes available immediate instructions and subroutines based on computed values.

The TMS320C1x devices contain a 32-bit ALU and accumulator for supporting double-precision, two's-complement arithmetic. The ALU is a general-purpose arithmetic unit; operations are done by using the 16-bit words taken from data RAM, the 16-bit words derived from immediate instructions, or the 32-bit result taken from the product register of the multiplier. In addition to the usual arithmetic instructions, the ALU can perform Boolean operations, providing the bit manipulation ability required of a high-speed controller. The accumulator stores the output from the ALU and is often an input to the ALU. The accumulator is 32 bits in length and is divided into a high-order word (bits 31 through 16) and a low-order word (bits 15 through 0). Instructions are provided for storing the high- and low-order accumulator words in memory.

The multiplier performs a 16 x 16-bit two's-complement multiplication with a 32-bit result in a single instruction cycle. The multiplier consists of three elements: the T Register, P Register, and multiplier array. The 16-bit T Register temporarily stores the multiplicand; the P Register stores the 32-bit product. Multiplier values either come from the data memory or are derived immediately from the MPYK (multiply immediate) instruction word. The fast on-chip multiplier allows the device to efficiently perform fundamental DSP operations such as convolution, correlation, and filtering.

Two shifters are available for manipulating data. The ALU barrel shifter performs a left-shift of 0 to 16 places on data memory words loaded into the ALU. This shifter extends the high-order bit of the data word and zero-fills the low-order bits for two's-complement arithmetic. The accumulator parallel shifter performs a left-shift of 0, 1, or 4 places on the entire accumulator and stores the resulting high-order accumulator bits into data RAM. Both shifters are useful for scaling and bit extraction.

The TMS320C1x devices have 144/256 words of on-chip data RAM and 1.5K/4K words of on-chip program ROM/EPROM to support program development. The EPROM cell utilizes standard PROM programmers and programs identically to a 64K CMOS EPROM (TMS27C64). The TMS320C1x devices are capable of executing programs from up to 4K words of memory at full speed for those applications requiring external program memory space. This allows for external RAM-based systems to provide multiple functionality. The TMS320C17/E17 does not provide memory expansion capability.

The TMS32010/C10 and TMS320C15/E15 devices offer two modes of operation defined by the state of the MC/$\overline{\text{MP}}$ pin: the microcomputer mode (high level) or the microprocessor mode (low level). In the microcomputer mode, on-chip ROM is mapped into the memory space with up to 4K words of memory available. In the microprocessor mode, all 4K words of memory are external.

The TMS320C1x devices contain a four-level hardware stack for saving the contents of the program counter during interrupts and subroutine calls. Instructions are available for saving the device's complete context. PUSH and POP instructions permit a level of nesting restricted only by the amount of available RAM. The interrupts used in these devices are maskable.

The 16-bit parallel data bus can be utilized to perform I/O functions in two cycles. The I/O ports are addressed by the three LSBs on the address lines. In addition, a polling input for bit test and branch operations ($\overline{\text{BIO}}$) and an interrupt pin ($\overline{\text{INT}}$) have been incorporated for increased system flexibility. Two of the I/O ports on the TMS320C17/E17 are dedicated to the serial port and companding hardware. I/O port 0 is dedicated to control register 0, which controls the serial port, interrupts, and companding hardware. I/O port 1 accesses control register 1, as well as both serial port channels, and the companding hardware. The six remaining I/O ports are available for external parallel interfaces. On the TMS320C17/E17, port 5 may be used for coprocessor interface. When port 5 is used as the coprocessor interface, ports 2, 3, 4, 6, and 7 are not accessible.

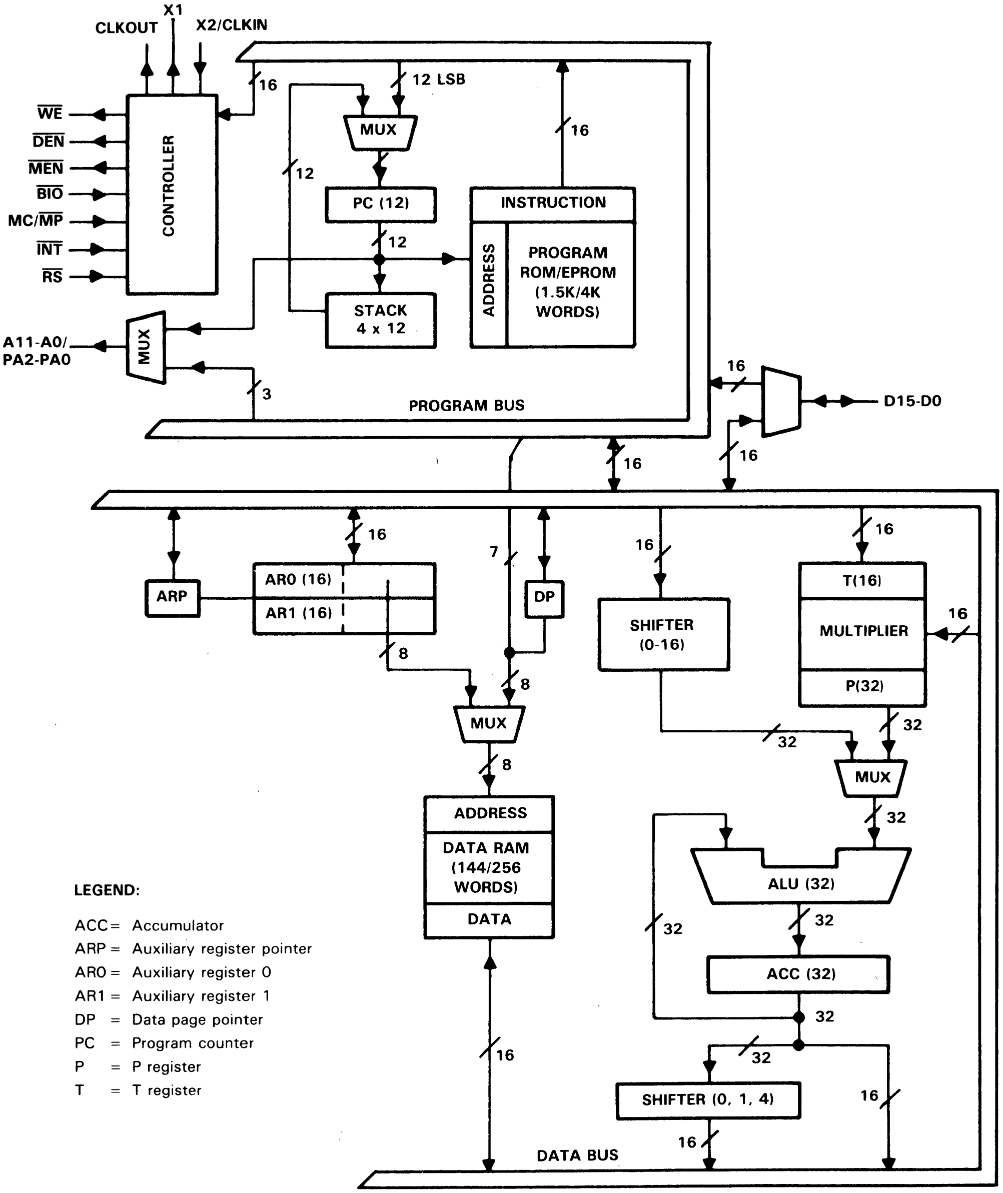
The TMS320C17/E17 offers a dual-channel serial port capable of full-duplex serial communication and direct interface to combo-codecs. Receive and transmit registers that operate with 8-bit data samples are I/O-mapped. Either internal or external framing signals for serial data transfers are selected through the system control register. The serial port clock provides the bit timing for transfers with the serial port, and may be either an input or output. A framing pulse signal provides framing pulses for combo-codec circuits, an 8-kHz sample clock for voice-band systems, or a timer for control applications.

On-chip hardware can compand (COMpress/exPAND) data in either μ-law (U.S. and Japan) or A-law (European) format. The companding logic operation is configured via the system control register. Data may be companded in either a serial mode for operation on serial port data (converting between linear and logarithmic PCM) or a parallel mode for computation inside the device. The TMS320C17/E17 allows the hardware companding logic to operate with either sign-magnitude or two's-complement numbers.

The coprocessor port on the TMS320C17/E17 provides a direct connection to most 4/8-bit microcomputers and 16/32-bit microprocessors. In the coprocessor mode, the 16-bit parallel port is reconfigured to operate as a 16-bit latched bus interface. Data widths of either 8 or 16 bits may be selected for the coprocessor port, accessed through I/O port 5 using IN and OUT instructions. The coprocessor interface allows the device to act as a peripheral (slave) microcomputer to a microprocessor, or as a master to a peripheral microcomputer. In the microcomputer mode, the 16 data lines are used for the six parallel 16-bit I/O ports.

3.2 Functional Block Diagrams

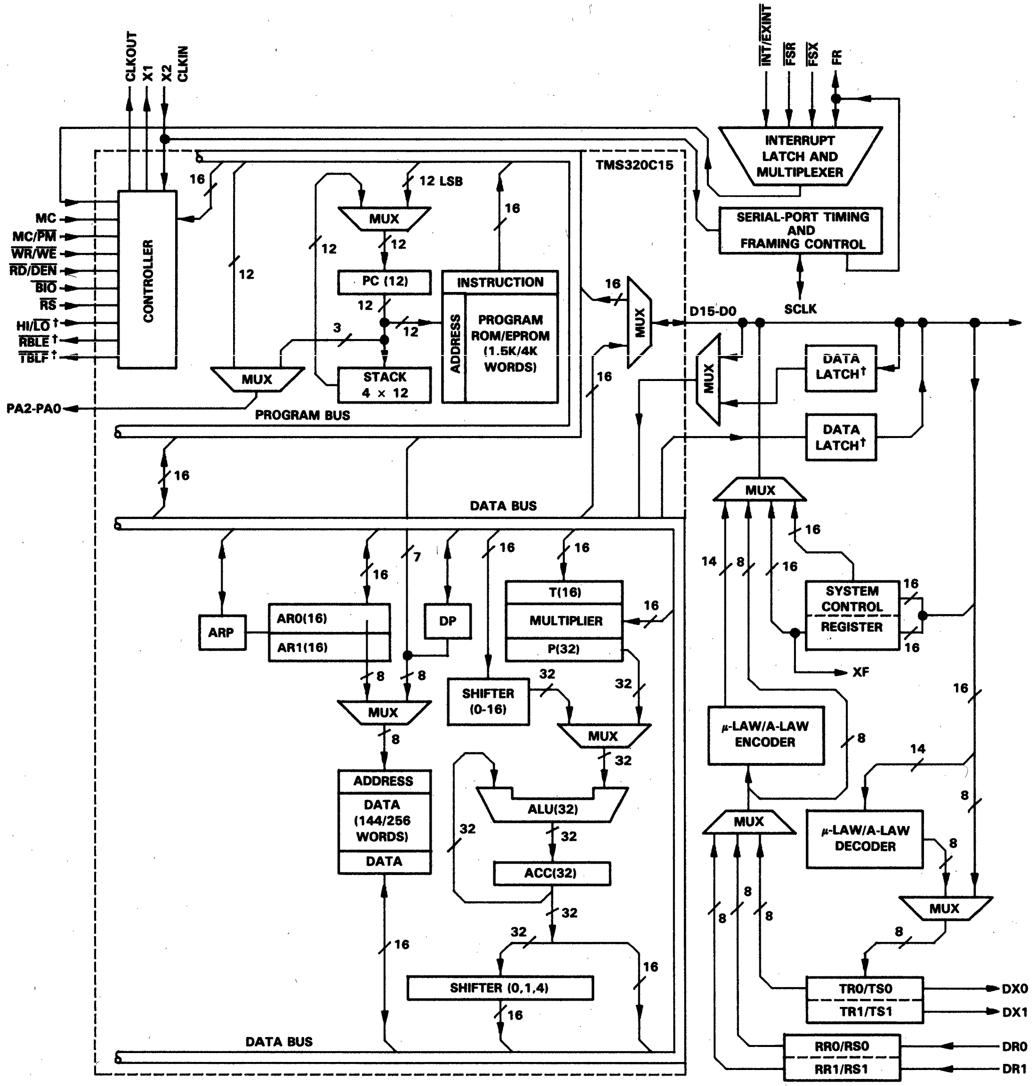
The functional block diagrams shown in this section outline the principal blocks and data paths within the TMS320C1x processors. Further details of functional blocks are given in the succeeding sections. The two block diagrams also show all the device interface pins for the respective processors.



LEGEND:

ACC = Accumulator
ARP = Auxiliary register pointer
AR0 = Auxiliary register 0
AR1 = Auxiliary register 1
DP = Data page pointer
PC = Program counter
P = P register
T = T register

Figure 3-1. TMS32010/C10/C15/E15 Block Diagram



LEGEND:

ACC	= ACCUMULATOR		PC	= PROGRAM COUNTER
ARP	= AUXILIARY REGISTER POINTER		P	= P REGISTER
AR0	= AUXILIARY REGISTER 0		T	= T REGISTER
AR1	= AUXILIARY REGISTER 1		TR	= TRANSMIT REGISTER
DP	= DATA PAGE POINTER		RR	= RECEIVE REGISTER

†TMS320C17/E17 only

Figure 3-2. TMS320C17/E17 Block Diagram

3.3 Internal Hardware Summary

The TMS320C1x internal hardware implements functions that other processors typically perform in software or microcode. For example, the device contains hardware for single-cycle 16 x 16-bit multiplication, data shifting, and address manipulation. This hardware-intensive approach provides computing power previously unavailable on a single chip.

Table 3-1 presents a summary of the TMS320C1x internal hardware. This summary table, which includes the internal processing elements, registers, and buses, is alphabetized within each functional grouping. All of the symbols used in this table correspond to the symbols used in the block diagrams of Section 3.2, the succeeding block diagrams in this section, and the text throughout this document.

Table 3-1. TMS320C1x Internal Hardware

UNIT	SYMBOL	FUNCTION
Accumulator	ACC	A 32-bit accumulator divided into a high-order word (bits 31 through 16) and a low-order word (bits 15 through 0). Used for storage of ALU output.
Arithmetic Logic Unit	ALU	A 32-bit two's-complement arithmetic logic unit having two 32-bit input ports and one 32-bit output port feeding the accumulator.
Auxiliary Registers	AR0,AR1	Two 16-bit registers used for data memory addressing and loop count control. Nine LSBs of each register are configured as up/down counters.
Auxiliary Register Pointer	ARP	A status bit that indicates the currently active auxiliary register.
Central Arithmetic Logic Unit	CALU	The grouping of the ALU, multiplier, accumulator, and shifters.
Data Bus	D(15-0)	A 16-bit bus used to route data to and from RAM.
Data Memory Page Pointer	DP	A status bit that points to the data RAM address of the current page. A data page contains 128 words.
Data RAM	-	144 or 256 words of on-chip random access memory containing data.
External Address Bus	A(11-0)/ PA(2-0)	A 12-bit bus used to address external program memory. The three LSBs are port addresses in the I/O mode.
Interrupt Flag	INTF	A single-bit flag that indicates an interrupt request has occurred (is pending).
Interrupt Mode	INTM	A status bit that masks the interrupt flag.
Multiplier	MULT	A 16 x 16-bit parallel hardware multiplier.
Overflow Flag	OV	A status bit flag that indicates an overflow in arithmetic operations.
Overflow Mode	OVM	A status bit that defines a saturated or unsaturated mode in arithmetic operations.
P Register	P	A 32-bit register containing the product of multiply operations.
Program Bus	P(15-0)	A 16-bit bus used to route instructions from program memory.
Program Counter	PC (11-0)	A 12-bit register used to address program memory. The PC always contains the address of the next instruction to be executed. The PC contents are updated following each instruction decode operation.
Program ROM/EPROM	-	1.5K or 4K words of on-chip read only memory (ROM or EPROM) containing the program code.
Shifters	-	Two shifters: the ALU barrel shifter that performs a left-shift of 0 to 16 bits on data memory words loaded into the ALU, and the accumulator parallel shifter that performs a left-shift of 0, 1, or 4 places on the entire accumulator and places the resulting high-order bits into data RAM.
Stack	-	A 4 x 12 hardware stack used to store the PC during interrupts or calls.

Table 3-1. TMS320C1x Internal Hardware (Concluded)

UNIT	SYMBOL	FUNCTION
Status Register	ST	A 16-bit status register that contains status and control bits.
T Register	T	A 16-bit register containing the multiplicand during multiply operations.
Additional Hardware on the TMS320C17/E17		
Companding Hardware	-	Data companding encoder/decoder in either µ-law or A-law PCM conversion format. Two modes of operation: serial mode for operating on serial port data (linear/logarithmic PCM conversions), or parallel mode for computation inside the device. Companding is selected through the control register.
Latched Data Bus	LD(15-0)	A 16-bit bidirectional latched data bus used in coprocessor mode. This bus is connected internally to two latches, one for input and one for output.
Serial Port Clock	SCLK	The clock that provides the timing control for data transfers with the serial port. SCLK is configured through the control register.
Serial Port Framing Control	FR	The FR signal provides serial port framing compatible with combo-codec devices. The FR pulse signifies a transmit/receive of new data on the serial port.
Serial Port Receive Registers	RR0,RR1	8-bit serial port registers that receive 8-bit data samples.
Serial Port Receive Shift Registers	RS0,RS1	8-bit registers used to shift in serial port data from pin DR0 or DR1.
Serial Port Transmit Registers	TR0,TR1	8-bit serial port transmit registers in a FIFO (first in, first out) configuration.
Serial Port Transmit Shift Registers	TS0,TS1	8-bit registers used to shift out serial port data onto pin DX0 or DX1.
System Control Register	CR(31-0)	A 32-bit register that controls interrupts, serial port channels, companding hardware, and coprocessor port channels. Control register 1, accessed through port 1, consists of the upper 16 bits (CR31-CR16). Control register 0, accessed through port 0, consists of the lower 16 bits (CR15-CR0).

3.4 Memory Organization

The TMS320C1x devices utilize a Harvard architecture, in which data and program memory reside in two separate spaces. The TMS320C1x provides 144/256 16-bit words of on-chip data RAM and 1.5K/4K words of program ROM. On-chip program EPROM versions are available. This section describes the TMS320C1x data and program memory, data movement, memory maps, auxiliary registers, microcomputer/microprocessor modes, and memory addressing modes.

3.4.1 Data Memory

Data memory consists of 144/256 words of 16-bit on-chip RAM (see Figure 3-3). The TMS32010/C10 provides 144 words. The TMS320C15/C17 offers expanded on-chip RAM of 256 words. See Section 3.4.4 for memory map configurations.

To expand data memory, the data operands may be stored off-chip, and then read into the on-chip RAM as they are needed. Two instruction pairs, TBLR/TBLW and IN/OUT, are available for accomplishing this. The table read (TBLR) instruction can transfer values from program memory, either on-chip ROM or off-chip ROM/RAM, to the on-chip data RAM. The table write (TBLW) instruction transfers values from the data RAM to off-chip program RAM. These instructions take three cycles to execute. When using the IN/OUT instruction pair, the IN instruction reads data from a peripheral and transfers it to the data RAM. With some extra hardware, the IN instruction, together with the OUT instruction, can be used to read and write from the data RAM to large amounts of external storage addressed as a peripheral. This method is faster since execution of the IN and OUT instructions takes only two cycles. See Section 6.1 for hardware applications using RAM/ROM expansion.

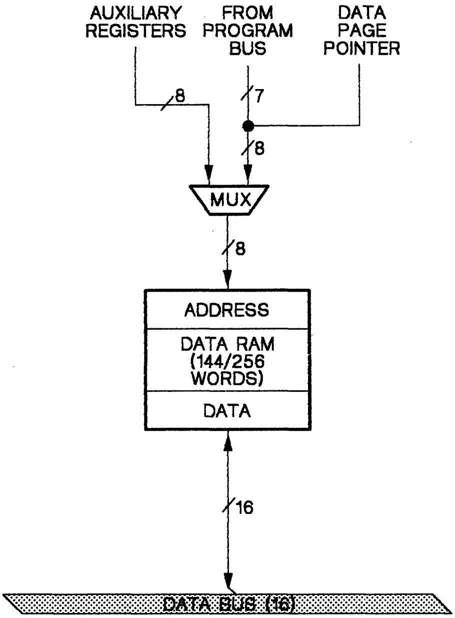


Figure 3-3. On-Chip Data Memory

3.4.2 Program Memory

Program memory consists of 1.5K/4K words on the TMS320C1x devices. The TMS32010/C10 provides 1.5K words, and the TMS320C15/C17 provides 4K words. The on-chip program ROM of up to 4K words allows program execution at full speed without the need for high-speed external program memory. On-chip program EPROM of 4K words, provided on the TMS320E15/E17, presents two additional benefits. First, application development is greatly facilitated since the EPROM can be directly programmed by the user. Second, these devices implement a security feature that can be used to protect proprietary algorithms by preventing the EPROM contents from being read.

Program memory operation is user-selectable by means of the MC/$\overline{\text{MP}}$ (microcomputer/microprocessor) pin. Setting MC/$\overline{\text{MP}}$ high places the device in the microcomputer mode. Holding the pin low places the device in the microprocessor mode.

In the microcomputer mode, only locations 0 through 1523 of the ROM on the TMS32010/C10 are available for the user's program. Locations 1524-1535 are reserved by Texas Instruments for testing purposes. The device architecture allows for an additional 2.5K words of program memory to reside off-chip on the TMS32010/C10. ROM locations 0 through 3999 on the TMS320C15/C17 are available for the user's program; locations 4000 through 4095 are reserved for testing purposes. Reserved locations may not be utilized by the user. In the microprocessor mode, all 4K words of memory are externa, Note that the microprocessor mode is not available for the TMS320C17/E17. See Section 3.4.4 for memory map configurations.

External RAM or ROM can be interfaced to the TMS320C1x (see Section 6.1) for those applications requiring external program memory space. This provides multiple functionality for external RAM-based systems. The TMS320C17/E17 provides no direct program memory expansion capability.

Twelve output pins are available for addressing external memory. These pins, A11 (MSB) through A0 (LSB), contain the buffered outputs of the program counter or the I/O port address. When an instruction is fetched from off-chip memory, the $\overline{\text{MEN}}$ (memory enable) strobe will be generated to enable the external memory. The instruction word is then transferred to the processor via the data bus (see Section 3.7).

When in the microcomputer mode, the processor selects internal program memory. The $\overline{\text{MEN}}$ strobe will still become active in this mode, and the address lines A11 through A0 will still output the current value of the program counter although the instruction word will be read from internal program memory. Note that $\overline{\text{MEN}}$ is never active at the same time as the $\overline{\text{WE}}$ or $\overline{\text{DEN}}$ signals. In effect, $\overline{\text{MEN}}$ will go low every clock cycle except when an I/O function is being performed by the IN, OUT, or TBLW instructions. In these multicycle instructions, $\overline{\text{MEN}}$ goes low during the clock cycles in which $\overline{\text{WE}}$ or $\overline{\text{DEN}}$ do not go low.

Figure 3-4 gives an example of external program memory expansion. Even when executing from external memory, the TMS320C1x performs at full speed. Note that some ports are reserved for on-chip peripheral logic.

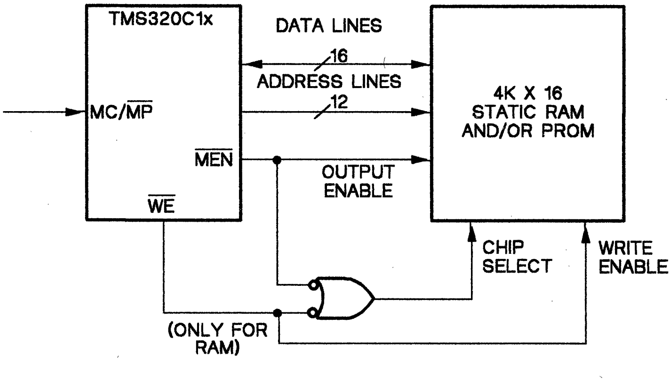


Figure 3-4. External Program Memory Expansion Example

3.4.3 Data Movement

The TMS320C1x provides instructions for data movement functions that efficiently utilize the on-chip RAM. The DMOV (data move) function is useful for implementing algorithms that use the z^{-1} delay operation, such as convolutions and digital filtering where data is being passed through a time window.

Implemented in on-chip RAM, the DMOV function allows a word to be copied from the currently addressed data memory location in on-chip RAM to the next higher location while the data from the addressed location is being operated upon in the same cycle (e.g., by the CALU). The LTD (load T register, accumulate previous product, and move data) instruction uses the data move function.

3.4.4 Memory Maps

The TMS320C1x devices provide three separate address spaces for program memory, data memory, and I/O, as shown in Figure 3-5 and Figure 3-6. Program memory is configured according to the state of the MC/$\overline{\text{MP}}$ pin. For further information about data and program memory, see Sections 3.4.1, 3.4.2, and 3.4.3. I/O functions are discussed in Section 3.7.

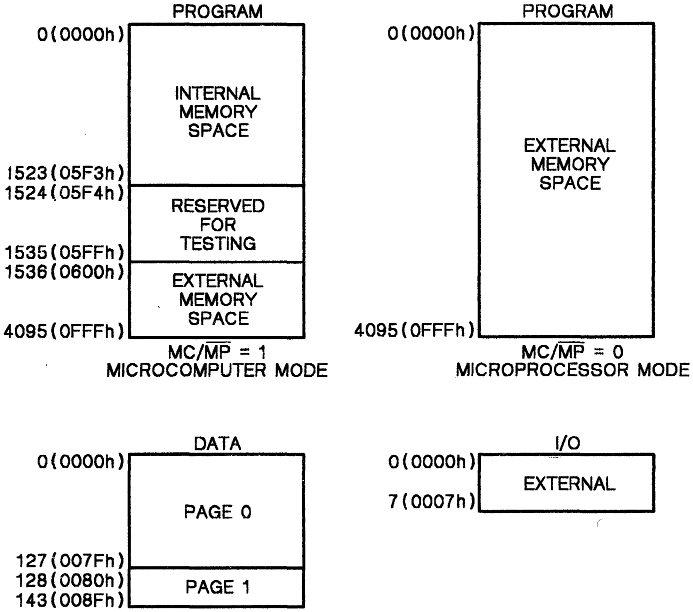
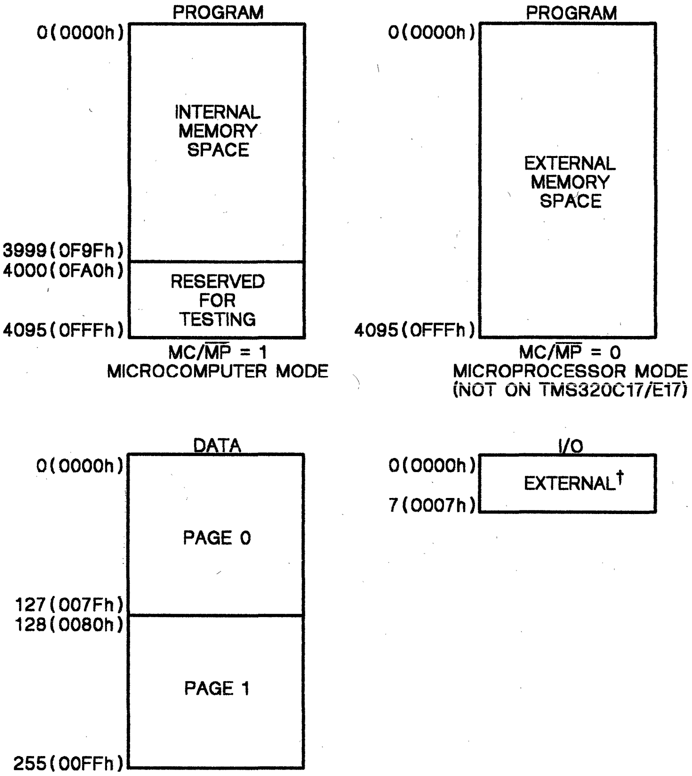


Figure 3-5. Memory Maps for the TMS32010/C10



†On the TMS320C17/E17, ports 0 and 1 are dedicated to the internal control register; no external I/O is available in the coprocessor mode.

Figure 3-6. Memory Maps for the TMS320C15/E15 and TMS320C17/E17

3.4.5 Auxiliary Registers

The TMS320C1x devices provide two 16-bit auxiliary registers (AR0 and AR1). This section discusses each register's function and how an auxiliary register is selected, loaded, and stored.

The auxiliary registers may be used for indirect addressing of data memory, temporary data storage, and loop control. Indirect addressing allows placement of the data memory address of an instruction operand into the least-significant eight bits of an auxiliary register. The registers are selected by a single-bit Auxiliary Register Pointer (ARP) that is loaded with a value of 0 or 1, designating AR0 or AR1, respectively. The ARP is part of the status register, and can be stored in memory.

When the auxiliary registers are autoincremented/decremented by an indirect addressing instruction or by the BANZ (branch on auxiliary register not zero) instruction, the lowest nine bits are affected (see Figure 3-7). The auxiliary registers are useful as counters when the BANZ instruction is used. This counter portion of an auxiliary register is a 9-bit counter, as shown in Figure 3-8 and Figure 3-9.

The upper seven bits of an auxiliary register (i.e., bits 9 through 15) are unaffected by any autoincrement/decrement operation. This includes autoincrement of 111111111 (the lowest nine bits go to 0) and autodecrement of 000000000 (the lowest nine bits go to 111111111); in each case, bits 9 through 15 are unaffected.

The auxiliary registers can be saved in and loaded from data memory with the SAR (store auxiliary register) and LAR (load auxiliary register) instructions. This is useful for performing context saves. SAR and LAR transfer entire 16-bit values to and from the auxiliary registers even though indirect addressing and loop counting utilize only a portion of the auxiliary register. See Section 4 for programming of the indirect addressing mode.

The BANZ instruction permits the auxiliary registers to also be used as loop counters. BANZ checks if an auxiliary register is zero. If not, it decrements and branches. See Section 5.3.3 for loop code using the auxiliary registers.

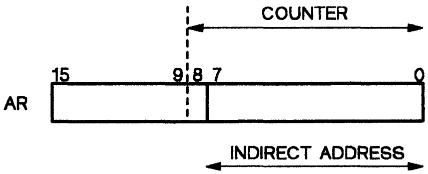


Figure 3-7. Auxiliary Register Counter

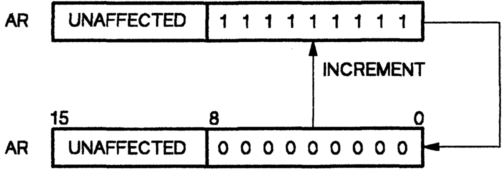


Figure 3-8. Indirect Addressing Autoincrement

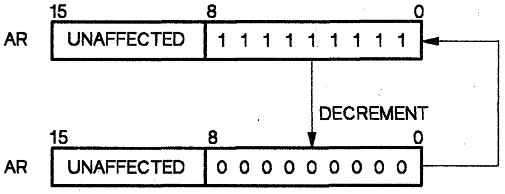


Figure 3-9. Indirect Addressing Autodecrement

3.4.6 Memory Addressing Modes

The TMS320C1x can address up to 4K words of program memory and up to 144/256 words of data memory. Three forms of instruction operand addressing can be used: direct, indirect, and immediate addressing. Figure 3-10 illustrates operand addressing in the three modes. The addressing modes are described in detail in Section 4.1.

In the direct addressing mode, the 1-bit data memory page pointer (DP) selects either page 0 consisting of memory locations 0-127 or page 1 consisting of locations 128-143/255. The data memory address (dma), specified by the seven LSBs of the instruction concatenated with the DP, addresses the desired word within the page. Note that DP is part of the status register and thus can be stored in data memory.

Indirect addressing uses the lower eight bits of the auxiliary registers as the data memory address. This is sufficient to address all 256 data words; no paging is necessary with indirect addressing. The current auxiliary register is selected by the auxiliary register pointer (ARP). In addition, the auxiliary registers can be made to autoincrement/decrement during any given indirect instruction. Note that the increment/decrement occurs after the current instruction is finished executing.

When an immediate operand is used, it is contained within the instruction word itself.

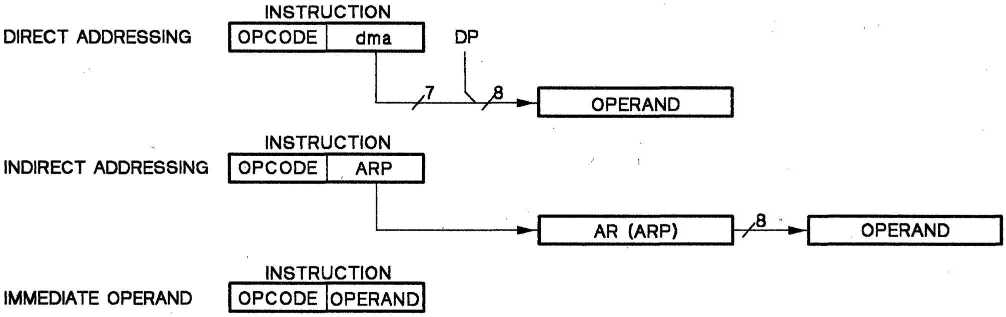


Figure 3-10. Methods of Instruction Operand Addressing

3.5 Central Arithmetic Logic Unit (CALU)

The Central Arithmetic Logic Unit (CALU) contains a 16 x 16-bit parallel multiplier, a 32-bit Arithmetic Logic Unit (ALU), a 32-bit accumulator (ACC), and two shifters. This section describes the CALU components and their functions. Figure 3-11 is a block diagram showing the components of the CALU.

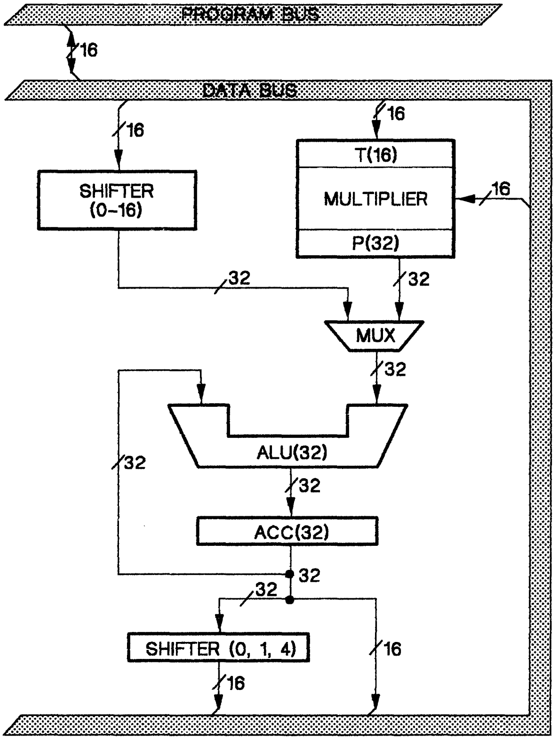


Figure 3-11. Central Arithmetic Logic Unit (CALU)

The following steps occur in the implementation of a typical ALU operation:

1) Data is fetched from the RAM on the data bus.

2) Data is passed through the barrel shifter where it can be left-shifted 0 to 16 bits, depending on the value specified by the instruction.

3) Data enters the ALU where it is operated upon and loaded into the accumulator.

4) The result obtained in the accumulator is passed through a parallel left-shifter present at the accumulator output to aid in scaling results.

5) The result is stored in the data RAM. Since the accumulator is 32 bits wide, both halves must be stored separately.

One input to the ALU is always provided from the accumulator, and the other input may be provided from the P Register of the multiplier or the barrel shifter that is loaded from data memory.

3.5.1 Shifters

Two shifters are available for manipulating data: a barrel shifter for shifting data from the data RAM into the ALU and a parallel shifter for shifting the accumulator into the data RAM (see Figure 3-11).

The barrel shifter has a 16-bit input connected to the data bus and a 32-bit output connected to the ALU The barrel shifter produces a left shift of 0 to 16 bits on all data memory words that are loaded into, subtracted from, or added to the accumulator by the LAC, SUB, and ADD instructions. The shifter zero-fills the LSBs and sign-extends the 16-bit data memory word to 32 bits by an arithmetic left-shift (i.e., the bits to the left of the MSB of the data word are filled with ones if the MSB is a one or with zeros if the MSB is a zero). This differs from a logical left-shift where the bits to the left of the MSB are always filled with zeros. A small amount of code is required to perform an arithmetic right-shift or a logical right-shift.

The following examples illustrate the barrel shifter's function:

- Data memory location 20 holds the two's-complement number: 7EBCh.

  The LAC (load accumulator) instruction is executed, specifying a left-shift of 4:

  ```
  LAC   20,4
  ```

  The accumulator then holds the following 32-bit signed two's-complement number:

31 16	15 0
0 0 0 7	E B C 0

  Since the MSB of 7EBCh is a zero, the upper accumulator was zero-filled.

- Data memory location 30 holds the two's-complement number: 8EBCh.

  The LAC (load accumulator) instruction is executed, specifying a left-shift of 8:

  ```
  LAC   30,8
  ```

  The accumulator then holds the following 32-bit signed two's-complement number:

31 16	15 0
F F 8 E	B C 0 0

  Since the MSB of 8EBCh is a one, the upper accumulator was filled with ones.

Instructions are provided that perform operations with the lower half of the accumulator and a data word without first sign-extending the data word (i.e., treating it as a 16-bit rather than a 32-bit word). The mnemonics of these instructions typically end with an 'S,' indicating that sign-extension is suppressed (e.g., ADDS, SUBS). Along with the instructions that operate on the upper half of the accumulator, these instructions allow the manipulation of 32-bit precision numbers.

The parallel shifter is activated only by the SACH (store high-order accumulator word) instruction. This instruction causes the shifter to be loaded with the 32-bit contents of the accumulator. The data is then left-shifted. The most-significant 16 bits from the shifter are stored in RAM, resulting in a loss of the high-order bits of data. The contents of the accumulator remain unchanged. The parallel shifter can execute a shift of only 0, 1, or 4. Shifts of 1 and 4 are used with multiplication operations. No right-shift is directly implemented. The following example illustrates the accumulator shifter's function:

- The accumulator holds the following 32-bit signed two's-complement number:

31 16	15 0
A 3 4 B	7 8 C D

The SACH instruction is executed, specifying that a left-shift of four be performed on the high-order accumulator word before it is stored in data memory location 40:

```
SACH 40,4
```

Data memory location 40 then contains the two's-complement number: 34B7h. The accumulator still retains 0A34B78CDh.

3.5.2 ALU and Accumulator

The 32-bit ALU and accumulator (see Figure 3-11) implement a wide range of arithmetic and logical functions, the majority of which execute in a single clock cycle. Once an operation is performed in the ALU, the result is transferred to the accumulator where additional operations such as shifting may occur. Data that is input to the ALU may be scaled by the barrel shifter.

The ALU is a general-purpose arithmetic logic unit that operates on 16-bit data words, producing a 32-bit result. The ALU can add, subtract, and perform logical operations. The accumulator is always the destination and the primary operand. The result of logical operations is shown in Table 3-2. A data memory value (dma) is the operand for the lower half of the accumulator (bits 15 through 0). Zero is the operand for the upper half of the accumulator.

Table 3-2. Accumulator Results of a Logical Operation

FUNCTION	ACC BITS 31-16	ACC BITS 15-0
XOR	(0).XOR.(ACC (31-16))	(dma).XOR.(ACC (15-0))
AND	(0).AND.(ACC (31-16))	(dma).AND.(ACC (15-0))
OR	(0).OR.(ACC (31-16))	(dma).OR.(ACC (15-0))

The 32-bit accumulator stores the output from the ALU and is also often an input to the ALU. The accumulator is divided into two 16-bit words for storage in data memory: a high-order word (bits 31 through 16) and a low-order word (bits 15 through 0). The SACH and SACL instructions are used to store the high- and low-order accumulator words in data memory. These instructions can be used in the implementation of double-precision arithmetic.

A shifter at the output of the accumulator provides a left-shift of 0, 1, or 4 places. This shift is performed while the data is being transferred to the data bus for storage. The contents of the accumulator remain unchanged. When the high-order word is shifted left, the LSBs are transferred from the low-order word, and the MSBs are lost.

The accumulator also has the ability to simulate the effect of saturation in analog systems. This capability is implemented using the accumulator overflow saturation mode, which is controlled by the OVM (overflow mode) status register bit. The accumulator saturation mode is enabled or disabled by setting or resetting the OVM bit, respectively, through the use of the SOVM and ROVM (set and reset OVM bit) instructions. If OVM is set and accumulator operation results in an overflow, the accumulator is loaded with either the largest positive or negative number, depending on the sign of the operands and the actual result. The value of the accumulator upon saturation is 7FFFFFFFh (positive) or 80000000h (negative). If OVM is reset and an overflow occurs, the overflowed results are loaded into the accumulator without modification. (Note that logical operations cannot result in overflow.)

It is particularly desirable to enable the saturation mode when the accumulator contents represent a signal value, since without saturation mode enabled, overflows cause undesirable discontinuities in the represented waveform. When saturation mode is enabled, behavior of the accumulator more closely resembles the tendency of an analog system to limit or saturate at a maximum level when subjected to excessively large size signals.

When an overflow occurs, the OV (overflow) bit in the status register is set, regardless of whether or not the OVM bit is set. The BV (branch on overflow) instruction, which branches only if OV is set, can be used to allow programs to make decisions based on whether or not an overflow has occurred and act accordingly. Once set, OV is reset only by the BV instruction, or by directly loading the status register. Since OV is part of the status register, its state can be stored in data memory using the SST (store status register) instruction or loaded using the LST (load status register) instruction. This allows the state of OV from different program contexts to be saved independently, if desired, and examined outside of time-critical code segments.

The TMS320C1x also has the capability of executing branch instructions that depend on the status of the ALU and accumulator. These instructions (BLZ, BLEZ, BGEZ, BGZ, BNZ, and BZ) cause a branch to be executed if a specific condition is met (see Section 4 for a complete list of TMS320C1x instructions).

3.5.3 Multiplier, T and P Registers

The TMS320C1x utilizes a 16 x 16-bit hardware multiplier (see Figure 3-11), which is capable of computing a 32-bit product in a single machine cycle. The following two registers are associated with the multiplier:

- A 16-bit Temporary Register (T) that holds one of the operands for the multiplier, and
- A 32-bit Product Register (P) that holds the product.

In order to use the multiplier, an operand must first be loaded into the T register from the data bus using an LT, LTA, or LTD instruction. Then, the MPY (multiply) or MPYK (multiply immediate) instruction provides the second operand (also from the data bus). If the MPY instruction is used, the multiplier value is a 16-bit number. If the MPYK instruction is used, the value is a 13-bit immediate constant contained in the MPYK instruction word. This 13-bit constant is right-justified and sign-extended. After execution of the multiply instruction, the product will be placed in the P register. The product can then be added to, subtracted from, or loaded into the accumulator by executing a PAC, APAC, SPAC, LTA, or LTD instruction. Pipelined multiply and accumulate operations can be accomplished with the LTA/LTD and MPY/MPYK instructions. Note that no provisions are made for the condition of 8000h × 8000h. If this condition arises, the product will be 0C0000000h.

Note that the contents of the P register cannot be restored without altering other registers. Interrupts are prevented from occurring until the instruction following the MPY/MPYK instruction has been executed. Therefore, the multiply instruction should always be followed by an instruction that combines the P register with the accumulator.

3.6 System Control

System control on the TMS320C1x processors is provided by the program counter and stack, the external reset signal, interrupts (see Section 3.8), and the status register. This section explains the function of these components in system control. On the TMS320C17/E17, a system control register controls the operation of the serial port, companding hardware, and the operation of the coprocessor port. The system control register for the TMS320C17/E17 is discussed in Section 3.12.

3.6.1 Program Counter and Stack

The program counter and stack enable the execution of branches, subroutine calls, interrupts, and table read/table write instructions. The program counter (PC) is a 12-bit register that contains the program memory address of the next instruction to be executed. The TMS320C1x reads the instruction from the program memory location addressed by the PC and increments the PC in preparation for the next instruction prefetch. The PC is initialized to zero by activating the reset ($\overline{\text{RS}}$) line.

The TMS320C1x devices utilize a modified Harvard architecture in which data memory and program memory lie in two separate spaces, thus permitting a full overlap of instruction fetch and execution. Figure 3-12 outlines the overlap of the instruction prefetch and execution. On the falling edge of CLKOUT, the program counter (PC) is loaded with the address of the instruction (load PC 2) to be prefetched while the current instruction (execute 1) is decoded and begins execution. The next instruction is then fetched (fetch 2) while the current instruction continues to execute (execute 1). Even as another prefetch occurs (fetch 3), both the current instruction (execute 2) and the previous instruction are still executing. This is possible because of a highly pipelined internal structure.

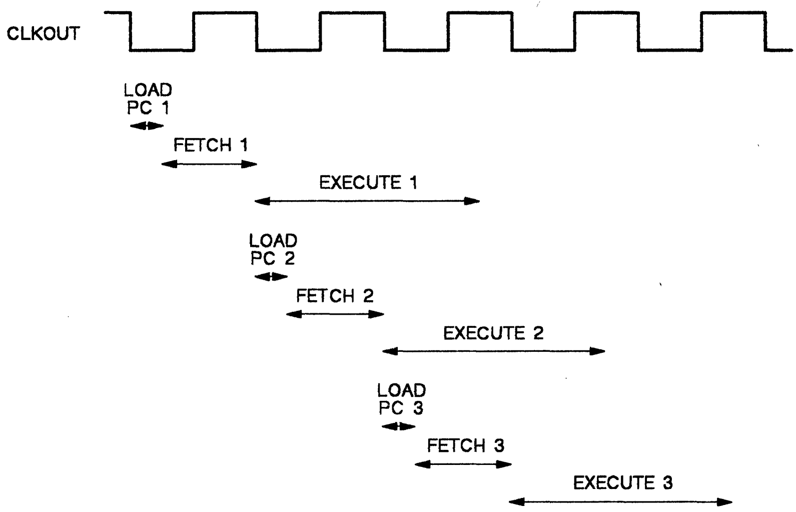


Figure 3-12. Instruction Pipeline Operation

To permit the use of external program memory, the PC outputs are buffered and sent to the external address bus pins, A11 through A0. The PC outputs appear on the address bus during all modes of operation. The nine MSBs of the PC (A11 through A3) have unique outputs assigned to them, while the three LSBs are multiplexed with the port address lines, PA2 through PA0. The port address field is used by the I/O instructions, IN and OUT.

Program memory is always addressed by the contents of the PC. The contents of the PC can be changed by a branch instruction if the particular branch condition being tested is true. Otherwise, the branch instruction simply increments the PC. All branches are absolute, rather than relative, i.e., a 12-bit value derived from the branch instruction word is loaded directly into the PC in order to accomplish the branch. When interrupts or subroutine call instructions occur, the contents of the PC are pushed onto the stack to preserve return linkage to the previous program context.

The stack is 12 bits wide and four levels deep. The PC stack is accessible through the use of the PUSH and POP instructions. The PUSH instruction pushes the twelve LSBs of the accumulator onto the top of the stack (TOS). Whenever the contents of the PC are pushed onto the TOS, the previous contents of each level are pushed down, and the fourth location of the stack is lost. Therefore, data will be lost if more than four successive pushes (stack overflow) occur before a pop. The reverse happens on pop operations. The POP instruction pops the TOS into the twelve LSBs of the accumulator. Any pop after three sequential pops yields the value at the fourth stack level. All four stack levels then contain the same value. Following the POP instruction, the TOS can be moved into data memory by storing the low-order accumulator word (SACL instruction). This allows expansion of the stack into data RAM. From data RAM, it can easily be copied into program RAM off-chip by using the TBLW (table write) instruction. In this way, the stack can be expanded to very large levels.

Note that the TBLR and TBLW instructions utilize one level of the stack; therefore, only three nested subroutines or interrupts can be accommodated without stack overflow occurring.

To handle subroutines and interrupts of much higher nesting levels, part of the data RAM or external RAM can be allocated to stack management. In this case, the TOS is popped immediately at the start of a subroutine or interrupt routine and stored in RAM. At the end of the subroutine or interrupt routine, the stack value stored in RAM is pushed back onto the TOS before returning to the main routine.

3.6.2 Reset

Reset ($\overline{\text{RS}}$) is a non-maskable external interrupt that can be used at any time to put the TMS320C1x into a known state. Reset is typically applied after powerup when the machine is in a random state. The reset input must be held low for a minimum of five clock cycles.

Driving the $\overline{\text{RS}}$ signal low causes the TMS320C1x to terminate execution and forces the program counter to zero. $\overline{\text{RS}}$ affects various registers and status bits. At powerup, the state of the processor is undefined. For correct system operation after powerup, a reset signal must be asserted low to guarantee a reset of the device (see Section 5.1 for other important reset considerations). Processor execution begins at location 0, which normally contains a B (branch) statement to also direct program execution to the system initialization routine (see Section 5.1 for an initialization routine example).

Upon receiving an $\overline{\text{RS}}$ signal, the following actions take place:

1) The control lines for $\overline{\text{DEN}}$, $\overline{\text{WE}}$, and $\overline{\text{MEN}}$ are forced high.

2) The data bus D15-D0 is placed in the high-impedance state.

3) The Program Counter (PC) is set to 0, and the address bus A11-A0 is driven with all zeroes after the next clock cycle from $\overline{\text{RS}}$ going low.

4) The interrupt is disabled, and the interrupt flag register is reset to all zeroes.

5) Control register bits on the TMS320C17/E17 are set as follows: CR11 is set to 0, CR15 to 1, and CR29 to 0.

The TMS320C1x can be held in the reset state indefinitely. Note that the ARP, DP, and OVM status bits are not initialized by reset. Accordingly, it is critical that these bits be initialized in software by the user following reset.

3.6.3 Status Register

The status register consists of five status bits. These status bits can be individually altered through dedicated instructions. In addition, the SST instruction provides for storing the status register in data memory. The LST instruction loads the status register from data memory, with the exception of the INTM bit. This bit can be changed only by the EINT/DINT (enable/disable interrupt) instructions. In this manner, the current status of the device may be saved on interrupts and subroutine calls.

Table 3-3 shows instructions that affect the status register contents. Note that several bits in the status registers are reserved and read from the status register as logic ones by the SST instruction.

Table 3-3. Status Register Field Definitions

FIELD	FUNCTION
ARP	Auxiliary Register Pointer. This single-bit field selects the AR to be used in indirect addressing. ARP = 0 selects AR0; ARP = 1 selects AR1. ARP may be modified by executing instructions that permit the indirect addressing option, and by the LARP, MAR, and LST instructions.
DP	Data Memory Page Pointer. The single-bit DP register is concatenated with the 7 LSBs of an instruction word to form a direct memory address of 8 bits. DP = 0 selects the first 128 words of data memory, i.e., page 0. DP = 1 selects page 1, the remaining words in data memory. DP may be modified by the LST, LDP, and LDPK instructions.
INTM	Interrupt Mode Bit. When an interrupt is serviced, the INTM bit is automatically set to one before the interrupt service routine begins. INTM = 0 enables all maskable interrupts; INTM = 1 disables all maskable interrupts. INTM is set and reset by the DINT and EINT instructions, respectively. $\overline{\text{RS}}$ also sets INTM. INTM has no effect on the unmaskable $\overline{\text{RS}}$ interrupt. Note that INTM is unaffected by the LST instruction.
OV	Overflow Flag. OV = 0 indicates that the accumulator has not overflowed. OV = 1 indicates that an overflow has occurred. Once an overflow occurs, the OV remains set until a reset, BV, or LST instruction clears the OV.
OVM	Overflow Mode Bit. OVM = 0 disables the overflow mode, causing the overflowed results to remain in the accumulator. OVM = 1 enables the overflow mode, causing the accumulator to be set to either its most positive or negative value upon encountering an overflow. The SOVM and ROVM instructions set and reset this bit. LST may also be used to modify the OVM.

The contents of the status register can be stored in data memory by executing the SST instruction. If the SST instruction is executed using the direct addressing mode, the device automatically stores this information on page 1 of data memory at the location specified by the instruction. Thus, an SST instruction using the direct addressing mode can only specify an address less than 16 on the TMS32010/C10 since the second page of memory contains only 16 words. The second page of memory on the TMS320C15/E15 and TMS320C17/E17 contains 128 words. If the indirect addressing mode is selected, the contents of the status register may be stored in any RAM location selected by the auxiliary register.

The SST instruction does not modify the contents of the status register. Figure 3-13 shows the position of the status bits as they appear in the appropriate data RAM location after execution of the SST instruction.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OV	OVM	INTM	1	1	1	1	ARP	1	1	1	1	1	1	0	DP

Figure 3-13. Status Register Organization

The LST instruction may be executed to load the status register. LST does not assume status bits are on page one. When direct memory addressing has been used, the DP must be set to one for the LST instruction to access status bits stored on page one. The interrupt mode (INTM) bit cannot be changed by the LST instruction. However, all other status bits can be modified by this instruction.

3.7 Input/Output Functions

The TMS320C1x implements a variety of different I/O functions for use in communicating with external devices. The 16-bit parallel data bus can be utilized to perform I/O functions in two cycles using the IN and OUT instructions. The I/O ports are addressed by the three LSBs of the address bus (PA2-PA0). In addition to I/O functions, a polling input ($\overline{\text{BIO}}$) for both bit test and branch operations and an interrupt input ($\overline{\text{INT}}$) have been incorporated for increased system flexibility. An external flag output pin (XF) is available on the TMS320C17/E17 to implement single-bit digital output.

I/O design is simplified by having I/O treated the same way as memory. I/O devices are mapped into the I/O address space using the processor's external address and data buses in the same manner as memory-mapped devices.

Input/output of data to and from a peripheral is accomplished by the IN and OUT instructions. Data is transferred over the 16-bit data bus to and from data memory by two independent strobes: data enable ($\overline{\text{DEN}}$) and write enable ($\overline{\text{WE}}$).

The bidirectional external data bus is always in the high-impedance state, except when $\overline{\text{WE}}$ is active (low), or during an IN instruction from port 0 or port 1 on the TMS320C17/E17 (see Section 3.7.1). $\overline{\text{WE}}$ goes low during the first cycle of the OUT instruction and the second cycle of the TBLW instruction.

Eight I/O addresses are available on the TMS32010/C10 and TMS320C15/E15 for interfacing to peripheral devices: eight 16-bit multiplexed input ports and eight 16-bit multiplexed output ports (see Figure 3-14). Since the system control register, serial port transmit and receive registers, and companding hardware have been mapped into I/O ports 0 and 1, only six input and six output ports are available on the TMS320C17/E17 for interfacing to peripheral devices.

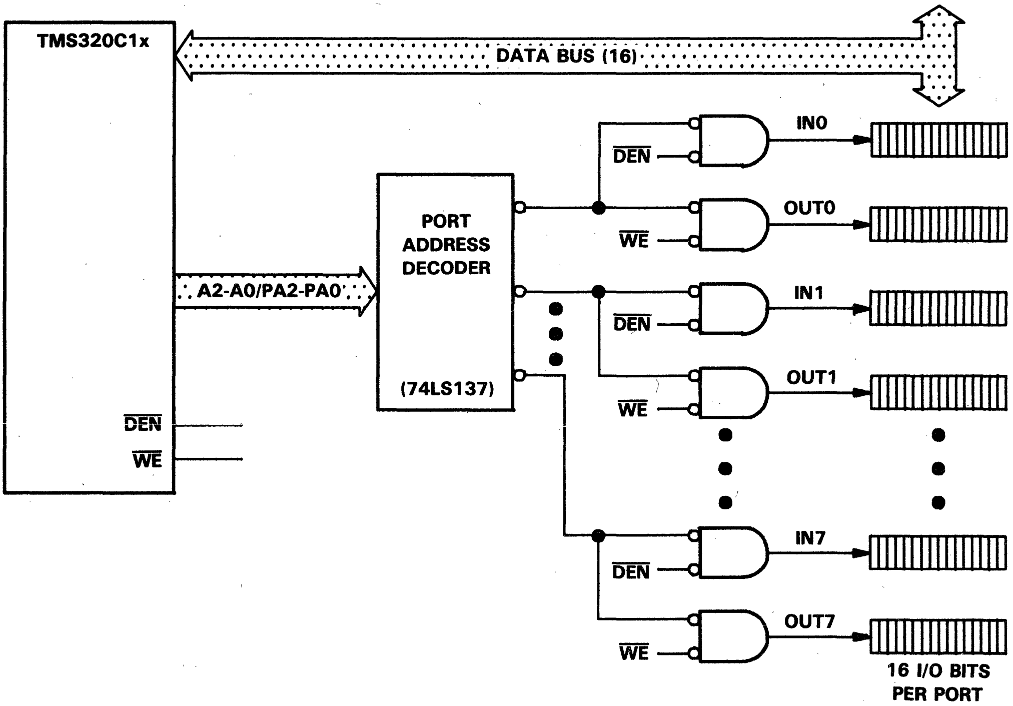


Figure 3-14. TMS320C1x External Device Interface

3.7.1 Input/Output Operation

The three port address pins (PA2-PA0) output the port address during IN and OUT instructions. Execution of an IN instruction generates the $\overline{\text{DEN}}$ strobe for transferring data from a peripheral device to the data RAM (see Figure 3-15). The IN instruction is the only instruction for which $\overline{\text{DEN}}$ will become active. Execution of an OUT instruction generates the $\overline{\text{WE}}$ strobe for transferring data from the data RAM to a peripheral device (see Figure 3-16). $\overline{\text{WE}}$ becomes active only during the OUT and TBLW (table write) instructions; see Appendix A for timing information.

When the three multiplexed LSBs of the address bus (PA2 through PA0) are used as a port address by the IN or OUT instruction, the remaining higher-order bits of the address bus (A11 through A3) are held at logic zero during execution of either instruction.

On the TMS320C17/E17, the purpose and usage of these pins (PA2 - PA0) are dependent upon the selected mode of operation (MC/$\overline{\text{MP}}$). In the microcomputer mode, the pins output the three LSBs of the program counter except during the IN and OUT instructions. During IN and OUT instructions, these three pins address the serial port, companding hardware, and off-chip I/O peripherals. During reset, the pins along with the program counter are synchronously cleared to zero during the cycle following $\overline{\text{RS}}$ low. Because program and data memories are contained on-chip, only these three address lines

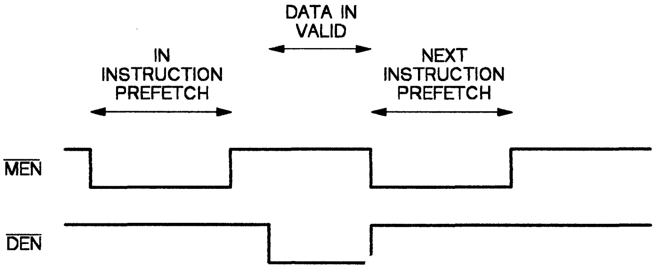


Figure 3-15. Input Instruction Timing

are outputted from the device. The memory enable ($\overline{\text{MEN}}$) signal is not implemented on the TMS320C17/E17 devices since all instruction execution is from on-chip program ROM. Additionally, the bidirectional external data bus on the TMS320C17/E17 is always in the high-impedance state, except when $\overline{\text{WE}}$ is active (low) or during an IN instruction from port 0 or port 1. $\overline{\text{WE}}$ goes low during the first cycle of the OUT instruction to provide the write strobe for writing data to a peripheral.

In the coprocessor mode, these pins (PA2 - PA0) have a different function; respectively referred to as $\overline{\text{TBLF}}$ (transfer buffer latch full), $\overline{\text{RBLE}}$ (receive buffer latch empty), and HI/$\overline{\text{LO}}$ (high or low transfer select of an 8-bit byte). Except for IN and OUT instructions to Port 5, no other activity will be seen on these pins. In this mode, the IN and OUT instructions to Ports 0 and 1 also provide the processor with an access to the on-chip serial ports and companding hardware and to the coprocessor port latches. And, the data bus will be in a high-impedance state, unless $\overline{\text{RD}}$ is active (low).

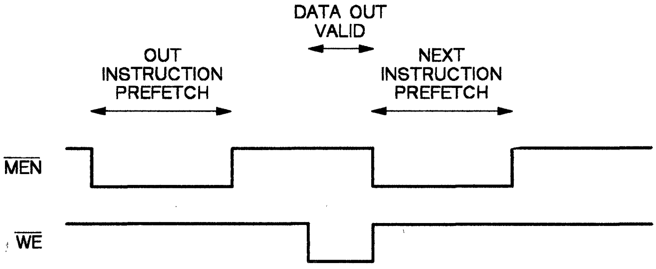


Figure 3-16. Output Instruction Timing

On the TMS320C17/E17, the system control register (see Section 3.12), serial port transmit and receive registers (Sections 3.9.1 and 3.9.2), and the companding hardware (Section 3.10) have been mapped into I/O ports 0 and 1.

During an OUT or IN instruction to port 0 or port 1, data appears on the external data bus (D15-D0). The data bus is not in the high-impedance state while accessing these dedicated I/O ports. Peripheral device interface should be to port addresses 2 through 7 to prevent bus conflicts with the system control register and serial port. Six 16-bit multiplexed input ports and six 16-bit multiplexed output ports are available for interfacing to peripheral devices.

3.7.2 Table Read/Table Write Operation

The TBLR and TBLW instructions allow words to be transferred between program and data spaces. TBLR is used to read words from on-chip ROM or off-chip program ROM/RAM into the data RAM. TBLW is used to write words from on-chip data RAM to off-chip program RAM on the TMS32010/C10/C15. External program memory cannot be addressed on the TMS320C17/E17.

Execution of the TBLR instruction generates $\overline{\text{MEN}}$ strobes to read the word from program memory (see Figure 3-17). Execution of a TBLW instruction generates a $\overline{\text{WE}}$ strobe (see Figure 3-18). Note that the data bus will be driven and the $\overline{\text{WE}}$ strobe will be generated even if the device is in the microcomputer mode and a TBLW is performed to a program location residing in on-chip ROM.

The dummy prefetch in Figure 3-17 and Figure 3-18 is a prefetch of the instruction following the TBLR or TBLW instruction and is discarded. The instruction following TBLR or TBLW is prefetched again at the end of the TBLR or TBLW instruction.

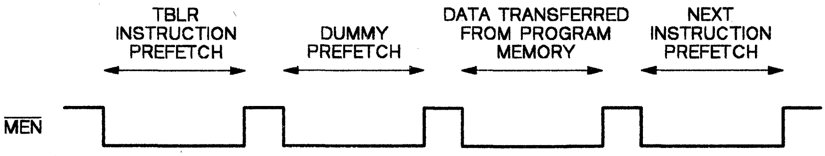

TBLR INSTRUCTION PREFETCH
DUMMY PREFETCH
DATA TRANSFERRED FROM PROGRAM MEMORY
NEXT INSTRUCTION PREFETCH
MEN

Figure 3-17. TBLR Instruction Timing

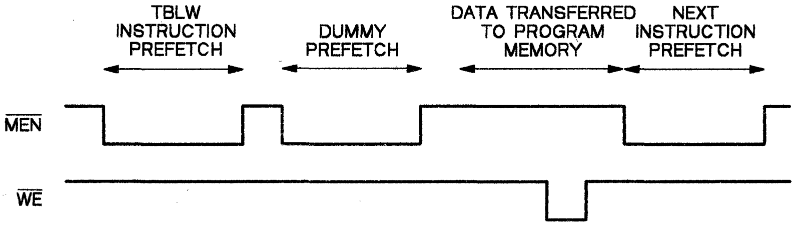

TBLW INSTRUCTION PREFETCH
DUMMY PREFETCH
DATA TRANSFERRED TO PROGRAM MEMORY
NEXT INSTRUCTION PREFETCH
MEN
WE

Figure 3-18. TBLW Instruction Timing

The $\overline{\text{MEN}}$, $\overline{\text{DEN}}$, and $\overline{\text{WE}}$ interface stobes are mutually exclusive. There are some very important considerations for those designs that utilize program memory. Since the OUT and TBLW instructions use only the $\overline{\text{WE}}$ signal to indicate valid data, these instructions cannot be distinguished from one another on the basis of the interface strobes. Execution of TBLW instructions will write data to peripherals, and execution of OUT instructions will overwrite program memory locations 0 through 7. Since it is impossible to use TBLW to uniquely write to program memory locations 0 through 7, it is advisable to avoid mapping both I/O and external program RAM into locations 0 through 7.

3.7.3 General-Purpose I/O Pins ($\overline{\text{BIO}}$ and XF)

The TMS320C1x provides two general-purpose pins which are software-controlled. The $\overline{\text{BIO}}$ pin is a branch control input pin for all of the TMS320C1x processors. The XF pin on the TMS320C17/E17 is an external flag output pin.

The $\overline{\text{BIO}}$ pin is an external pin that supports bit test and branch operations. When the $\overline{\text{BIO}}$ input pin is active (low), execution of the BIOZ instruction causes a branch to occur. The $\overline{\text{BIO}}$ pin is useful for monitoring peripheral device status. It is especially useful as an alternative to using an interrupt when time-critical loops must not be disturbed.

For systems using asynchronous inputs to the $\overline{\text{BIO}}$ pin on a TMS32010 (NMOS) device, external hardware is required to ensure proper execution of the BIOZ instruction. This hardware synchronizes the $\overline{\text{BIO}}$ input signal with the rising edge of CLKOUT on the TMS32010. See Appendix A for information regarding this system design consideration.

The XF (external flag) output pin, specific to the TMS320C17/E17, is an external logic output flag. Programmed through control register bit 10 (CR10), this pin is the direct output of the CR10 latch. When the CR10 bit is set to a 1, the XF pin is set to a logic high; when CR10 is reset to a 0, the XF pin is driven low.

3.8 Interrupts

The TMS320C1x provides an external interrupt input for communication with time-critical external operations. The interrupt can be generated either by applying a negative-going edge or a logic low level to the interrupt input pin. On the TMS320C17/E17, there are also three additional internal interrupts, which are generated by the two serial ports. All interrupts on the TMS320C1x are maskable through the use of the status register interrupt mode bit and various mask bits. When operating in the coprocessor mode on the TMS320C17/E17, the $\overline{\text{EXINT}}$ and $\overline{\text{BIO}}$ pins will be ignored. Internally comparable signals are supplied as a result of pulses on the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ pins in the coprocessor mode.

For systems using asynchronous inputs to the interrupt ($\overline{\text{INT}}$) pin on a TMS32010 (NMOS) device, external hardware is required to ensure proper processing of interrupts. This hardware synchronizes the $\overline{\text{INT}}$ input signal with the rising edge of CLKOUT on the TMS32010. See Appendix A for information regarding this system design consideration.

A simplified diagram of the internal interrupt circuitry for TMS320C1x CMOS devices is shown in Figure 3-19. Note that the TMS32010 requires external synchronizing flip-flops on interrupts and $\overline{\text{BIO}}$. These synchronizing flip-flops are not required on the TMS320C10/C15/C17.

When interrupts are enabled, an interrupt becomes active either due to a low-voltage input on the $\overline{\text{INT}}$ pin or when a negative edge has been latched into the interrupt flag (INTF). If the interrupt mode register (INTM) is set to zero, an interrupt active signal to the internal interrupt processor becomes valid.

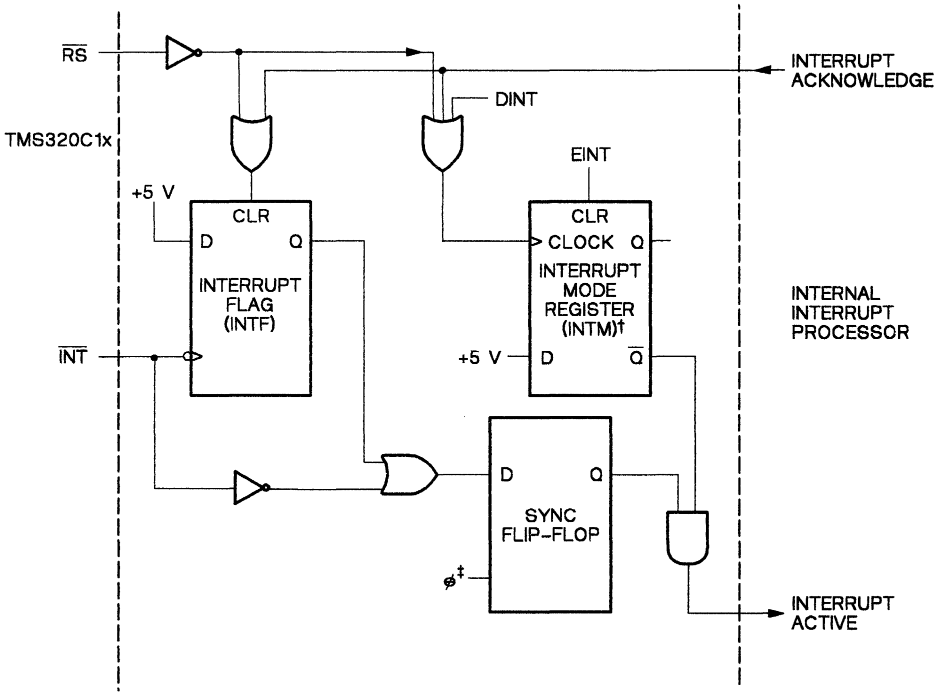
Interrupt servicing begins with the following sequence of events:

1) The interrupt is acknowledged, which clears the INTF (interrupt flag) bit to 0.
2) The INTM (interrupt mode) bit is set to 1 to disable further interrupts.
3) The current PC is pushed onto TOS (top of stack).
4) The new PC is set to 2.

The user begins interrupt servicing at program memory address 2h. At the end of the interrupt servicing, the user executes an EINT instruction to clear the INTM register (set to zero) to enable the interrupts. A DINT instruction or a hardware reset will also set the INTM register to one (see Figure 3-19), disabling interrupts. The user must execute an EINT instruction to enable interrupts again.

Note that interrupt servicing will be delayed in each of the following cases:

1) Until the end of all cycles of a multicycle instruction,
2) Until the instruction following the MPY or MPYK instruction has completed, or
3) Until the instruction following the EINT instruction has been executed (when interrupts have been previously disabled). This allows the RET instruction to be executed after interrupts become enabled at the end of an interrupt routine.



† Q = 0 indicates interrupts enabled.
Q = 1 indicates interrupts disabled.
‡ ϕ = phase of internal clock.
NOTE: The TMS32010 requires external synchronizing flip-flops.

Figure 3-19. TMS320C1x Simplified Interrupt Logic Diagram

Figure 3-20 shows the instruction sequence that occurs once an interrupt becomes active. The dummy fetch is an instruction that is fetched but not executed. This instruction will be refetched and executed after the interrupt routine is completed.

The TMS320C17/E17 has four maskable interrupts: $\overline{\text{EXINT}}$, $\overline{\text{FSR}}$, $\overline{\text{FSX}}$, and FR. On these devices, the TMS32010/C10/C15 interrupt function has been expanded to fully support the serial-port interface. An interrupt latch and multiplexer is used to generate the master interrupt signal, which functions identically to the $\overline{\text{INT}}$ interrupt on the TMS32010. Thus, all the maskable interrupts have the same priority and require the use of interrupt polling techniques when multiple interrupts are enabled.

Two steps must be taken to enable an active interrupt to the device. First, the individual interrupt must be enabled by writing a logic 1 to the appropriate system control register bit (CR7-CR4). Then, the master interrupt circuitry is enabled via the EINT instruction. An interrupt flag represents a valid interrupt condition to the processor if interrupts have been enabled. Thus, prior to enabling interrupts, the flag bits of all undesired interrupts should be cleared. In the coprocessor port mode, the external interrupt ($\overline{\text{EXINT}}$) flag cannot be cleared until four cycles after the data from the coprocessor port has been read. In a reset initialization routine, the interrupt flag bits (CR3-CR0) should be cleared before the EINT instruction to insure that a false interrupt does not occur (see Section 3.12 for detailed interrupt bit descriptions).

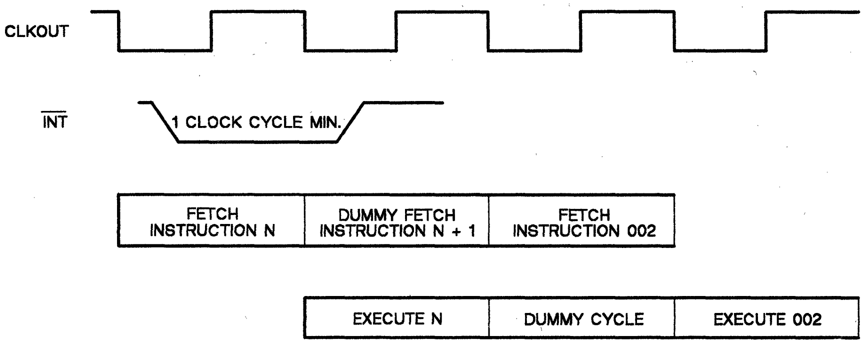


Figure 3-20. Interrupt Timing

The interrupt latch synchronizes all interrupts to the device output clock (CLKOUT). A block diagram of the interrupt latch and multiplexer is shown in Figure 3-21.

The external interrupt flag ($\overline{\text{EXINT}}$) is set by one of two conditions: (1) an asynchronous input to the device for external control, or (2) a master processor interrupt signal when the TMS320C17/E17 is being operated in coprocessor mode. The $\overline{\text{EXINT}}$ flag cannot be cleared while an interrupt condition is presented.

The other three interrupts are normally associated with the serial port framing signals. A bit in the system control register (CR9) designates whether FR is to be used for framing or alternatively, $\overline{\text{FSX}}$ and $\overline{\text{FSR}}$. When $\overline{\text{FSX}}$ and $\overline{\text{FSR}}$ control the serial port framing, FR can function as an independent timer interrupt with the timer clocked by the SCLK source. When the serial port is controlled by the internal framing pulse (FR), the $\overline{\text{FSX}}$ and $\overline{\text{FSR}}$ inputs are available as independent edge-triggered interrupts.

Due to the asynchronous operation of the interrupts, the time between the occurrence of an active interrupt signal and the device actually vectoring to ROM location 2 is four CLKOUT cycles; see Appendix A for further timing information.

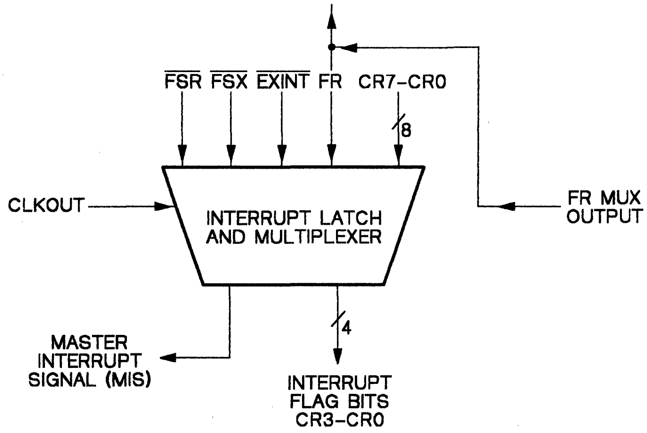

FSR
FSX
EXINT
FR
CR7-CR0
8
CLKOUT
INTERRUPT LATCH AND MULTIPLEXER
FR MUX OUTPUT
4
MASTER INTERRUPT SIGNAL (MIS)
INTERRUPT FLAG BITS CR3-CR0

Figure 3-21. Interrupt Latch and Multiplexer

3.9 Serial Port (TMS320C17/E17)

Two of the I/O ports on the TMS320C17/E17 are dedicated to the serial port and companding hardware. I/O port 0 is dedicated to control register 0, which controls the serial port, interrupts, and companding hardware. I/O port 1 accesses control register 1, as well as both serial port channels, and the companding hardware. The six remaining I/O ports are available for external parallel interfaces.

The on-chip dual-channel serial port, provided on the TMS320C17/E17, is capable of full-duplex serial communications and direct interface to combo-codec PCM systems, serial A/D converters, and other serial systems. The interface signals are directly compatible with codecs and many other serial devices, and require a minimum of external hardware. An example of a codec interface is provided in Section 6.2. For additional information on combo-codecs, refer to the *TCM29C13/C14/C16/C17 Combined Single-Chip PCM Codec and Filter Data Sheet.*

Two receive and two transmit registers are mapped into I/O port 1, and operate with 8-bit data samples. Either internal or external framing signals for serial data transfers (MSB first) are selected via the system control register. The serial port clock, SCLK, provides the bit timing for transfers with the serial port, and may be either an input or output. A framing pulse signal provides framing pulses for combo-codec circuits, a sample clock for voice-band systems, or a timer for control applications. The serial port is accessed through IN and OUT instructions. A block diagram of the serial port and companding hardware is shown in Figure 3-22.

3.9.1 Receive Registers

Two receive registers are mapped into I/O port 1 via the port decode logic. Data is clocked into the shift registers on the next eight negative serial clock (SCLK) transitions after an active framing pulse is detected. SCLK controls the bit-level timing for all serial-port data transfers. Note that the MSB is always shifted first.

On an active framing pulse, serial data is clocked into the receive registers from the DR pins. Channel 0 data is received in shift register RS0 from pin DR0, and channel 1 data is received in shift register RS1 from pin DR1. To read the data from the registers, an IN instruction is executed from port 1. On the first IN instruction after a framing pulse, channel 0 data is output onto the external data bus where it is read by the CPU. On the second IN instruction, channel 1 data is output onto the external data bus.

An active framing pulse initiates the receive operation, as shown in Figure 3-23. External framing pulses ($\overline{\text{FSR}}$) are active low, and the internal framing (FR) signal is active high. With external framing ($\overline{\text{FSR}}$), the falling edge of the framing pulse gates the serial-port clock to the receive shift registers, and the data is clocked into the shift registers on the next eight consecutive negative transitions of the clock.

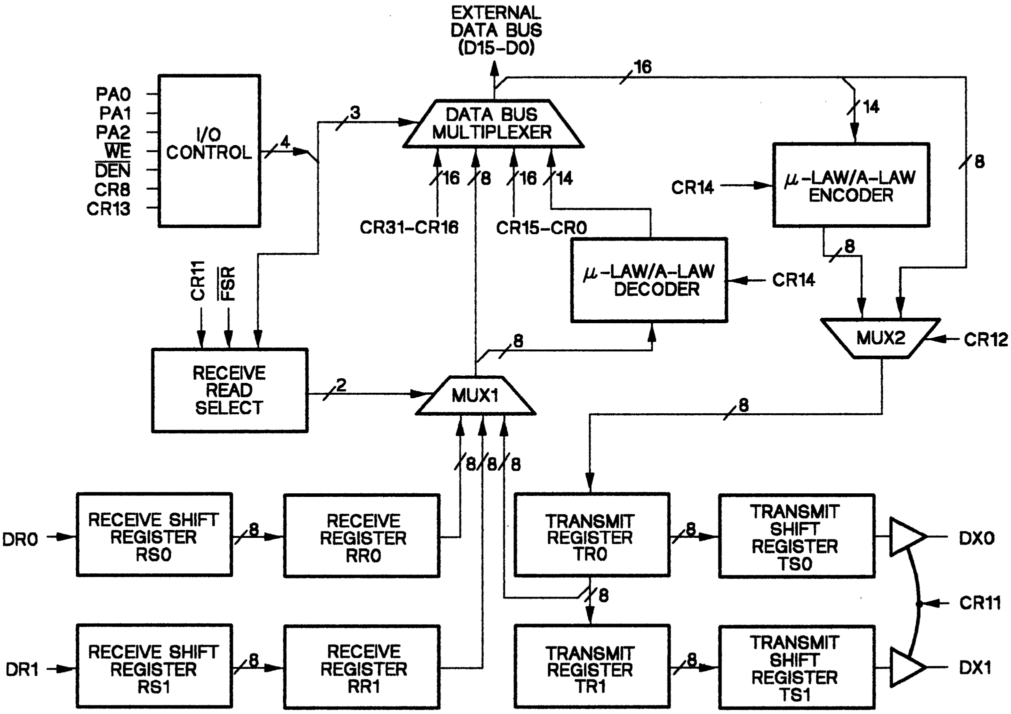


Figure 3-22. Serial Port and Companding Hardware

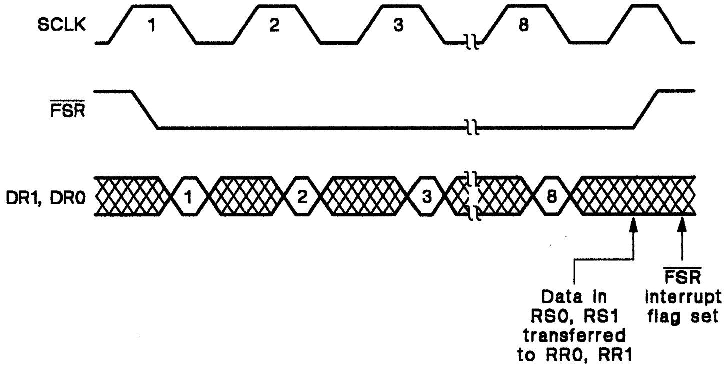


Figure 3-23. Receive Timing for External Framing

The rising edge of the framing pulse transfers the data from the receive shift registers to the receive registers and sets the $\overline{\text{FSR}}$ flag bit (CR1) in the system control register (see Figure 3-23), causing an interrupt to occur if the $\overline{\text{FSR}}$ is enabled. External framing pulses are sensed during the high portion of the SCLK cycle and latched internally with the falling edge of SCLK. Only one $\overline{\text{FSR}}$ state can be detected per SCLK period.

Internal framing (FR) pulses can be selected in either fixed data-rate or variable data-rate modes for combo-codec interface. With the fixed data-rate mode, the FR pulse is one SCLK cycle wide, and appears in the cycle preceding the first data bit. The falling edge of the pulse initiates both the transmit and receive operations, as shown in Figure 3-24. Received data is clocked into the receive shift registers on the next eight consecutive negative transitions of the clock. After data bit 8 has been received, data is transferred from the receive shift registers to the receive registers, and an interrupt is generated when the FR flag bit (CR3) is set in the system control register, thus causing an interrupt to occur if enabled.

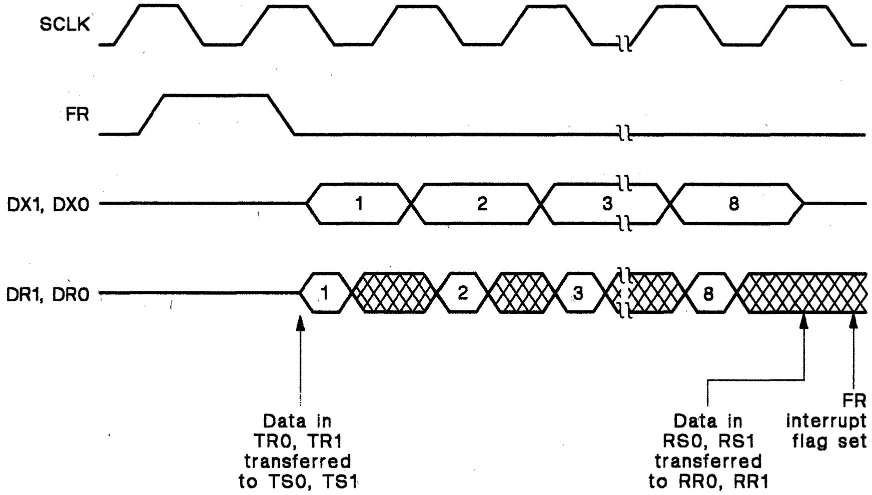


Figure 3-24. Fixed-Data Rate for Internal Framing

In the variable data-rate mode shown in Figure 3-25, the FR pulse is eight SCLK cycles wide, and appears in the same SCLK cycle as the first data bit. The rising edge of the pulse initiates the transmit and receive operations. The falling edge of the pulse transfers data from the receive shift registers to the receive registers and sets the FR flag bit (CR3) in the system control register, causing an interrupt to occur if enabled.

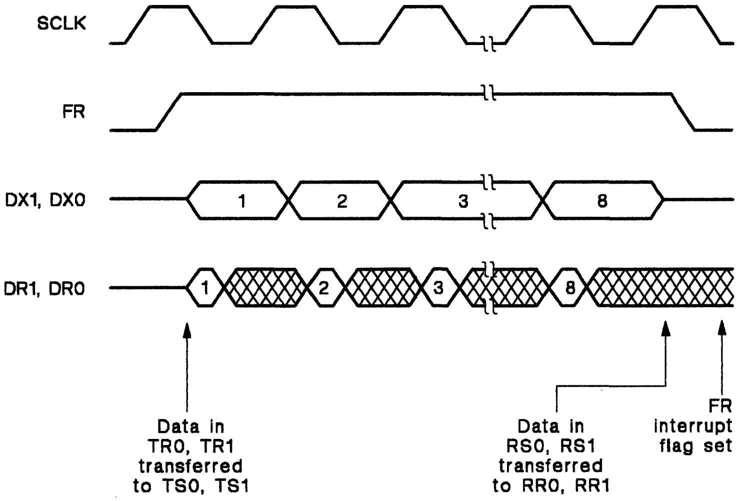

SCLK
FR
DX1, DX0
1
2
3
8
DR1, DR0
1
2
3
8
Data in TR0, TR1 transferred to TS0, TS1
Data in RS0, RS1 transferred to RR0, RR1
FR interrupt flag set

Figure 3-25. Variable-Data Rate for Internal Framing

3.9.2 Transmit Registers

Two transmit registers are mapped into I/O port 1 via the port decode logic. The transmit registers are connected to the port 1 data bus in a FIFO (first in, first out) configuration. On the first OUT instruction to port 1 after a framing pulse, the data to be transmitted is put into transmit register TR0. On the next framing pulse, the TR0 contents are latched into transmit shift register TS0 and the data is transmitted on channel 0 (pin DX0) on the next eight positive transitions of the serial-port clock (SCLK), as shown in Figure 3-26. External framing pulses ($\overline{FSX}$) are active low, and the internal framing (FR) signal is active high. Data sent to port 1 is always put into the transmit registers. Only when control register bit 11 (CR11) is high will the data be enabled onto the transmit pins. The transmit pins are in the high-impedance state when not transmitting. External framing pulses are sensed during the high portion of the SCLK cycle and latched internally with the falling edge of SCLK. Only one $\overline{FSX}$ state can be detected per SCLK period.

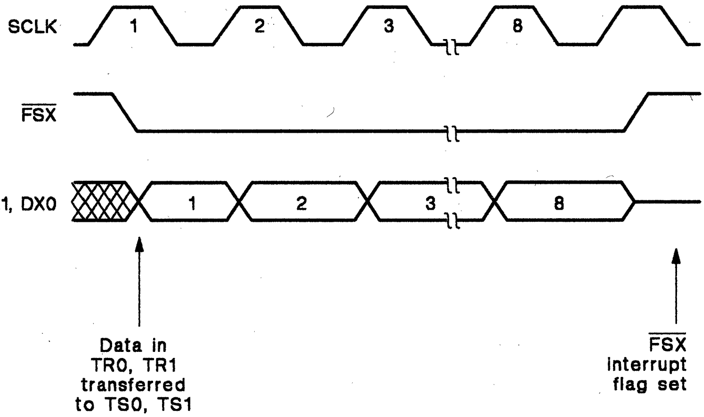

SCLK
1
2
3
8
$\overline{FSX}$
DX1, DX0
1
2
3
8
Data in TR0, TR1 transferred to TS0, TS1
$\overline{FSX}$ Interrupt flag set

Figure 3-26. Transmit Timing for External Framing

Internal framing (FR) pulses can be selected in either fixed data-rate or variable data-rate modes for combo-codec interface. With the fixed data-rate mode, the FR pulse is one SCLK cycle wide, and appears in the cycle preceding the first data bit. The falling edge of the pulse initiates both the transmit and receive operations, as shown in Figure 3-24. Data is transferred from the transmit registers to the transmit shift registers. Transmitted data is clocked into the transmit shift registers on the next eight consecutive negative transitions from the clock. After data bit 8 has been transmitted, an interrupt is generated when the FR flag bit (CR3) is set in the system control register, thus causing an interrupt to occur if enabled.

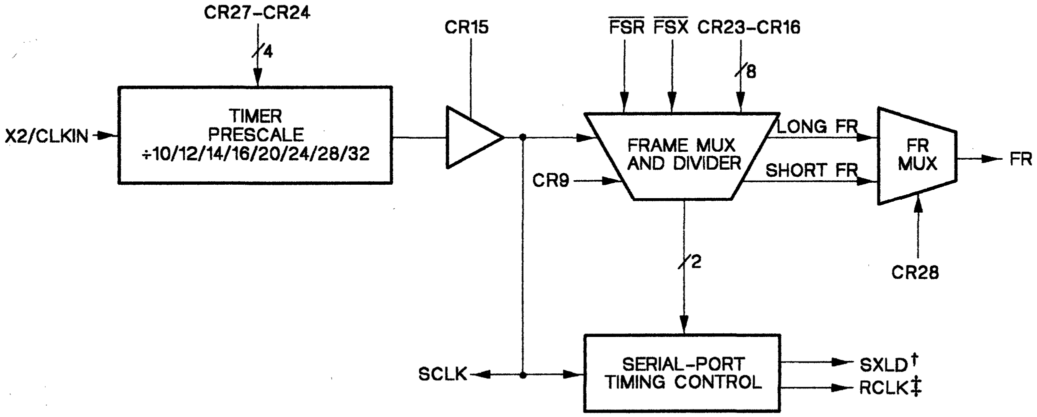
In the variable data-rate mode shown in Figure 3-25, the FR pulse is eight SCLK cycles wide, and appears in the same SCLK cycle as the first data bit. The rising edge of the pulse initiates the transmit and receive operations. The falling edge of the pulse sets the FR flag bit (CR3) in the system control register, causing an interrupt to occur if enabled.

When two OUT instructions to port 1 are executed between framing pulses, both transmit registers are loaded with data for transmission. The first OUT instruction loads data into transmit register TR0. The second OUT pushes the data from TR0 into TR1 and puts the new data into TR0. On an active framing pulse edge, the transmit register contents are latched into the transmit shift registers and the data clocked out on the next eight consecutive positive transitions of SCLK. Thus, for single-channel operation, only one OUT instruction to port 1 should be executed between framing pulses to insure data transmission on channel 0. Only TR0 may be read back to the serial-port data bus by an IN instruction. This feature is used for the parallel companding mode.

Both transmit channels always output data on an active framing pulse when CR11 is high. During single-channel operation (using channel 0), channel 1 still transmits the data from transmit register TR1. Transmit channel 1 cannot be disabled during single-channel operation.

3.9.3 Timing and Framing Control

The serial-port timing and framing control is shown in Figure 3-27. The serial-port clock (SCLK) provides the timing control for data transfers with the serial port. SCLK may be configured as either an input or output through the control register. As an input, SCLK is an external serial system clock that provides the framing synchronization and timing for the serial port. As an output, SCLK provides the system clock for standalone serial applications and is derived from the microcomputer system clock (X2/CLKIN).



†SXLD = Load transmit shift registers (TS0,TS1) from transmit registers (TR0,TR1)
‡RCLK = Load receive registers (RR0,RR1) from receive shift registers (RS0,RS1)

Figure 3-27. Serial-Port Timing and Framing Control

The serial-port clock prescaler determines the divide ratio for SCLK when configured as an output. The TMS320C17/E17 system clock (X2/CLKIN) is input to the prescaler, along with control register bits CR27-CR24. Table 3-4 shows the prescale divide ratios selectable as divide by 10, 12, 14, 16, 20, 24, 28, and 32 through system control register bits CR27-CR24. These divide ratios are available only for SCLK when it is configured as an output from the device (see Section 3.12 for control register bit configurations).

The frame multiplexer determines which framing pulses cause serial-port data transfers to occur and configures the internal framing pulse (FR) frequency. The inputs to the multiplexer are SCLK, control register bit 9 (CR9), control register bits CR23-CR16, external transmit framing ($\overline{\text{FSX}}$) pulse, and external receive framing ($\overline{\text{FSR}}$) pulse. The outputs of the multiplexer go to the serial-port control for receive and transmit timing generation for the serial-port registers and to the FR multiplexer for determining which FR framing pulse will be generated.

The outputs of the frame counter are input to the FR multiplexer for selection of long or short FR pulses. The short FR pulse provides fixed data-rate framing pulses for standalone serial interface to the Texas Instruments TCM29Cxx family of combo-codec circuits. The long FR framing pulse provides variable data-rate framing pulses to the combo-codec.

The FR frequency is determined at the beginning of the framing pulse cycle. The FR frequency is equal to SCLK/(CNT + 2) where CNT is the binary value of CR23-CR16. When reconfiguring the frequency, the upper control register bits determine the new divide ratio. However, the new frequency is not implemented until the next FR framing pulse.

Table 3-4. Serial Clock (SCLK) Divide Ratios (X2/CLKIN = 20.48 MHz)

CR27	CR26	CR25	CR24	DIVIDE RATIO	SCLK FREQUENCY	UNIT
0	0	0	0	32	0.640	MHz
0	0	0	1	28	0.731	MHz
0	0	1	0	24	0.853	MHz
0	1	0	0	20	1.024	MHz
1	0	0	0	16	1.280	MHz
1	0	0	1	14	1.463	MHz
1	0	1	0	12	1.706	MHz
1	1	0	0	10	2.048	MHz

3.10 Companding Hardware (TMS320C17/E17)

The on-chip companding hardware enables the TMS320C17/E17 to compand (COMpress and exPAND) data in either μ-law or A-law format with either sign-magnitude or two's-complement numbers. The standard employed in the United States and Japan is μ-law companding. The European standard is referred to as A-law companding. Configuration and connections of the encoder and decoder (see Figure 3-22) are controlled through the system control register.

When sign magnitude is selected, the μ-law encoding and decoding require a bias adjustment in the sample value. For μ-law encoding, a bias of 33 must be added to the sign magnitude before encoding; likewise, after μ-law decoding, the bias of 33 must be subtracted from the sign-magnitude value. No additional bias adjustment is required for μ-law encoding and decoding when the selected conversion uses two's-complement notation. Note that A-law encoding and decoding do not require a bias adjustment in either case.

Upon reset, the TMS320C17/E17 is programmed to operate in sign-magnitude mode. This mode can be changed by modifying control register bit 29 (CR29). Refer to the *TCM29C13/TCM29C14/TCM29C16/TCM29C17 Combined Single-Chip PCM Codec and Filter Data Sheet* for further information on companding. If software companding is desired without the use of companding hardware, descriptive algorithms are given in the book, *Digital Signal Processing Applications with the TMS320 Family* (literature number SPRA012A); refer to the application report, "Companding Routines for the TMS32010/TMS32020."

The specification for μ-law and A-law log PCM is part of the CCITT G.711 recommendation. Part of the coding format specifies certain bits to be inverted prior to transmission or upon receipt of transmitted data. The companding hardware in the TMS320C17/E17 implements the bit inversion as well as the logarithmic compression and decompression.

Data may be companded via four modes: serial-port encode, serial-port decode, parallel encode, and parallel decode. In the serial mode, transmitted data is encoded according to the specified companding law, and received data is decoded to either sign-magnitude or two's-complement format. In the parallel modes, encoding or decoding is performed on data from the RAM for computations within the device. Note that in parallel mode when two's-complement notation is selected, at least one instruction must be inserted between successive OUT and IN instructions to I/O port 1.

Table 3-5 shows the control register bit combinations that determine the serial or parallel modes of the companding hardware operation. Note that the serial and parallel companding modes require separate control register settings. When using the serial mode, parallel companding is not available unless the control register is reconfigured.

Table 3-5. Serial- and Parallel-Mode Bit Configurations

CR BIT #			MODE OF OPERATION
13	12	11	
0	0	0	Parallel mode. Encoder and decoder are disabled. No operation performed on data written to or read from port 1.
0	0	1	Serial mode. Encoder and decoder are disabled. The transmit registers are enabled for data transmission on an active framing pulse. The 8-bit value written to port 1 is transmitted and the 8-bit value in the receive register is read with an IN instruction from port 1.
0	1	0	Parallel encode. Encoder is enabled. A linear sample written to port 1 with an OUT instruction is compressed to 8-bit log PCM. The 8-bit value is then read from port 1 with an IN instruction.
0	1	1	Serial encode. Encoder is enabled. A linear sample written to port 1 is compressed to 8-bit log PCM and put into the transmit register for transmission on an active framing pulse.
1	0	0	Parallel decode. Decoder is enabled. An 8-bit log PCM data written to port 1 is decoded to linear notation with an IN instruction from port 1.
1	0	1	Serial decode. Decoder is enabled. An 8-bit log PCM sample from one of the receive registers is expanded to linear notation with an IN instruction from port 1.
1	1	0	Parallel encode and decode. Encoder and decoder enabled. In this state, data is compressed on an OUT instruction to port 1 and then expanded with the IN instruction from the port.
1	1	1	Serial encode and decode. Encoder and decoder enabled. Linear data written to port 1 is encoded and put into one of the transmit registers for serial transmission. The 8-bit log PCM data from one of the receive registers is decoded with an IN instruction from port 1.

3.10.1 μ-Law/A-Law Encoder

The encoder compresses linear PCM (14 bits of dynamic range for μ-law format or 13 bits of dynamic range for A-law format) to 8-bit logarithmic PCM. Selection between μ-law or A-law conversion is determined by the system control register bit 14 (CR14). This bit is input directly to the encoder to determine the conversion law to be used. The μ-255 law conversion is performed if CR14 is logic 0, and A-law conversion if CR14 is logic 1. Data is input to the encoder from the data bus with an OUT instruction to port 1. The converted 8-bit log PCM sample is then presented to the multiplexer (MUX2 shown in Figure 3-22). The multiplexer controls whether the encoder output or the eight low-order data bus bits are input to transmit register TR0 of the serial port. Note that the transmit registers are connected to the port 1 data bus in a FIFO (first in, first out) configuration. The encoder compresses data written to port 1 at all times, but the output will be enabled to the TR0 only when CR12 is logic 1.

In the serial-encode mode, data written to port 1 is encoded, and the value put into transmit register TR0. The transmit register is then loaded with the 8-bit value on an active framing pulse, and the 8 bits are clocked out on the positive edge of SCLK.

For the parallel-encode mode, the linear-PCM value is written to port 1 with an OUT instruction. The encoded 8-bit value is then stored in TR0. An IN instruction from port 1 reads TR0 to the data bus for storage in RAM. Care should be taken to have only one OUT and one IN instruction to port 1 for each data sample in the parallel-encode mode. If there are two OUT instructions to port 1, the first sample will be pushed into transmit register TR1, which cannot be read back to the data bus. Note that when two's-complement notation is selected, there must be at least one instruction executed after the OUT instruction to port 1 and before the IN instruction from port 1.

3.10.2 μ-Law/A-Law Decoder

The μ-law/A-law decoder converts 8-bit log-PCM samples to linear PCM. The conversion-law selection is governed by control register bit 14 (CR14). The μ-law conversion is performed if CR14 is logic 0, and A-law conversion if CR14 is logic 1. Data input to the decoder may come from either the serial-port receive registers or transmit register TR0. The multiplexer (MUX1 shown in Figure 3-22) sends data to the data bus either through the decoder or directly to the bus. This multiplexer is controlled in part by control register bit 13 (CR13). If this bit is logic 0, the multiplexer output is sent to the data bus directly. If the bit is logic 1, the multiplexer output is sent to the data bus through the decoder.

In the serial-decode mode, received data from the serial-port receive registers is input to the decoder from the multiplexer, and the received data is decoded according to either μ-law or A-law format.

For the parallel-decode mode, the 8-bit PCM sample to be decoded is written to port 1 with an OUT instruction. This stores the sample in transmit register TR0. The sample is then decoded by reading the value from port 1 with an IN instruction. The IN instruction brings the sample from TR0 through the multiplexer (MUX1) to the decoder, which performs the expansion on the 8-bit sample. Again, there should be only one OUT and one IN instruction to port 1 for each sample to be decoded in order to avoid losing a sample in transmit register TR1. Note that when two's-complement notation is selected, there must be at least one instruction executed after the OUT instruction to port 1 and before the IN instruction from port 1.

3.11 Coprocessor Port (TMS320C17/E17)

The coprocessor port on the TMS320C17/E17 provides a direct interface to most 4/8-bit microcomputers and 16/32-bit microprocessors. The port is accessed through I/O port 5 using IN and OUT instructions. The coprocessor interface allows the device to act as a peripheral (slave) microcomputer to a microprocessor, or as a master to a peripheral microcomputer such as the TMS7042. The coprocessor port is enabled by setting MC/$\overline{\mathrm{PM}}$ and MC low. The microcomputer mode is enabled by setting these two pins high. (Note that the MC/$\overline{\mathrm{PM}}$ and MC pins must be in the same state.) In the microcomputer mode, the 16 data lines are used for the 6 parallel 16-bit I/O ports.

Interprocessor communication through the coprocessor interface (see Figure 3-28) is accomplished asynchronously as in memory-mapped I/O operations. In coprocessor mode, the 16-bit data bus is reconfigured to operate as a 16-bit latched bus interface. Control bit 30 (CR30) in control register 1 is used to configure the coprocessor port to either an 8-bit or a 16-bit length for data transfers. Use of the HI/$\overline{\mathrm{LO}}$ pin allows the full 16-bit data latches to be used even when the 8-bit mode is selected.

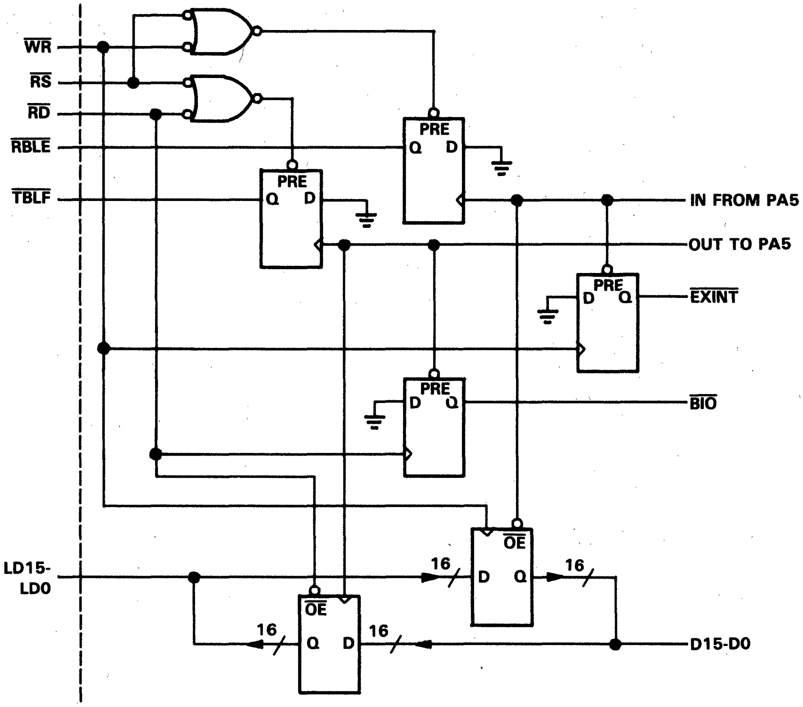


Figure 3-28. TMS320C17/E17 Simplified Coprocessor Port Logic Diagram

Several key characteristics of the coprocessor interface are worth noting and listed below.

- The $\overline{\text{BIO}}$ and $\overline{\text{EXINT}}$ signals are internal to the processor. No inputs should be made on the $\overline{\text{BIO}}$ and $\overline{\text{EXINT}}$ pins.

- Only transfers made when HI/$\overline{\text{LO}}$ is in a low state can activate the internal $\overline{\text{BIO}}$ and $\overline{\text{EXINT}}$ signals.

- The interrupt condition is kept internally until it is cleared by the TMS320C17/E17 reading the data in the coprocessor port latch. The interrupt flag cannot be cleared until the port is read. Four instruction cycles must take place between the read (IN instruction) from the coprocessor port (PA5) and the write to control register CR0 to clear the interrupt flag.

- When the TMS320C17/E17 reads the coprocessor port, it clears the data in the latch.

- The 16 data lines (LD15-LD0) remain in a high-impedance state unless a logic low is asserted on $\overline{\text{RD}}$. When $\overline{\text{RD}}$ is asserted, the TMS320C17/E17 drives the data bus with the data in the coprocessor port latch.

The following sequences of events occur depending upon the configuration and use of the coprocessor port:

- 16-bit data interface (CR30 = 1):
  - All 16 bits of the data port are available for 16-bit transfers to 16/32-bit microprocessors.
  - The HI/$\overline{\text{LO}}$ pin is maintained at a logic low level for all transfers.
  - Transfers to the TMS320C17/E17 (see Figure 3-29):
    1) The $\overline{\text{WR}}$ signal is driven low by the microprocessor.
    2) The $\overline{\text{RBLE}}$ (receive buffer latch empty) signal transitions to a logic high level in response to $\overline{\text{WR}}$.
    3) Data is written from LD15-LD0 to the receive buffer latch when the $\overline{\text{WR}}$ signal is driven high by the microprocessor.
    4) The internal $\overline{\text{EXINT}}$ signal is generated, causing the interrupt flag to be set in the TMS320C17/E17.
    5) The TMS320C17/E17 responds to the interrupt condition and reads port 5 using an IN instruction.
    6) The receive buffer is cleared. (Subsequent reads by the TMS320C17/E17 will be zero value.)
    7) The $\overline{\text{RBLE}}$ signal transitions to a logic low level, signaling the microprocessor that the receive buffer is empty.
    8) The internal $\overline{\text{EXINT}}$ signal is removed, allowing the interrupt flag to be cleared.
    9) The interrupt flag is cleared by writing to control register 0.

- Transfers from the TMS320C17/E17 (see Figure 3-30):
  1) The $\overline{\text{RD}}$ signal is driven low by the microprocessor.
  2) The $\overline{\text{TBLF}}$ (transmit buffer latch full) signal transitions to a logic high level in response to $\overline{\text{RD}}$.
  3) Data is driven from the transmit buffer latch to LD15-LD0 until the $\overline{\text{RD}}$ signal is driven high by the microprocessor.
  4) The internal $\overline{\text{BIO}}$ signal transitions to a logic low level, indicating to the TMS320C17/E17 that the transmit buffer is empty.
  5) The TMS320C17/E17 responds to the $\overline{\text{BIO}}$ condition and writes to port 5 using an OUT instruction.
  6) The $\overline{\text{TBLF}}$ signal transitions to a logic low level, signaling the microprocessor that the transmit buffer is full.
  7) The internal $\overline{\text{BIO}}$ signal transitions back to a logic high state.

- 8-bit data interface (CR30 = 0):
  - Only the least-significant eight bits of the data port are available for 8-bit transfers to 4/8-bit microcomputers.
  - Eight-bit microcomputers may complete full 16-bit transfers by first transferring data with the HI/$\overline{\text{LO}}$ signal in a logic high state (steps 1 through 4 below), and then with HI/$\overline{\text{LO}}$ in a logic low state. Composing 16-bit data in this manner requires two external bus cycles but only one internal port access. The HI/$\overline{\text{LO}}$ pin may be maintained at a logic low level if only 8-bit transfers are desired.
  - Transfers to the TMS320C17/E17 (see Figure 3-29):
    1) The HI/$\overline{\text{LO}}$ signal is driven high by the microcomputer to allow transfers to the upper eight bits of the internal latch.
    2) The $\overline{\text{WR}}$ signal is driven low by the microcomputer.
    3) Data is written from LD7-LD0 to the receive buffer latch (D15-D8) when the $\overline{\text{WR}}$ signal is driven high by the microcomputer.
    4) The HI/$\overline{\text{LO}}$ signal is driven low by the microcomputer to allow transfers to the lower eight bits of the internal latch.
    5) The $\overline{\text{WR}}$ signal is driven low by the microcomputer.
    6) The $\overline{\text{RBLE}}$ (receive buffer latch empty) signal transitions to a logic high level.
    7) Data is written from LD7-LD0 to the receive buffer latch (D7-D0) when the $\overline{\text{WR}}$ signal is driven high by the microcomputer.
    8) The internal $\overline{\text{EXINT}}$ signal is generated, causing the interrupt flag to be set in the TMS320C17/E17.
    9) The TMS320C17/E17 responds to the interrupt condition and reads port 5 using an IN instruction.

10) The receive buffer is cleared. (Subsequent reads by the TMS320C17/E17 will be zero value.)

11) The $\overline{\text{RBLE}}$ signal transitions to a logic low level, signaling the microcomputer that the receive buffer is empty.

12) The internal $\overline{\text{EXINT}}$ signal is removed, allowing the interrupt flag to be cleared.

13) The interrupt flag is cleared by writing to control register 0.

- Transfers from the TMS320C17/E17 (see Figure 3-30):

  1) The HI/$\overline{\text{LO}}$ signal is driven high by the microcomputer to allow transfers from the upper eight bits of the internal latch.

  2) The $\overline{\text{RD}}$ signal is driven low by the microcomputer.

  3) Data is driven from the transmit buffer latch (D15-D8) to LD7-LD0 until the $\overline{\text{RD}}$ signal is driven high by the microcomputer.

  4) The HI/$\overline{\text{LO}}$ signal is driven low by the microcomputer to allow transfers from the lower eight bits of the internal latch.

  5) The $\overline{\text{RD}}$ signal is driven low by the microcomputer.

  6) The $\overline{\text{TBLF}}$ (transmit buffer latch full) signal transitions to a logic high level.

  7) Data is driven from the transmit buffer latch (D7-D0) to LD7-LD0 until the $\overline{\text{RD}}$ signal is driven high by the microcomputer.

  8) The internal $\overline{\text{BIO}}$ signal transitions to a logic low level, indicating to the TMS320C17/E17 that the transmit buffer is empty.

  9) The TMS320C17/E17 responds to the $\overline{\text{BIO}}$ condition and writes to port 5 using an OUT instruction.

  10) The $\overline{\text{TBLF}}$ signal transitions to a logic low level, signaling the microcomputer that the transmit buffer is full.

  11) The internal $\overline{\text{BIO}}$ signal transitions back to a logic high state.

Examples of the use of a coprocessor interface are provided in Section 6.5 and the data sheet of Appendix A.

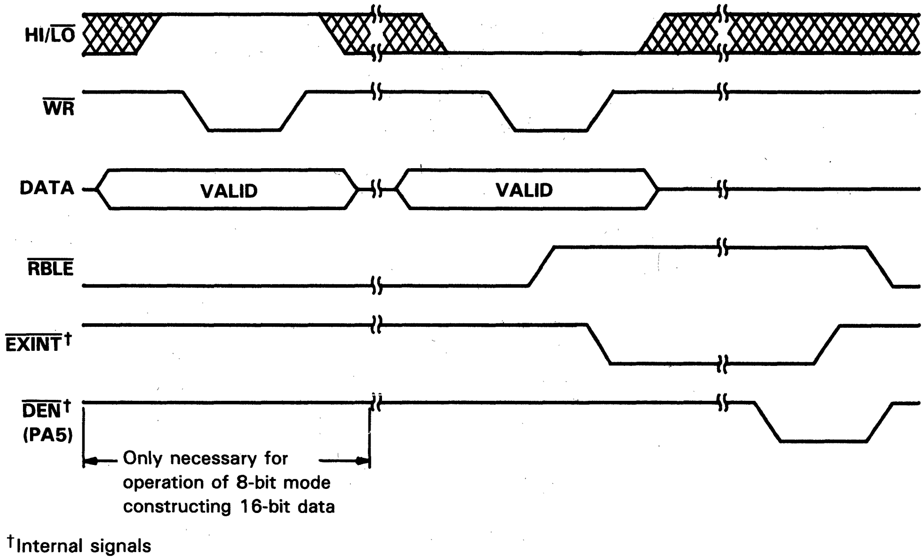


Figure 3-29. External Write Timing to the Coprocessor Port

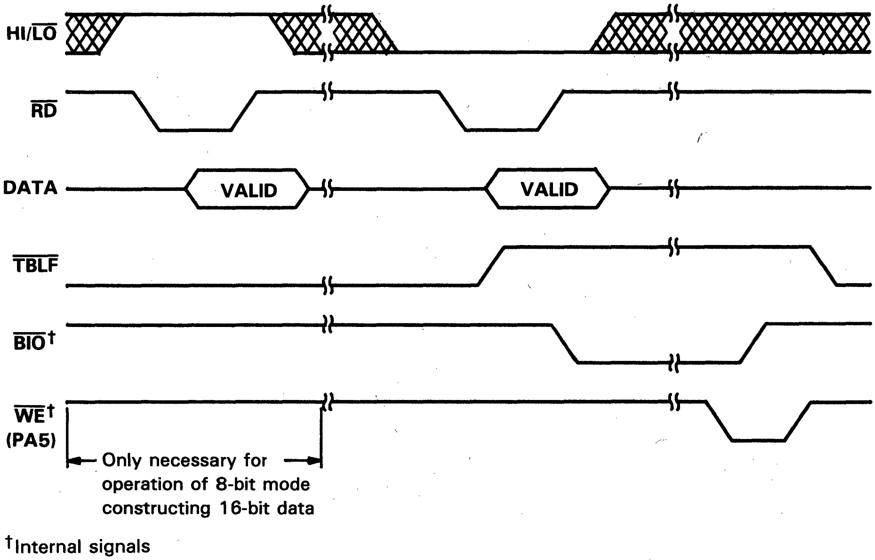


Figure 3-30. External Read Timing from the Coprocessor Port

3.12 System Control Register (TMS320C17/E17)

The TMS320C17/E17 provides additional hardware for interfacing ease in serial applications. This hardware is interfaced to the microcomputer portion of the device via the external data bus (D15-D0). The additional hardware is controlled by a 32-bit system control register (see Figure 3-31), thereby eliminating any additions to the TMS320 instruction set.

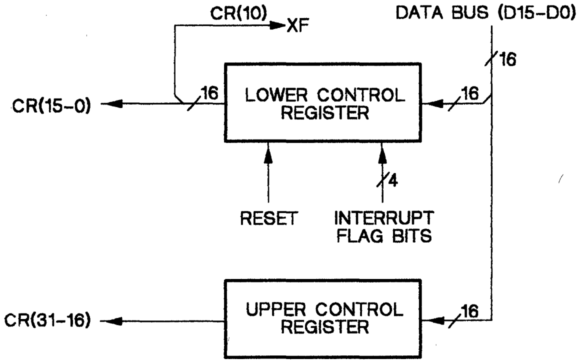


Figure 3-31. System Control Register

The lower 16 register bits (CR15-CR0) are accessed through port 0. These bits control interrupts, serial-port configuration, the external logic output flag, internal and external framing pulses, and the μ-law/A-law encoder and decoder. The interrupt inputs (EXINT, FSX, FSR, and FR) are synchronized to CLKOUT and control the interrupt flag bits (CR3-CR0). The interrupts are maskable via the interrupt enable bits (CR7-CR4). Bit 8 (CR8) controls I/O port 1 configuration.

The upper 16 bits (CR31-CR16) are accessed through port 1. These bits control the internal framing pulse (FR) output frequency, serial-clock divide ratios, pulse-width control for the FR framing pulse, and companding conversions. The bit width of the coprocessor mode is controlled by CR30.

The external data bus provides on-chip communication with the system control register, serial port, companding hardware, and coprocessor port. With a write to port 0, the lower control register is addressed and data latched into the register by the rising edge of the write enable (WE) signal. To write to the upper control register bits, bit 8 of the lower control register must be set to logic 1. If CR8 is logic 0, a write to port 1 accesses the serial port and companding hardware.

Table 3-6 gives a detailed description of the control register bits and their operation. The control register bits are configured through OUT instructions to port 0 and port 1. WE goes low during the first cycle of the OUT instruction, enabling the port data onto the external data bus. The control register bits are latched on the rising edge of WE. There is a propagation delay time for these bits to access the appropriate hardware (see Appendix A for timing information). An allowance for this write delay should be made when reconfiguring (writing to) the control register. The most critical factor is receiving an external framing pulse while reconfiguring the control register. If an external framing pulse is received at that time, it may not be detected and the serial-port registers will contain random data (see Section 3.9 for further details).

Table 3-6. Control Register Bit Definitions

CR BIT #	DESCRIPTION
3-0	Interrupt flags. When an interrupt occurs on any of the four maskable interrupts, the appropriate flag is set to logic 1 whether the interrupt is enabled or disabled. To clear the flag, a logic 1 is written to the appropriate bit by an OUT instruction to port 0. The bits may be read by an IN instruction to determine interrupt sources when multiple interrupts are enabled. Bit # Flag 0 $\overline{\text{EXINT}}$ 1 $\overline{\text{FSR}}$ 2 $\overline{\text{FSX}}$ 3 FR
7-4	Interrupt enable bits. When one of these bits is set to logic 1, an interrupt occurring on that input sets the appropriate flag and activates the microcomputer interrupt circuitry. When disabled, the interrupt flag is still set, but the device is not interrupted. Bit # Flag 4 $\overline{\text{EXINT}}$ 5 $\overline{\text{FSR}}$ 6 $\overline{\text{FSX}}$ 7 FR
8	Port 1 control bit. When set to logic 0, I/O port 1 is connected to either the serial-port registers or the companding hardware, depending on the state of CR11. When set to logic 1, I/O port 1 is connected to the upper control register. This bit must be set with an OUT instruction to port 0 before port 1 may access the upper control register bits CR31-CR16.
9	External framing enable. This bit controls which framing pulses cause serial port data transmission to occur. When set to logic 0, serial port transmit and receive operations occur simultaneously and are controlled by the internal framing (FR) pulse. When set to logic 1, transmit operations are controlled by the external transmit framing ($\overline{\text{FSX}}$) pulse, and receive operations are controlled by the external receive framing ($\overline{\text{FSR}}$) pulse.
10	XF output latch. This bit controls the logic level of the external logic output flag (XF) pin. A write delay time occurs when reconfiguring this latch (see Appendix A for timing information).
11	Serial port enable. When set to logic 0, the transmit and receive registers are disabled in order to use the parallel companding mode. When set to logic 1, the serial port registers are enabled and data transfers with the serial port are via OUT and IN instructions to port 1. A reset sets this bit to zero.

Table 3-6. Control Register Bit Definitions (Concluded)

CR BIT #	DESCRIPTION
12	µ-law/A-law encoder enable. When set to logic 0, the encoder is disabled. When set to logic 1, the encoder is enabled, and data written to port 1 is µ-law or A-law encoded. The encoder must be enabled for compression of linear data in both the serial and parallel modes of operation.
13	µ-law/A-law decoder enable. When set to logic 0, the decoder is disabled. When set to logic 1, the decoder is enabled, and data read from port 1 is µ-law or A-law decoded to linear format. The decoder must be enabled for expansion of log PCM data in both the serial and parallel modes of operation.
14	µ-law or A-law encode/decode select. When set to logic 0, the companding hardware performs µ-255-law conversion. When set to logic 1, the companding hardware performs A-law conversion.
15	Serial clock control. When set to logic 0, the serial port clock (SCLK) is an output, and its frequency is derived from the microcomputer system clock, X2/CLKIN. When set to logic 1, SCLK is an input that provides the clock for all data transfers with the serial port and the frame counter in timing logic. A reset sets this bit to one.
23-16	Frame counter modulus. The value of these bits determines the divide ratio for the FR output frequency. The FR frequency is given as SCLK/(CNT + 2) where CNT is a binary value of CR23-CR16. The following should be noted when configuring the divide ratio: 1. CNT must be in the range given by 7 ≤ CNT ≤ 254. 2. Bits are operational whether SCLK is an input or an output.
27-24	SCLK prescale control bits. As an output, SCLK is derived from the microcomputer system clock, X2/CLKIN. Prescale divide ratios are selectable through these control bits (see Section 3.9.3 for the available divide ratios).
28	FR pulse-width control. This bit controls the pulse width of the FR output to select data-transfer rates with combo-codec circuits. When set to logic 0, the FR output framing pulse is one SCLK cycle wide for the fixed data-rate mode and appears in the serial-clock cycle preceding the first serial-bit transmission. When set to logic 1, the FR output framing pulse is eight SCLK cycles wide for the variable data-rate mode. In this mode, the framing pulse is active high for the duration of the eight bits transmitted and received.
29	Two's-complement µ-law/A-law conversion enable. When set to logic 0, sign-magnitude companding is enabled. When set to logic 1, two's-complement companding is enabled. When two's-complement companding has been selected along with the parallel companding mode of operation, one instruction must be inserted between successive OUT and IN instructions to port 1. A reset sets this bit to zero.
30	8/16-bit length coprocessor mode select. When set to logic 0, the 8-bit byte length is used. When set to logic 1, the 16-bit word length is selected.
31	Reserved for future expansion. This bit should be set to zero.

Section 4

Assembly Language Instructions

The instruction set of the TMS320C1x (first-generation TMS320) processors supports numeric-intensive signal processing operations and general-purpose applications, such as high-speed control. The instruction set shown in Table 4-2 consists primarily of single-cycle, single-word instructions, permitting execution rates of up to 6.25 million instructions per second. Only infrequently used branch and I/O instructions are multi-cycle.

For operations involving multiplication, the TMS320C1x instruction set offers a single-cycle instruction (MPY). For ease of use in a Harvard architecture, table read (TBLR) and table write (TBLW) instructions are provided, which allow information transfer between data and program memory. The IN and OUT instructions permit a data word to be read into the on-chip RAM in only two cycles. The SUBC (conditional subtract) instruction performs the shifting and conditional branching necessary to implement a divide efficiently and quickly.

This section describes the TMS320C1x assembly language instructions. Included in this section are the following major topics:

- Memory Addressing Modes (Section 4.1 on page 4-2)
  Direct addressing
  Indirect addressing (using two auxiliary registers)
  Immediate addressing

- Instruction Set (Section 4.2 on page 4-7)
  Symbols and abbreviations used in the instructions
  Instruction set summary (listed according to function)

- Individual Instruction Descriptions (Section 4.3 on page 4-11)
  Presented in alphabetical order and providing the following:
  - Assembler syntax
  - Operands
  - Execution
  - Encoding
  - Description
  - Words
  - Cycles
  - Example(s)

4.1 Memory Addressing Modes

The TMS320C1x instruction set provides three memory addressing modes:

- Direct addressing mode
- Indirect addressing mode
- Immediate addressing mode.

Both direct and indirect addressing can be used to access data memory. Direct addressing concatenates seven bits of the instruction word with the 1-bit data memory page pointer to form the 8-bit data memory address. Indirect addressing accesses data memory through the two auxiliary registers. In immediate addressing, the data is based on a portion of the instruction word(s). The following sections describe each addressing mode and give the opcode formats and some examples for each mode.

4.1.1 Direct Addressing Mode

In the direct memory addressing mode, the instruction word contains the lower seven bits of the data memory address (dma). This field is concatenated with the one-bit data memory page pointer (DP) register to form the full 8-bit data memory address. This implements a paging scheme in which the first page contains 128 words and the second page contains 16/128 words. In a typical application, infrequently accessed system variables, such as those used when performing an interrupt routine, are stored on the second page. The 7-bit address in the instruction points to the specific location within that data memory page. The DP register is loaded through the LDP (load data memory page pointer), LDPK (load data memory page pointer immediate), or LST (load status bits from data memory) instructions. The data page pointer is part of the status register and thus can be stored in data memory.

Note:

The data page pointer is not initialized by reset and is therefore undefined after powerup. The TMS320C1x development tools, however, utilize default values for many parameters, including the data page pointer. Because of this, programs that do not explicitly initialize the data page pointer may execute improperly depending on whether they are executed on a TMS320C1x device or using a development tool. Thus, it is critical that all programs initialize the data page pointer in software.

Figure 4-1 illustrates how the 8-bit data address is formed.

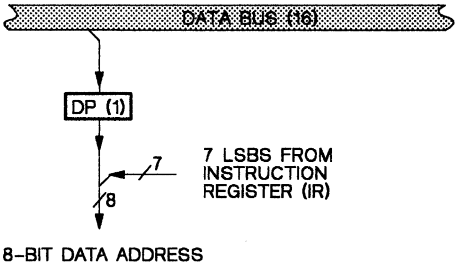


Figure 4-1. Direct Addressing Block Diagram

Direct addressing can be used with all instructions except CALL, the branch instructions, immediate operand instructions, and instructions with no operands. The direct addressing format is as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Opcode								0	dma						

Bits 15 through 8 contain the opcode. Bit 7 = 0 defines the addressing mode as direct. Bits 6 through 0 contain the data memory address (dma), which can directly address up to 128 words (1 page) of data memory. Use of the data memory page pointer is required to address the full data memory space.

Example of Direct Addressing Format:

ADD 9,5 Add to accumulator the contents of data memory location 9 left-shifted 5 bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1

The opcode of the ADD 9,5 instruction is 05h and appears in bits 15 through 8. The notation nnh indicates nn is a hexadecimal number. The shift count of 5h appears in bits 11 through 8 of the opcode. The data memory address 09h appears in bits 6 through 0.

4.1.2 Indirect Addressing Mode

Indirect addressing forms the data memory address from the least significant eight bits of one of the two auxiliary registers, AR0 and AR1. This is sufficient to address all the data memory; no paging is necessary with indirect addressing. The Auxiliary Register Pointer (ARP) selects the current auxiliary register. The auxiliary registers can be automatically incremented or decremented in parallel with the execution of any indirect instruction to permit single-cycle manipulation of data tables. The increment/decrement occurs AFTER the current instruction has completed executing.

In indirect addressing, the 8-bit addresses contained in the auxiliary registers may be loaded by the instructions LAR (load auxiliary register) and LARK (load auxiliary register immediate). The auxiliary registers may be modified by the MAR (modify auxiliary register) instruction or, equivalently, by the indirect addressing field of any instruction supporting indirect addressing. AR(ARP) denotes the auxiliary register selected by ARP.

The following symbols are used in indirect addressing:

*	Contents of AR(ARP) are used for data memory address.
*-	Contents of AR(ARP) are used for address, then decremented after data memory access.
*+	Contents of AR(ARP) are used for address, then incremented after data memory access.

The indirect addressing format is as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Opcode								1	0	INC	DEC	NAR	0	0	ARP

NOTE: NAR = new auxiliary register control bit.

Bits 15 through 8 contain the opcode, and bit 7 = 1 defines the addressing mode as indirect. Bits 6 through 0 contain the indirect addressing control bits.

Bit 3 and bit 0 control the Auxiliary Register Pointer (ARP). If bit 3 = 0, the contents of bit 0 are loaded into the ARP after execution of the current instruction. If bit 3 = 1, the contents of the ARP remain unchanged. ARP = 0 defines the contents of AR0 as a memory address. ARP = 1 defines the contents of AR1 as a memory address. Note that NAR denotes the new auxiliary register control bit.

Bit 5 and bit 4 control the auxiliary registers. If bit 5 = 1, the current auxiliary register is incremented by 1 after execution. If bit 4 = 1, the current auxiliary register is decremented by 1 after execution. If bit 5 and bit 4 are 0, then neither auxiliary register is incremented nor decremented. Bits 6, 2, and 1 are reserved and should always be programmed to 0.

The auxiliary registers may also be used for temporary storage via the load and store auxiliary register instructions. LAR and SAR, respectively.

The examples that follow illustrate the indirect addressing format. Indirect addressing is indicated by an asterisk (*) in these examples and in the TMS320C1x assembler.

Example 1:

ADD *+,8 Add to the accumulator the contents of the data memory address defined by the contents of the current auxiliary register. This data is left-shifted 8 bits before being added. The current auxiliary register is autoincremented by one. The opcode is 08A8h, as shown below.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0

Example 2:

ADD *,8 As in Example 1, but with no autoincrement; the opcode is 0888h.

Example 3:

ADD *-,8 As in Example 1, except that the current auxiliary register is decremented by 1; the opcode is 0898h.

Example 4:

ADD *+,8,1 As in Example 1, except that the auxiliary register pointer is loaded with the value 1 after execution; the opcode is 08A1h.

Example 5:

ADD *+,8,0 As in Example 4, except that the auxiliary register pointer is loaded with the value 0 after execution; the opcode is 08A0h.

4.1.3 Immediate Addressing Mode

Included in the TMS320C1x instruction set are five immediate operand instructions, in which the immediate operand is contained within the instruction word. These instructions execute within a single instruction cycle. The length of the constant operand is instruction-dependent. The immediate instructions are:

LACK	Load accumulator immediate short (8-bit constant)
LARK	Load auxiliary register immediate short (8-bit constant)
LARP	Load auxiliary register pointer (1-bit constant)
LDPK	Load data memory page pointer immediate (1-bit constant)
MPYK	Multiply immediate (13-bit constant)

The following examples illustrate immediate addressing format:

Example 1:

MPYK 2781 Multiply the value 2781 with the contents of the T register. The result is loaded into the P register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	13-bit constant												

Example 2:

LACK 221 Load the constant 221 in the lower eight bits of the accumulator right-justified. The upper 24 bits of the accumulator are zero.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	0	8-bit constant							

4.2 Instruction Set

The following sections list the symbols and abbreviations used in the TMS320C1x instruction set summary and in the instruction descriptions. The complete instruction set summary is organized according to function. A detailed description of each instruction is listed in the instruction set summary.

4.2.1 Symbols and Abbreviations

Table 4-1 lists symbols and abbreviations used in the instruction set summary (Table 4-2) and the individual instruction descriptions.

Table 4-1. Instruction Symbols

SYMBOL	MEANING
A	Port address
ACC	Accumulator
ARn	Auxiliary Register n (AR0 and AR1) are predefined assembler symbols equal to 0 and 1, respectively.)
ARP	Auxiliary register pointer
B	Branch address
D	Data memory address field
DATn	Label assigned to data memory location n
dma	Data memory address
DP	Data page pointer
I	Addressing mode bit
INTM	Interrupt mode bit
K	Immediate operand field
nnh	Indicates nn is a hexadecimal number. (All others are assumed to be decimal values.)
OVM	Overflow (saturation) mode flag bit
P	Product register
PA	Port address (PA0 through PA7 are predefined assembler symbols equal to 0 through 7, respectively.)
PC	Program counter
pma	Program memory address
PRGn	Label assigned to program memory location n
R	1-bit operand field specifying auxiliary register
S	4-bit left-shift code
T	Temporary register
TOS	Top of stack
X	3-bit accumulator left-shift field
→	Is assigned to
| |	An absolute value
< >	User-defined items
[ ]	Optional items
( )	"Contents of"
{ }	Alternative items, one of which must be entered
< >	Angle brackets back-to-back indicate "not equal".
	Blanks or spaces must be entered where shown.

4.2.2 Instruction Set Summary

Table 4-2 provides the TMS320C1x instruction set summary, arranged according to function and alphabetized within each functional grouping. Additional information is presented in the individual instruction descriptions in the following section.

The instruction set summary consists primarily of single-cycle, single-word instructions. Only infrequently used branch and I/O instructions are multicycle.

Table 4-2. Instruction Set Summary

ACCUMULATOR MEMORY REFERENCE INSTRUCTIONS				
Mnemonic and Description		Cycles	Words	16-Bit Opcode MSB LSB
ABS	Absolute value of accumulator	1	1	0111 1111 1000 1000
ADD	Add to accumulator with shift	1	1	0000 SSSS I DDD DDDD
ADDH	Add to high accumulator	1	1	0110 0000 I DDD DDDD
ADDS	Add to low accumulator with sign-extension suppressed	1	1	0110 0001 I DDD DDDD
AND	AND with accumulator	1	1	0111 1001 I DDD DDDD
LAC	Load accumulator with shift	1	1	0010 SSSS I DDD DDDD
LACK	Load accumulator immediate short	1	1	0111 1110 KKKK KKKK
OR	OR with accumulator	1	1	0111 1010 I DDD DDDD
SACH	Store high accumulator with shift	1	1	0101 1XXX I DDD DDDD
SACL	Store low accumulator	1	1	0101 0000 I DDD DDDD
SUB	Subtract from accumulator with shift	1	1	0001 SSSS I DDD DDDD
SUBC	Conditional subtract	1	1	0110 0100 I DDD DDDD
SUBH	Subtract from high accumulator	1	1	0110 0010 I DDD DDDD
SUBS	Subtract from low accumulator with sign-extension suppressed	1	1	0110 0011 I DDD DDDD
XOR	Exclusive-OR with low accumulator	1	1	0111 1000 I DDD DDDD
ZAC	Zero accumulator	1	1	0111 1111 1000 1001
ZALH	Zero low accumulator and load high accumulator	1	1	0110 0101 I DDD DDDD
ZALS	Zero accumulator and load low accumulator with sign-extension suppressed	1	1	0110 0110 I DDD DDDD
AUXILIARY REGISTER AND DATA PAGE POINTER INSTRUCTIONS				
Mnemonic and Description		Cycles	Words	16-Bit Opcode MSB LSB
LAR	Load auxiliary register	1	1	0011 100R I DDD DDDD
LARK	Load auxiliary register immediate short	1	1	0111 000R KKKK KKKK
LARP	Load auxiliary register pointer immediate	1	1	0110 1000 1000 000K
LDP	Load data memory page pointer	1	1	0110 1111 I DDD DDDD
LDPK	Load data memory page pointer immediate	1	1	0110 1110 0000 000K
MAR	Modify auxiliary register	1	1	0110 1000 I DDD DDDD
SAR	Store auxiliary register	1	1	0011 000R I DDD DDDD
T REGISTER, P REGISTER, AND MULTIPLY INSTRUCTIONS				
Mnemonic and Description		Cycles	Words	16-Bit Opcode MSB LSB
APAC	Add P register to accumulator	1	1	0111 1111 1000 1111
LT	Load T register	1	1	0110 1010 I DDD DDDD
LTA	Load T register and accumulate previous product	1	1	0110 1100 I DDD DDDD
LTD	Load T register, accumulate previous product, and move data	1	1	0110 1011 I DDD DDDD
MPY	Multiply (with T register, store product in P register)	1	1	0110 1101 I DDD DDDD
MPYK	Multiply immediate	1	1	100K KKKK KKKK KKKK
PAC	Load accumulator with P register	1	1	0111 1111 1000 1110
SPAC	Subtract P register from accumulator	1	1	0111 1111 1001 0000

Table 4-2. Instruction Set Summary (Concluded)

BRANCH/CALL INSTRUCTIONS				
Mnemonic and Description		Cycles	Words	16-Bit Opcode MSB LSB
B	Branch unconditionally	2	2	1111 1001 0000 0000 0000 BBBB BBBB BBBB
BANZ	Branch on auxiliary register not zero	2	2	1111 0100 0000 0000 0000 BBBB BBBB BBBB
BGEZ	Branch if accumulator ≥ 0	2	2	1111 1101 0000 0000 0000 BBBB BBBB BBBB
BGZ	Branch if accumulator > 0	2	2	1111 1100 0000 0000 0000 BBBB BBBB BBBB
BIOZ	Branch on I/O status = 0	2	2	1111 0110 0000 0000 0000 BBBB BBBB BBBB
BLEZ	Branch if accumulator ≤ 0	2	2	1111 1011 0000 0000 0000 BBBB BBBB BBBB
BLZ	Branch if accumulator < 0	2	2	1111 1010 0000 0000 0000 BBBB BBBB BBBB
BNZ	Branch if accumulator ≠ 0	2	2	1111 1110 0000 0000 0000 BBBB BBBB BBBB
BV	Branch on overflow	2	2	1111 0101 0000 0000 0000 BBBB BBBB BBBB
BZ	Branch if accumulator = 0	2	2	1111 1111 0000 0000 0000 BBBB BBBB BBBB
CALA	Call subroutine indirect	2	1	0111 1111 1000 1100
CALL	Call subroutine	2	2	1111 1000 0000 0000 0000 BBBB BBBB BBBB
RET	Return from subroutine	2	1	0111 1111 1000 1101

CONTROL INSTRUCTIONS				
Mnemonic and Description		Cycles	Words	16-Bit Opcode MSB LSB
DINT	Disable interrupt	1	1	0111 1111 1000 0001
EINT	Enable interrupt	1	1	0111 1111 1000 0010
LST	Load status register from data memory	1	1	0111 1011 I DDD DDDD
NOP	No operation	1	1	0111 1111 1000 0000
POP	Pop top of stack to low accumulator	2	1	0111 1111 1001 1101
PUSH	Push low accumulator onto stack	2	1	0111 1111 1001 1100
ROVM	Reset overflow mode	1	1	0111 1111 1000 1010
SOVM	Set overflow mode	1	1	0111 1111 1000 1011
SST	Store status register	1	1	0111 1100 I DDD DDDD

I/O AND DATA MEMORY OPERATIONS				
Mnemonic and Description		Cycles	Words	16-Bit Opcode MSB LSB
DMOV	Data move in data memory	1	1	0110 1001 I DDD DDDD
IN	Input data from port	2	1	0100 0AAA I DDD DDDD
OUT	Output data to port	2	1	0100 1AAA I DDD DDDD
TBLR	Table read	3	1	0110 0111 I DDD DDDD
TBLW	Table write	3	1	0111 1101 I DDD DDDD

4.3 Individual Instruction Descriptions

Each instruction in the instruction set summary is described in the following pages. Instructions are listed in alphabetical order. Information, such as assembler syntax, operands, execution, encoding, description, words, cycles, and examples, is provided for each instruction. An example instruction is provided on the next two pages to familiarize the user with the special format used and explain its content. Refer to Section 4.1 for further information on memory addressing. Code examples using many of the instructions are given in Section 5 on Software Applications.

EXAMPLE — Example Instruction

Syntax

Direct: [<label>] EXAMPLE <dma>[,<shift>]
Indirect: [<label>] EXAMPLE {*|*+|*-}[,<shift>[,<next ARP>]]
Immediate: [<label>] EXAMPLE [<constant>]

Each instruction begins with an assembler syntax expression. The optional comment field that concludes the syntax is not included in the syntax expression. Space(s) are required between each field (label, command, operand, and comment fields) as shown in the syntax. The syntax example illustrates both direct and indirect addressing, as well as immediate addressing in which the operand field includes <constant>.

Operands

$0 \leq \text{dma} \leq 127$
ARP = 0 or 1
$0 \leq \text{constant} \leq 255$

Operands may be constants or assembly-time expressions which refer to memory, I/O and register addresses, pointers, shift counts, and a variety of constants. The operand values used in the example syntax are shown.

Execution

(PC) + 1 → PC
(ACC) + (dma) × 2^{shift} → ACC

1 → interrupt mode (INTM) status bit
Affects INTM.

This section provides an example of the instruction operation sequence, describing the processing that takes place when the instruction is executed. Conditional effects of status register specified modes are also given. In addition, those bits in the status registers that are affected by the instruction are listed.

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	0	0	0	Shift				0	Data Memory Address						
Indirect:	0	0	0	0	Shift				1	See Section 4.1						
Immediate:	1	0	0	13-Bit Constant												

Opcode examples are shown of both direct and indirect addressing or of the use of an immediate operand.

Description This section decribes the instruction execution and its effect on the rest of the processor or memory contents. Any constraints on the operands imposed by the processor or the assembler are also described here. The description parallels and supplements the information given by the execution block.

Words 1

The digit specifies the number of memory words required to store the instruction and its extension words.

Cycles 1

The digit specifies the number of cycles required to execute the instruction.

Example

```
ADD    DAT1,3   ;(DP = 0)
or
ADD    *,3      ;If current auxiliary register contains 1.
```

	Before Instruction		After Instruction
Data Memory 1	2h	Data Memory 1	2h
ACC	7h	ACC	17h

The sample code presented in the above format shows the effect of the code on memory and/or registers.

ABS — Absolute Value of Accumulator

Syntax [<label>] ABS

Operands None

Execution
(PC) + 1 → PC
If (ACC) < 0:
 Then −(ACC) → ACC

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0

Description If the contents of the accumulator are greater than or equal to zero, the accumulator is unchanged by the execution of ABS. If the contents of the accumulator are less than zero, the accumulator is replaced by its two's-complement value.

Note that 80000000h is a special case. When the overflow mode is not set, the ABS of 80000000h is 80000000h. When in the overflow mode, the ABS of 80000000h is 7FFFFFFFh.

Words 1

Cycles 1

Example `ABS`

	Before Instruction		After Instruction
ACC	1234h	ACC	1234h
ACC	0FFFFFFFFh	ACC	1h

Syntax

Direct: [<label>] ADD <dma>[,<shift>]
Indirect: [<label>] ADD {*|*+|*-}[,<shift>[,<next ARP>]]

Operands $0 \leq \text{dma} \leq 127$
ARP = 0 or 1

Execution (PC) + 1 → PC
$(\text{ACC}) + (\text{dma}) \times 2^{\text{shift}} \rightarrow \text{ACC}$

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	0	0	0	Shift				0	Data Memory Address						
Indirect:	0	0	0	0	Shift				1	See Section 4.1						

Description Contents of the addressed data memory location are left-shifted and added to the accumulator. During shifting, low-order bits are zero-filled, and high-order bits are sign-extended. The result is stored in the accumulator.

Words 1

Cycles 1

Example 1

```
ADD   DAT1,3    ;(DP = 0)
or
ADD   *,3       ;If current auxiliary register contains 1.
```

	Before Instruction		After Instruction
Data Memory 1	2h	Data Memory 1	2h
ACC	7h	ACC	17h

Example 2

```
ADD   DAT2,4    ;(DP = 0)
or
ADD   *,4       ;If current auxiliary register contains 2.
```

	Before Instruction		After Instruction
Data Memory 2	8B0Eh	Data Memory 2	8B0Eh
ACC	0h	ACC	0FFF8B0E0h

ADDH Add to High Accumulator

Syntax

Direct: [<label>] ADDH <dma>
Indirect: [<label>] ADDH {*|*+|*-}[,<next ARP>]

Operands

$0 \leq \text{dma} \leq 127$
ARP = 0 or 1

Execution

(PC) + 1 → PC
$(\text{ACC}) + (\text{dma}) \times 2^{16} \rightarrow \text{ACC}$

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	0	0	0	0	0	0	Data Memory Address						
Indirect:	0	1	1	0	0	0	0	0	1	See Section 4.1						

Description

Contents of the addressed data memory location are added to the upper half of the accumulator (bits 31 through 16). Low-order bits are unaffected by ADDH.

The ADDH instruction may be used in performing 32-bit arithmetic.

Words

1

Cycles

1

Example

```
ADDH   DAT5   ;(DP = 0)
```
or
```
ADDH   *      ;If current auxiliary register contains 5.
```

	Before Instruction		After Instruction
Data Memory 5	4h	Data Memory 5	4h
ACC	13h	ACC	40013h

ADDS — Add to Accumulator with Sign-Extension Suppressed — ADDS

Syntax

Direct: [<label>] ADDS <dma>
Indirect: [<label>] ADDS {*|*+|*-}[,<next ARP>]

Operands

$0 \leq \text{dma} \leq 127$
ARP = 0 or 1

Execution

(PC) + 1 → PC
(ACC) + (dma) → ACC
(dma) is a 16-bit unsigned number.
Affects OV; affected by OVM.

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	0	0	0	0	1	0	Data Memory Address						
Indirect:	0	1	1	0	0	0	0	1	1	See Section 4.1						

Description

Contents of the specified data memory location are added with sign-extension suppressed. The data is treated as a 16-bit unsigned number rather than a two's-complement number. Therefore, there is no sign-extension as with the ADD instruction.

The ADDS instruction can be used in implementing 32-bit arithmetic.

Words 1

Cycles 1

Example

```
ADDS    DAT11   ;(DP = 0)
or
ADDS    *       ;If current auxiliary register contains 11.
```

	Before Instruction		After Instruction
Data Memory 11	0F006h	Data Memory 11	0F006h
ACC	3h	ACC	0F009h

AND — AND with Low-Order Bits of Accumulator

Syntax

Direct: [<label>] AND <dma>
Indirect: [<label>] AND {*|*+|*-}[,<next ARP>]

Operands $0 \leq \text{dma} \leq 127$
ARP = 0 or 1

Execution (PC) + 1 → PC
(ACC(15-0)).AND.(dma) → ACC(15-0)
0 → ACC(31-16)

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	1	1	0	0	1	0	Data Memory Address						
Indirect:	0	1	1	1	1	0	0	1	1	See Section 4.1						

Description The lower half of the accumulator is ANDed with the contents of the addressed data memory location. The upper half of the accumulator is ANDed with all zeroes. Therefore, the upper half of the accumulator is always zeroed by the AND instruction.

Words 1

Cycles 1

Example

```
AND     DAT16   ;(DP = 0)
or
AND     *       ;If current auxiliary register contains 16.
```

	Before Instruction		After Instruction
Data Memory 16	0FFh	Data Memory 16	0FFh
ACC	12345678h	ACC	78h

Add P Register to Accumulator APAC

Syntax [<label>] APAC

Operands None

Execution (PC) + 1 → PC
(ACC) + (P register) → ACC
Affects OV; affected by OVM.

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1

Description The contents of the P register, the result of a multiply, are added to the contents of the accumulator. The result is stored in the accumulator.

The APAC instruction is a subset of the LTA and LTD instructions.

Words 1

Cycles 1

Example

```
APAC
```

	Before Instruction		After Instruction
P	40h	P	40h
ACC	20h	ACC	60h

B Branch Unconditionally

Syntax [<label>] B <pma>

Operands 0 ≤ pma ≤ 4095

Execution pma → PC

Encoding

<table>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="16">Program Memory Address</td></tr>
</table>

Description Control passes to the designated program memory address (pma). Pma can be either a symbolic or a numeric address.

Words 2

Cycles 2

Example

```
B     PRG191     ;191 is loaded into the program counter,
                 ;and the program continues running from
                 ;that location.
```

Branch on Auxiliary Register Not Zero BANZ

Syntax [<label>] BANZ <pma>

Operands $0 \leq \text{pma} \leq 4095$

Execution

```
If (AR bits 8-0) ≠ 0:
  Then pma → PC;
  Else (PC) + 2 → PC
(AR) - 1 → AR.
```

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
Program Memory Address															

Description If the lower nine bits of the current auxiliary register are not equal to zero, then the address contained in the following word is loaded into the program counter. If these bits are equal to zero, the current program counter is incremented by two. In either case, the auxiliary register is decremented. Note that the test for zero is performed before decrementing the auxiliary register. The branch to a location in program is specified by the program memory address (pma). Pma can be either a symbolic or numeric address.

Words 2

Cycles 2

Example

```
BANZ      PRG35
```

	Before Instruction		After Instruction
AR	1h	AR	0h
PC	46h	PC	35h

or

	Before Instruction		After Instruction
AR	0h	AR	0FFFFh
PC	46h	PC	48h

> **Note:**
>
> BANZ is designed for loop control using the auxiliary registers as loop counters. The auxiliary register is decremented after testing for zero. The auxiliary registers also behave as modulo 512 counters.

BGEZ Branch if Accumulator Greater Than or Equal to Zero BGEZ

Syntax [<label>] BGEZ <pma>

Operands 0 ≤ pma ≤ 4095

Execution If (ACC) ≥ 0:
Then pma → PC;
Else (PC) + 2 → PC.

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0
Program Memory Address															

Description If the contents of the accumulator are greater than or equal to zero, then branch to the specified program memory location. The branch to a location in program is specified by the program memory address (pma). Pma can be either a symbolic or numeric address.

Words 2

Cycles 2

Example

```
BGEZ    PRG217     ;217 is loaded into the program counter
                   ;if the accumulator is greater than or
                   ;equal to zero.
```

Syntax [<label>] BGZ <pma>

Operands 0 ≤ pma ≤ 4095

Execution
If (ACC) > 0:
Then pma → PC;
Else (PC) + 2 → PC.

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
Program Memory Address															

Description If the contents of the accumulator are greater than zero, then branch to the specified program memory location. The branch to a location in program is specified by the program memory address (pma). Pma can be either a symbolic or numeric address.

Words 2

Cycles 2

Example

```
BGZ     PRG342     ;342 is loaded into the program counter
                   ;if the accumulator is greater than zero.
```

BIOZ — Branch on I/O Status Equal to Zero

Syntax [<label>] BIOZ <pma>

Operands 0 ≤ pma ≤ 4095

Execution If $\overline{\text{BIO}} = 0$:
Then pma → PC;
Else (PC) + 2 → PC.

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0
Program Memory Address															

Description If the $\overline{\text{BIO}}$ pin is active low, then branch to the specified program memory location. Otherwise, the program counter is incremented to the next instruction. The branch to a location in program is specified by the program memory address (pma). Pma can be either a symbolic or numeric address.

The BIOZ instruction in conjunction with the $\overline{\text{BIO}}$ pin can be used to test if a peripheral is ready to send or receive data. Polling the $\overline{\text{BIO}}$ pin using BIOZ may be preferable to an interrupt when executing time-critical loops.

Words 2

Cycles 2

Example

```
BIOZ    PRG64    ;If the BIO pin is active (low), then
                 ;a branch to location 64 occurs. Otherwise,
                 ;the program counter is incremented.
```

BLEZ Branch if Accumulator Less Than or Equal to Zero BLEZ

Syntax [<label>] BLEZ <pma>

Operands 0 ≤ pma ≤ 4095

Execution

```
If (ACC) ≤ 0:
 Then pma → PC;
 Else (PC) + 2 → PC.
```

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0
Program Memory Address															

Description If the contents of the accumulator are less than or equal to zero, then branch to the specified program memory location. The branch to a location in program is specified by the program memory address (pma). Pma can be either a symbolic or numeric address.

Words 2

Cycles 2

Example

```
BLEZ    PRG63    ;63 is loaded into the program counter if
                 ;the accumulator is less than or equal to
                 ;zero.
```

Syntax [<label>] BLZ <pma>

Operands $0 \leq \text{pma} \leq 4095$

Execution If (ACC) < 0:
  Then pma → PC;
  Else (PC) + 2 → PC.

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
Program Memory Address															

Description If the contents of the accumulator are less than zero, then branch to the specified program memory location. The branch to a location in program is specified by the program memory address (pma). Pma can be either a symbolic or numeric address.

Words 2

Cycles 2

Example

```
BLZ   PRG481   ;481 is loaded into the program counter
               ;if the accumulator is less than zero.
```

Branch if Accumulator Not Equal to Zero — BNZ

Syntax [<label>] BNZ <pma>

Operands 0 ≤ pma ≤ 4095

Execution If (ACC) ≠ 0:
Then pma → PC;
Else (PC) + 2 → PC.

Encoding

<table>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="16">Program Memory Address</td></tr>
</table>

Description If the contents of the accumulator are not equal to zero, then branch to the specified program memory location. The branch to a location in program is specified by the program memory address (pma). Pma can be either a symbolic or numeric address.

Words 2

Cycles 2

Example

```
BNZ     PRG320     ;320 is loaded into the program counter
                   ;if the accumulator does not equal zero.
```

Syntax [<label>] BV <pma>

Operands 0 ≤ pma ≤ 4095

Execution

If overflow (OV) status bit = 1:
 Then pma → PC and 0 → OV;
 Else (PC) + 2 → PC.
Affects OV; affected by OV.

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
Program Memory Address															

Description If the overflow (OV) flag has been set, then a branch to the specified program memory location occurs and the overflow flag is cleared. Otherwise, the program counter is incremented to the next instruction. The branch to a location in program is specified by the program memory address (pma). Pma can be either a symbolic or numeric address.

Words 2

Cycles 2

Example

```
BV      PRG610     ;If an overflow has occurred since the
                   ;overflow flag was last cleared, then 610
                   ;is loaded into the program counter and
                   ;OV is cleared. Otherwise, the program
                   ;counter is incremented.
```

Branch if Accumulator Equals Zero BZ

Syntax [<label>] BZ <pma>

Operands 0 ≤ pma ≤ 4095

Execution If (ACC) = 0:
Then pma → PC;
Else (PC) + 2 → PC.

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Program Memory Address															

Description If the contents of the accumulator are equal to zero, then branch to the specified program memory location. The branch to a location in program is specified by the program memory address (pma). Pma can be either a symbolic or numeric address.

Words 2

Cycles 2

Example

```
BZ      PRG102    ;102 is loaded into the program counter
                  ;if the accumulator is equal to zero.
```

Syntax [<label>] CALA

Operands None

Execution (PC) + 1 → TOS
(ACC(11-0)) → PC

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0

Description The current program counter is incremented and pushed onto the top of the stack. Then, the contents of the 12 least significant bits of the accumulator are loaded into the PC.

The CALA instruction is used to perform computed subroutine calls.

Words 1

Cycles 2

Example `CALA`

	Before Instruction		After Instruction
PC	25h	PC	83h
ACC	83h	ACC	83h
Stack	32h 75h 84h 49h	Stack	26h 32h 75h 84h

Syntax [<label>] CALL <pma>

Operands 0 ≤ pma ≤ 4095

Execution (PC) + 2 → TOS
pma → PC

Encoding

<table>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="16">Program Memory Address</td></tr>
</table>

Description The current program counter is incremented by two and pushed onto the top of the stack. The specified program memory address (pma) is then loaded into the PC. Pma can be either a symbolic or a numeric address.

Words 2

Cycles 2

Example

```
CALL    PRG109
```

	Before Instruction		After Instruction
PC	33h	PC	6Dh
Stack	71h 48h 16h 80h	Stack	35h 71h 48h 16h

DINT — Disable Interrupt

Syntax [<label>] DINT

Operands None

Execution
(PC) + 1 → PC
1 → interrupt mode (INTM) status bit
Affects INTM.

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1

Description The interrupt mode (INTM) status bit is set to logic 1. Maskable interrupts are disabled immediately after the DINT instruction executes. Interrupts are also disabled by a reset. Note that the LST instruction does not affect INTM.

The unmaskable interrupt, $\overline{RS}$, is not disabled by this instruction. Interrupts are also disabled by a reset.

Words 1

Cycles 1

Example

```
DINT            ;Maskable interrupts are disabled, and INTM
                ;is set to one.
```

Data Move in Data Memory — DMOV

Syntax

Direct: [<label>] DMOV <dma>
Indirect: [<label>] DMOV {*|*+|*-}[,<next ARP>]

Operands

$0 \leq \text{dma} \leq 127$
ARP = 0 or 1

Execution

(PC) + 1 → PC
(dma) → dma + 1

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	0	1	0	0	1	0	Data Memory Address						
Indirect:	0	1	1	0	1	0	0	1	1	See Section 4.1						

Description

The contents of the specified data memory address are copied into the contents of the next higher address. When data is copied from the addressed location to the next higher location, the contents of the addressed location remain unaltered.

The data move function is useful in implementing the z^{-1} delay encountered in digital signal processing. The DMOV function is included in the LTD instruction (see LTD for more information).

Words

1

Cycles

1

Example

```
DMOV    DAT8
```

or

```
DMOV    *        ;If current auxiliary register contains 8.
```

	Before Instruction		After Instruction
Data Memory 8	43h	Data Memory 8	43h
Data Memory 9	2h	Data Memory 9	43h

Syntax [<label>] EINT

Operands None

Execution (PC) + 1 → PC
0 → interrupt mode (INTM) status bit
Affects INTM.

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0

Description The interrupt mode (INTM) status bit is cleared to logic 0. Maskable interrupts are enabled after the instruction following EINT executes. This allows an interrupt service routine to re-enable interrupts and execute a RET instruction before any other pending interrupts are processed. Note that the EINT instruction should not be used immediately preceding a branch instruction.

The LST instruction does not affect INTM. (See the DINT instruction for further information.)

Words 1

Cycles 1

Example

```
EINT            ;Maskable interrupts are enabled, and INTM
                ;is set to zero.
```

Input Data from Port IN

Syntax

Direct: [<label>] IN <dma>,<PA>
Indirect: [<label>] IN {*|*+|*-},<PA>[,<next ARP>]

Operands

$0 \leq \text{dma} \leq 127$
ARP = 0 or 1
$0 \leq \text{port address PA} \leq 7$

Execution

(PC) + 1 → PC
Port address → address lines A2/PA2-A0/PA0
0 → address bus A11-A3
Data bus D15-D0 → dma

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	0	0	0	Port Address			0	Data Memory Address						
Indirect:	0	1	0	0	0	Port Address			1	See Section 4.1						

Description

The IN instruction reads data from a peripheral and places it in data memory. This is a two-cycle instruction. During the first cycle, the port address is sent to address lines A2/PA2-A0/PA0. $\overline{\text{DEN}}$ goes low during the same cycle, strobing in the data that the addressed peripheral places on the data bus D15-D0. On the TMS32010/C10/C15/E15, $\overline{\text{MEN}}$ will remain high when $\overline{\text{DEN}}$ is active. On the TMS320C17/E17, the $\overline{\text{MEN}}$ signal is not available.

Words 1

Cycles 2

Example

```
IN      STAT,PA5      ;Read in word from peripheral on port
                      ;address 5.  Store in data memory
                      ;location STAT.
```

or

```
LARK    1,20          ;Load AR1 with decimal 20.
LARP    1             ;Load ARP with decimal 1.
IN      *-,PA1,0      ;Read in word from peripheral on port
                      ;address 1.  Store in data memory
                      ;location 20.  Decrement AR1 to 19.
                      ;Load the ARP with 0.
```

Syntax

Direct: [<label>] LAC <dma>[,<shift>]
Indirect: [<label>] LAC {*|*+|*-}[,<shift>[,<next ARP>]]

Operands

$0 \leq \text{dma} \leq 127$
ARP = 0 or 1
$0 \leq \text{shift} \leq 15$ (defaults to 0)

Execution

(PC) + 1 → PC
(dma) x 2^{shift} → ACC

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	0	1	0	Shift				0	Data Memory Address						
Indirect:	0	0	1	0	Shift				1	See Section 4.1						

Description

Contents of the specified data memory address are left-shifted and loaded into the accumulator. During shifting, low-order bits are zero-filled. High-order bits are sign-extended.

Words

1

Cycles

1

Example

```
LAC   DAT6,4   ;(DP = 0)
or
LAC   *,4      ;If current auxiliary register contains 6.
```

	Before Instruction		After Instruction
Data Memory 6	1h	Data Memory 6	1h
ACC	0h	ACC	10h

Load Accumulator Immediate LACK

Syntax [<label>] LACK <constant>

Operands 0 ≤ constant ≤ 255

Execution (PC) + 1 → PC
8-bit positive constant → ACC

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	0	8-Bit Constant							

Description The 8-bit constant is loaded into the accumulator right-justified. The upper 24 bits of the accumulator are zeroed (i.e., sign extension is suppressed).

Words 1

Cycles 1

Example

```
LACK   15h
```

	Before Instruction		After Instruction
ACC	31h	ACC	15h

Syntax

Direct: [<label>] LAR <AR>,<dma>
Indirect: [<label>] LAR <AR>,{*|*+|*-}[,<next ARP>]

Operands

$0 \leq \text{dma} \leq 127$
auxiliary register AR = 0 or 1
ARP = 0 or 1

Execution

(PC) + 1 → PC
(dma) → auxiliary register AR

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	0	1	1	1	0	0	AR	0	Data Memory Address						
Indirect:	0	0	1	1	1	0	0	AR	1	See Section 4.1						

Description

The contents of the specified data memory address are loaded into the designated auxiliary register. The LAR and SAR (store auxiliary register) instructions can be used to load and store the auxiliary registers during subroutine calls and interrupts. If an auxiliary register is not being used for indirect addressing, LAR and SAR enable the register to be used as an additional storage register, especially for swapping values between data memory locations without affecting the contents of the accumulator.

If indirect addressing is used to load the current auxiliary register (i.e., if the AR specified in the LAR instruction is the AR pointed to by the ARP), then the new value will be loaded into the auxiliary register from data memory and any decrement or increment specified will not be performed.

Words 1

Cycles 1

Example

```
LAR     AR0,DAT19
```

	Before Instruction		After Instruction
Data Memory 19	18h	Data Memory 19	18h
AR0	6h	AR0	18h

also,

```
LARP   0
LAR    AR0,*-
```

Data Memory 7	32h	Data Memory 7	32h
AR0	7h	AR0	32h

Load Auxiliary Register Immediate LARK

Syntax [<label>] LARK <AR>,<constant>

Operands 0 ≤ constant ≤ 255
auxiliary register AR = 0 or 1

Execution (PC) + 1 → PC
8-bit constant → auxiliary register AR

Encoding

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>AR</td><td colspan="8">8-Bit Constant</td></tr>
</table>

Description The 8-bit positive constant is loaded into the designated auxiliary register right-justified and zero-filled (i.e., sign-extension suppressed).

LARK is useful for loading an initial loop counter value into an auxiliary register for use with the BANZ instruction.

Words 1

Cycles 1

Example

```
LARK    AR0,21h
```

	Before Instruction		After Instruction
AR0	0h	AR0	21h

Syntax [<label>] LARP <constant>

Operands 0 ≤ constant ≤ 1

Execution
(PC) + 1 → PC
Constant → ARP
Affects ARP

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	ARP

Description The auxiliary register pointer is loaded with the one-bit constant identifying the desired auxiliary register. ARP can also be modified by the LST and MAR instructions, as well as any instruction that is used in the indirect addressing mode.

The LARP instruction is a subset of MAR; i.e., the opcode is the same as MAR in the indirect addressing mode. The instruction `MAR *,<next ARP>` has the same effect as LARP.

Words 1

Cycles 1

Example

```
LARP    1       ;Any succeeding instructions will use
                ;auxiliary register AR1 for indirect
                ;addressing.
```

Load Data Memory Page Pointer

Syntax

Direct: [<label>] LDP <dma>
Indirect: [<label>] LDP {*|*+|*-}[,<next ARP>]

Operands

0 ≤ dma ≤ 127
ARP = 0 or 1

Execution

(PC) + 1 → PC
LSB of (dma) → data memory page pointer (DP = 0 or 1)
Affects DP.

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	0	1	1	1	1	0	Data Memory Address						
Indirect:	0	1	1	0	1	1	1	1	1	See Section 4.1						

Description

The least significant bit of the contents of the specified data memory address is loaded into the DP (data memory page pointer) register. All higher-order bits are ignored in the data word. DP = 0 defines page 0 that contains words 0-127. DP = 1 defines page 1 that contains words 128-143/255. The DP may also be loaded by the LST and LDPK instructions.

Words

1

Cycles

1

Example

```
LDP    DAT1     ;LSB of location DAT1 is loaded into DP.
or
LDP    *,1      ;LSB of location currently addressed by
                ;auxiliary register is loaded into DP.
                ;ARP is set to 1.
```

	Before Instruction		After Instruction
Data Memory 1	0FEDCh	Data Memory 1	0FEDCh
DP	1h	DP	0h

LDPK Load Data Memory Page Pointer Immediate

Syntax [<label>] LDPK <constant>

Operands 0 ≤ constant ≤ 1

Execution
(PC) + 1 → PC
Constant → data memory page pointer (DP)
Affects DP.

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	DP

Description The DP (data memory page pointer) register is loaded with a 1-bit constant. DP = 0 defines page 0 that contains words 0-127. DP = 1 defines page 1 that contains words 128-143/255. The DP may also be loaded by the LST and LDP instructions.

Words 1

Cycles 1

Example

```
LDPK   0   ;The data page pointer is set to 0.
```

Syntax

Direct: [<label>] LST <dma>
Indirect: [<label>] LST {*|*+|*-}[,<next ARP>]

Operands

$0 \leq \text{dma} \leq 127$
ARP = 0 or 1

Execution

(PC) + 1 → PC
(dma) → status register bits
Affects ARP, OV, OVM, and DP.
Does not affect INTM.

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	1	1	0	1	1	0	Data Memory Address						
Indirect:	0	1	1	1	1	0	1	1	1	See Section 4.1						

Description

The status register is loaded with the addressed data memory value. Note that the INTM (interrupt mode) bit is unaffected by LST.

The LST instruction is used to load the status register after interrupts and subroutine calls. The status register contains the status bits: OV (overflow flag) bit, OVM (overflow mode) bit, ARP (auxiliary register pointer), and DP (data memory page pointer). These bits were stored (by the SST instruction) in the data memory word as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OV	OVM	INTM	1	1	1	1	ARP	1	1	1	1	1	1	0	DP

Words

1

Cycles

1

Example

```
LARP    0
or
LST     *,1     ;The data memory word addressed by the
                ;contents of auxiliary register AR0
                ;replaces the status bits. ARP becomes 1.
```

Note:

When using direct addressing, the SST instruction always saves status on page 1. The LST instruction will not automatically restore status from page 1. Therefore, the user must specify the correct data page pointer.

LT Load T Register

Syntax

Direct: [<label>] LT <dma>
Indirect: [<label>] LT {*|*+|*-}[,<next ARP>]

Operands $0 \le \text{dma} \le 127$
ARP = 0 or 1

Execution (PC) + 1 → PC
(dma) → T register

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	0	1	0	1	0	0	Data Memory Address						
Indirect:	0	1	1	0	1	0	1	0	1	See Section 4.1						

Description The T register is loaded with the contents of the specified data memory location. The LT instruction may be used to load the T register in preparation for multiplication (see the LTA, LTD, MPY, and MPYK instructions).

Words 1

Cycles 1

Example

```
LT    DAT24    ;(DP = 0)
or
LT    *        ;If current auxiliary register contains 24.
```

	Before Instruction		After Instruction
Data Memory 24	62h	Data Memory 24	62h
T	3h	T	62h

Load T Register and Accumulate Previous Product LTA

Syntax

Direct: [<label>] LTA <dma>
Indirect: [<label>] LTA {*|*+|*-}[,<next ARP>]

Operands 0 ≤ dma ≤ 127
ARP = 0 or 1

Execution (PC) + 1 → PC
(dma) → T register
(ACC) + (P register) → ACC
Affects OV; affected by OVM.

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	0	1	1	0	0	0	Data Memory Address						
Indirect:	0	1	1	0	1	1	0	0	1	See Section 4.1						

Description The T register is loaded with the contents of the specified data memory address. The P register, containing the previous product of the multiply operation, is added to the accumulator, and the result is stored in the accumulator.

The function of the LTA instruction is included in the LTD instruction.

Words 1

Cycles 1

Example

```
LTA    DAT24    ;(DP = 0)
or
LTA    *        ;If current auxiliary register contains 24.
```

	Before Instruction		After Instruction
Data Memory 24	62h	Data Memory 24	62h
T	3h	T	62h
P	0Fh	P	0Fh
ACC	5h	ACC	14h

LTD — Load T Register, Accumulate Previous Product, and Move Data — LTD

Syntax

Direct: [<label>] LTD <dma>
Indirect: [<label>] LTD {*|*+|*-}[,<next ARP>]

Operands

0 ≤ dma ≤ 127
ARP = 0 or 1

Execution

(PC) + 1 → PC
(dma) → T register
(dma) → dma + 1
(ACC) + (P register) → ACC
Affects OV; affected by OVM.

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	0	1	0	1	1	0	Data Memory Address						
Indirect:	0	1	1	0	1	0	1	1	1	See Section 4.1						

Description

The T register is loaded with the contents of the specified data memory address. The contents of the P register are added to the accumulator, and the result is placed in the accumulator. The contents of the specified data memory address are also copied to the next higher data memory address. This function is described under the instruction DMOV.

Words

1

Cycles

1

Example

```
LTD    DAT24    ;(DP = 0)
or
LTD    *        ;If current auxiliary register contains 24.
```

	Before Instruction		After Instruction
Data Memory 24	62h	Data Memory 24	62h
Data Memory 25	0h	Data Memory 25	62h
T	3h	T	62h
P	0Fh	P	0Fh
ACC	5h	ACC	14h

Modify Auxiliary Register MAR

Syntax

Direct: [<label>] MAR <dma>
Indirect: [<label>] MAR {*|*+|*-}[,<next ARP>]

Operands 0 ≤ dma ≤ 127
ARP = 0 or 1

Execution (PC) + 1 → PC
Modifies AR(ARP), ARP as specified by the indirect addressing field (acts as a NOP in direct addressing).

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	0	1	0	0	0	0	Data Memory Address						
Indirect:	0	1	1	0	1	0	0	0	1	See Section 4.1						

Description In the indirect addressing mode, the auxiliary registers are either incremented or decremented and the ARP is modified; however, no use is made of the memory being referenced. MAR is used only to modify the auxiliary registers or the ARP. ARP may also be loaded by an LST instruction.

MAR acts as a no-operation (NOP) instruction in the direct addressing mode. Also, the LARP instruction is a subset of MAR (i.e., MAR *,0 performs the same function as LARP 0).

Words 1

Cycles 1

Example 1

```
MAR  *,1    ;Load the ARP with 1.
```

	Before Instruction		After Instruction
ARP	0	ARP	1

Example 2

```
MAR  *-    ;Decrement current auxiliary register (in this
           ;case, AR1)
```

	Before Instruction		After Instruction
AR1	35h	AR1	34h

Example 3

```
MAR  *+,0  ;Increment current auxiliary register (AR1)
           ;and load ARP with 0.
```

	Before Instruction		After Instruction
AR1	34h	AR1	35h
ARP	1	ARP	0

Syntax

Direct: [<label>] MPY <dma>
Indirect: [<label>] MPY {*|*+|*-}[,<next ARP>]

Operands

0 ≤ dma ≤ 127
ARP = 0 or 1

Execution

(PC) + 1 → PC
(T register) x (dma) → P register

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	0	1	1	0	1	0	Data Memory Address						
Indirect:	0	1	1	0	1	1	0	1	1	See Section 4.1						

Description

The contents of the T register are multiplied by the contents of the addressed data memory location. The result is placed in the P register.

During an interrupt, all registers except the P register can be saved and restored directly. However, the first-generation TMS320 devices have hardware protection against servicing an interrupt between an MPY or MPYK instruction and the following instruction. For this reason, it is advisable to follow MPY and MPYK with LTA, LTD, PAC, APAC, or SPAC.

Note that no provisions are made for the condition of 8000h x 8000h. If this condition arises, the product will be 0C0000000h.

Words

1

Cycles

1

Example

```
MPY    DAT13    ;(DP = 0)
or
MPY    *        ;If current auxiliary register contains 13.
```

	Before Instruction		After Instruction
Data Memory 13	7h	Data Memory 13	7h
T	6h	T	6h
P	36h	P	2Ah

Syntax [<label>] MPYK <constant>

Operands $-2^{12} \leq \text{constant} < 2^{12}$

Execution (PC) + 1 → PC
(T register) x constant → P register

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	13-Bit Constant												

Description The contents of the T register are multiplied by the signed 13-bit constant. The result is loaded into the P register.

During an interrupt, all registers except the P register can be saved and restored directly. Since no provision is made to save the contents of the P register during an interrupt, the MPYK instruction should be followed by one of the following instructions: PAC, APAC, SPAC, LTA, or LTD. Provision is made in hardware to inhibit interrupt during MPYK until the next instruction is executed.

Words 1

Cycles 1

Example

```
MPYK   -9
```

	Before Instruction		After Instruction
T	7h	T	7h
P	2Ah	P	0FFFFFFC1h

No Operation NOP

Syntax [<label>] NOP

Operands None

Execution (PC) + 1 → PC

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0

Description No operation is performed. NOP affects only the PC.

NOP is useful as a pad or temporary instruction during program development.

Words 1

Cycles 1

Example

```
NOP
```

Syntax

Direct: [<label>] OR <dma>
Indirect: [<label>] OR {*|*+|*-}[,<next ARP>]

Operands $0 \leq \text{dma} \leq 127$
ARP = 0 or 1

Execution (PC) + 1 → PC
(ACC(15-0)) .OR.dma → ACC(15-0)
(ACC(31-16)) → ACC(31-16)

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	1	1	0	1	0	0	Data Memory Address						
Indirect:	0	1	1	1	1	0	1	0	1	See Section 4.1						

Description The low-order bits of the accumulator are ORed with the contents of the addressed data memory location. The high-order bits of the accumulator are ORed with all zeroes. Therefore, the upper half of the accumulator is unaffected by this instruction. The result is stored in the accumulator.

The OR instruction is useful for comparing selected bits of a data word.

Words 1

Cycles 1

Example

```
OR    DAT88   ;(DP = 0)
or
OR    *       ;Where current auxiliary register contains 88.
```

	Before Instruction		After Instruction
Data Memory 88	0F000h	Data Memory 88	0F000h
ACC	100002h	ACC	10F002h

Output Data to Port OUT

Syntax

Direct: [<label>] OUT <dma>,<PA>
Indirect: [<label>] OUT {*|*+|*-},<PA>[,<next ARP>]

Operands

0 ≤ dma ≤ 127
ARP = 0 or 1
0 ≤ port address PA ≤ 7

Execution

(PC) + 1 → PC
Port address PA → address bus A2/PA2-A0/PA0
0 → address bus A11-A3
(dma) → data bus D15-D0

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	0	0	1	Port Address			0	Data Memory Address						
Indirect:	0	1	0	0	1	Port Address			1	See Section 4.1						

Description

The OUT instruction transfers data from data memory to an external peripheral. The first cycle of this instruction places the port address onto address lines A2/PA2-A0/PA0. During the same cycle, $\overline{\text{WE}}$ goes low and the data word is placed on the data bus D15-D0. On the TMS32010/C10/C15/E15, $\overline{\text{MEN}}$ remains high during the first cycle. On the TMS320C17/E17, the $\overline{\text{MEN}}$ signal is not available.

Words

1

Cycles

2

Example

```
OUT     120,7   ;Output data word stored in data memory
                ;location 120 to peripheral on port
                ;address 7.
or
OUT     *,5     ;Output data word referenced by current
                ;auxiliary register to peripheral on port
                ;address 5.
```

PAC — Load Accumulator with P Register

Syntax [<label>] PAC

Operands None

Execution
(PC) + 1 → PC
(P register) → ACC

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	0

Description The contents of the P register resulting from a multiply are loaded into the accumulator.

Words 1

Cycles 1

Example

```
PAC
```

	Before Instruction		After Instruction
P	144h	P	144h
ACC	23h	ACC	144h

Pop Top of Stack to Low Accumulator POP

Syntax [<label>] POP

Operands None

Execution (PC) + 1 → PC
(TOS) → ACC(11-0)
0 → ACC(31-12)
Pop stack one level.

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1

Description The contents of the top of the stack (TOS) are copied to the low accumulator, and the stack popped after the contents are copied. The next element on the stack becomes the top of the stack. The upper bits (31-12) of the accumulator are zeroed. The hardware stack is a last-in, first-out stack with four locations. Any time a pop occurs, every stack value is copied to the next higher stack location, and the top value is removed from the stack. After a pop, the bottom two stack words will have the same value. Because each stack value is copied, if more than three pops (due to POP or RET instructions) occur before any pushes occur, all levels of the stack contain the same value.

Words 1

Cycles 2

Example `POP`

	Before Instruction		After Instruction
ACC	82h	ACC	45h
Stack	45h 16h 7h 33h	Stack	16h 7h 33h 33h

PUSH — Push Low Accumulator onto Stack

Syntax [<label>] PUSH

Operands None

Execution
(PC) + 1 → PC
Push all stack locations down one level.
(ACC(11-0)) → TOS

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0

Description The contents of the lower 12 bits (11-0) of the accumulator are copied onto the top of the hardware stack. The stack is pushed down before the accumulator value is copied. The hardware stack is a last-in, first-out stack with four locations. If more than four pushes (due to CALA, CALL, PUSH, TBLR, or TBLW instructions or interrupts) occur before a pop, the first data values written will be lost with each succeeding push.

Words 1

Cycles 2

Example

```
PUSH
```

	Before Instruction		After Instruction
ACC	7h	ACC	7h
Stack	2h 5h 3h 0h	Stack	7h 2h 5h 3h

Return from Subroutine — RET

Syntax [<label>] RET

Operands None

Execution (TOS) → PC
Pop stack one level.

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	1

Description The contents of the top of stack are copied into the program counter. The stack is then popped one level. RET is used in conjunction with CALA and CALL for subroutines and interrupts.

Words 1

Cycles 2

Example

```
RET
```

	Before Instruction		After Instruction
PC	96h	PC	37h
Stack	37h 45h 75h 75h	Stack	45h 75h 75h 75h

ROVM Reset Overflow Mode

Syntax [<label>] ROVM

Operands None

Execution
(PC) + 1 → PC
0 → OVM status bit
Affects OVM.

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	0

Description The OVM status bit is reset to logic zero. This disables the overflow mode, in which the device was placed by the SOVM instruction. If an overflow occurs with OVM reset, the OV (overflow flag) is set, and the overflowed result is placed in the accumulator. OVM may also be loaded by the LST and SOVM instructions (see the SOVM instruction).

Words 1

Cycles 1

Example

```
ROVM                ;The overflow mode bit OVM is reset,
                    ;disabling the overflow mode on any
                    ;subsequent arithmetic operations.
```

Store High Accumulator with Shift — SACH

Syntax

Direct: [<label>] SACH <dma>[,<shift>]
Indirect: [<label>] SACH {*|*+|*-}[,<shift>[,<next ARP>]]

Operands

$0 \leq \text{dma} \leq 127$
ARP = 0 or 1
shift = 0, 1, or 4

Execution

(PC) + 1 → PC
16 MSBs of (ACC) x 2^{shift} → dma

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	0	1	1	Shift			0	Data Memory Address						
Indirect:	0	1	0	1	1	Shift			1	See Section 4.1						

Description

The SACH instruction copies the entire accumulator into a shifter. It then left-shifts this entire 32-bit number 0, 1, or 4 bits, and copies the upper 16 bits of the shifted value into data memory. The accumulator itself remains unaffected.

Words

1

Cycles

1

Example

```
SACH    DAT70,1 ;(DP = 0)
```
or
```
SACH    *,1     ;If current auxiliary register contains 70.
```

	Before Instruction		After Instruction
ACC	4208001h	ACC	4208001h
Data Memory 70	0h	Data Memory 70	841h

SACL — Store Low Accumulator

Syntax

Direct: [<label>] SACL <dma>
Indirect: [<label>] SACL {*|*+|*-}[,<0>[,<next ARP>]]

Operands $0 \leq \text{dma} \leq 127$
ARP = 0 or 1
shift = 0

Execution (PC) + 1 → PC
(ACC(15-0)) → dma

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	0	1	0	0	0	0	0	Data Memory Address						
Indirect:	0	1	0	1	0	0	0	0	1	See Section 4.1						

Description The low-order bits of the accumulator are stored in data memory. There is no shift associated with this instruction, although a shift code of zero MUST be specified if the ARP is to be changed.

Words 1

Cycles 1

Example

```
SACL    DAT71    ;(DP = 0)
or
SACL    *        ;If current auxiliary register contains 71.
```

	Before Instruction		After Instruction
Data Memory 71	5h	Data Memory 71	8421h
ACC	7C638421h	ACC	7C638421h

Store Auxiliary Register SAR

Syntax

Direct: [<label>] SAR <AR>,<dma>
Indirect: [<label>] SAR <AR>,{*|*+|*-}[,<next ARP>]

Operands

0 ≤ dma ≤ 127
auxiliary register AR = 0 or 1
ARP = 0 or 1

Execution

(PC) + 1 → PC
(auxiliary register AR) → dma

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	0	1	1	0	0	0	AR	0	Data Memory Address						
Indirect:	0	0	1	1	0	0	0	AR	1	See Section 4.1						

Description

The contents of the designated auxiliary register are stored in the addressed data memory location. For more information, see the LAR instruction.

Words

1

Cycles

1

Example 1

```
SAR   AR0,DAT30 ;(DP = 0)
```

or

```
SAR   AR0,*     ;If current auxiliary register contains 30.
```

	Before Instruction		After Instruction
AR0	37h	AR0	37h
Data Memory 30	18h	Data Memory 30	37h

Example 2

```
LARP   AR0
SAR    AR0,*+
```

AR0	5h	AR0	6h
Data Memory 5	0h	Data Memory 5	6h

Warning:

Special problems arise when SAR is used to store the current auxiliary register with indirect addressing if auto-increment/decrement is used.

```
LARP   AR0
LARK   AR0,10
SAR    AR0,*+  or  SAR   AR0,*-
```

In this case, SAR AR0,*+ will cause the value 11 to be stored in location 10. SAR AR0,*- will cause the value 9 to be stored in location 10.

Set Overflow Mode SOVM

Syntax [<label>] SOVM

Operands None

Execution

(PC) + 1 → PC
1 → overflow mode (OVM) status bit
Affects OVM.

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1

Description The OVM status bit is set to logic 1, which enables the overflow (saturation) mode. If an overflow occurs with OVM set, the overflow flag OV is set, and the accumulator is set to the largest representable 32-bit positive (7FFFFFFFh) or negative (80000000h) number according to the direction of overflow. OVM may also be loaded by the LST and ROVM instructions. (See the ROVM instruction for further information.)

Words 1

Cycles 1

Example

```
SOVM            ;The overflow mode bit OVM is set, enabling
                ;the overflow mode on any subsequent
                ;arithmetic operations.
```

SPAC — Subtract P Register from Accumulator

Syntax [<label>] SPAC

Operands None

Execution
(PC) + 1 → PC
(ACC) - (P register) → ACC
Affects OV; affected by OVM.

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0

Description The contents of the P register are subtracted from the contents of the accumulator. The result is stored in the accumulator. Note that the P register is always sign-extended.

Words 1

Cycles 1

Example `SPAC`

	Before Instruction		After Instruction
P	24h	P	24h
ACC	3Ch	ACC	18h

Syntax

Direct: [<label>] SST <dma>
Indirect: [<label>] SST {*|*+|*-}[,<next ARP>]

Operands

$0 \leq \text{dma} \leq 15$ (TMS32010/C10/11)
$0 \leq \text{dma} \leq 127$ (TMS320C15/C17)
ARP = 0 or 1

Execution

(PC) + 1 → PC
(status register) → specified dma (page 1 only in direct addressing)

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	1	1	1	0	0	0	Data Memory Address						
Indirect:	0	1	1	1	1	1	0	0	1	See Section 4.1						

Description

The status bits are saved into the specified data memory address (page 1 only if direct memory addressing is used).

In the direct addressing mode, the status register is always stored in page 1 regardless of the value of the DP register. The processor automatically forces the page to be 1, and the specific location within that page is defined in the instruction. Note that the DP register is not physically modified. This allows storage of the DP register in the data memory on interrupts, etc., in the direct addressing mode without having to change the DP. In the indirect addressing mode, the data memory address is obtained from the auxiliary register selected. (See the LST instruction for more information.)

The SST instruction can be used to store the status bits after interrupts and subroutine calls. These status bits include the OV (overflow flag) bit, OVM (overflow mode) bit, INTM (interrupt mode) bit, ARP (auxiliary register pointer) bit, and DP (data memory page pointer) bit. The status bits are stored in the data memory word as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OV	OVM	INTM	1	1	1	1	ARP	1	1	1	1	1	1	X	DP

X = reserved

Words

1

Cycles

1

Example

```
SST    DAT1    ;(DP = don't care)
or
SST    *,1     ;If current auxiliary register contains 1.
```

	Before Instruction		After Instruction
Status Register	5EFEh	Status Register	5EFEh
Data Memory 1	0Ah	Data Memory 1	5EFEh

SUB

Subtract from Accumulator with Shift

Syntax

Direct: [<label>] SUB <dma>[,<shift>]
Indirect: [<label>] SUB {*|*+|*-}[,<shift>[,<next ARP>]]

Operands

$0 \leq \text{dma} \leq 127$
ARP = 0 or 1
$0 \leq \text{shift} \leq 15$ (defaults to 0)

Execution

(PC) + 1 → PC
(ACC) - [(dma) x 2^{shift}] → ACC
Affects OV; affected by OVM.

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	0	0	1	Shift				0	Data Memory Address						
Indirect:	0	0	0	1	Shift				1	See Section 4.1						

Description

The contents of the addressed data memory location are left-shifted and subtracted from the accumulator. During shifting, the low-order bits are zero-filled. The high-order bit is sign-extended. The result is stored in the accumulator.

Words 1

Cycles 1

Example

```
SUB     DAT59     ;(DP = 0)
or
SUB     *         ;If current auxiliary register contains 59.
```

	Before Instruction		After Instruction
ACC	24h	ACC	13h
Data Memory 59	11h	Data Memory 59	11h

Conditional Subtract SUBC

Syntax

Direct: [<label>] SUBC <dma>
Indirect: [<label>] SUBC {*|*+|*-}[,<next ARP>]

Operands

$0 \leq \text{dma} \leq 127$
ARP = 0 or 1

Execution

(PC) + 1 → PC
$(\text{ACC}) - [(\text{dma}) \times 2^{15}] \rightarrow$ ALU output
If ALU output ≥ 0:
 Then (ALU output) x 2 + 1 → ACC;
 Else (ACC) x 2 → ACC.
Affects OV but NOT affected by OVM (no saturation).

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	0	0	1	0	0	0	Data Memory Address						
Indirect:	0	1	1	0	0	1	0	0	1	See Section 4.1						

Description

The SUBC instruction performs conditional subtraction, which may be used for division. The 16-bit dividend is placed in the low accumulator, and the high accumulator is zeroed. The divisor is in data memory. SUBC is executed 16 times for 16-bit division. After completion of the last SUBC, the quotient of the division is in the lower-order 16-bit field of the accumulator, and the remainder is in the high-order 16 bits of the accumulator. SUBC assumes the divisor and the dividend are both positive.

If the 16-bit dividend contains less than 16 significant bits, the dividend may be placed in the accumulator left-shifted by the number of leading non-significant zeroes. The number of executions of SUBC is reduced from 16 by that number. However, at least one leading zero must always be present since both operands of the SUBC instruction must be positive. Note that the next instruction after SUBC cannot use the accumulator.

The SUBC instruction affects OV but is not affected by OVM. Therefore, the accumulator does not saturate upon positive or negative overflows when executing this instruction.

The above description is for 16-bit integer division. SUBC can also be used in fixed-point division.

Words 1

Cycles 1

Example

```
     LARP   AR0
     LARK   AR0,15
DIV  SUBC   DAT2    ;(DP = 0)
     BANZ   DIV
```

	Before Instruction		After Instruction
Data Memory 2	7h	Data Memory 2	7h
ACC	41h	ACC	20009h

The results above show the execution of all the instructions in the code example.

Syntax

Direct: [<label>] SUBH <dma>
Indirect: [<label>] SUBH {*|*+|*-}[,<next ARP>]

Operands $0 \leq \text{dma} \leq 127$
ARP = 0 or 1

Execution (PC) + 1 → PC
$(\text{ACC}) - [(\text{dma}) \times 2^{16}] \rightarrow \text{ACC}$
Affects OV; affected by OVM.

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	0	0	0	1	0	0	Data Memory Address						
Indirect:	0	1	1	0	0	0	1	0	1	See Section 4.1						

Description The contents of the addressed data memory location are subtracted from the upper 16 bits of the accumulator. The 16 low-order bits of the accumulator are unaffected. The result is stored in the accumulator.

The SUBH instruction can be used for performing 32-bit arithmetic.

Words 1

Cycles 1

Example

```
SUBH    DAT33    ;(DP = 0)
or
SUBH    *        ;If current auxiliary register contains 33.
```

	Before Instruction		After Instruction
Data Memory 33	4h	Data Memory 33	4h
ACC	0A0013h	ACC	60013h

Subtract from Low Accumulator with Sign-Extension Suppressed

Syntax

Direct: [<label>] SUBS <dma>
Indirect: [<label>] SUBS {*|*+|*-}[,<next ARP>]

Operands

0 ≤ dma ≤ 127
ARP = 0 or 1

Execution

(PC) + 1 → PC
(ACC) - (dma) → ACC
Affects OV; affected by OVM.

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	0	0	0	1	1	0	Data Memory Address						
Indirect:	0	1	1	0	0	0	1	1	1	See Section 4.1						

Description

The contents of the addressed data memory location are subtracted from the accumulator with sign-extension suppressed. The data is treated as a 16-bit unsigned number, rather than a two's-complement number. The accumulator behaves as a signed number.

Words

1

Cycles

1

Example

```
SUBS    DAT2    ;(DP = 0)
```
or
```
SUBS    *       ;If current auxiliary register contains 2.
```

	Before Instruction		After Instruction
Data Memory 2	0F003h	Data Memory 2	0F003h

	Before Instruction		After Instruction
ACC	0F105h	ACC	102h

Syntax

Direct: [<label>] TBLR <dma>
Indirect: [<label>] TBLR {*|*+|*-}[,<next ARP>]

Operands 0 ≤ dma ≤ 127
ARP = 0 or 1

Execution (PC) + 1 → TOS
(ACC(11-0)) → PC
(pma) → dma
Modify AR(ARP) and ARP as specified
(TOS) → PC

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	0	0	1	1	1	0	Data Memory Address						
Indirect:	0	1	1	0	0	1	1	1	1	See Section 4.1						

Description The TBLR instruction transfers a word from a location in program memory to a data memory location specified by the instruction. The program memory address is defined by the low-order 12 bits of the accumulator. For this operation, a read from program memory is performed, followed by a write to data memory. The contents of the lowest stack location are lost when using TBLW.

The TBLR instruction is useful for reading coefficients that have have beer stored in program ROM, or time-dependent data stored in RAM.

Words 1

Cycles 3

Example

```
TBLR    DAT6    ;(DP = 0)
or
TBLR    *       ;If current auxiliary register contains 6.
```

	Before Instruction		After Instruction
ACC	9h	ACC	9h
Program Memory 9	306h	Program Memory 9	306h
Data Memory 6	75h	Data Memory 6	306h
Stack	71h 48h 16h 80h	Stack	71h 48h 16h 16h

TBLW Table Write

Syntax

Direct: [<label>] TBLW <dma>
Indirect: [<label>] TBLW {*|*+|*-}[,<next ARP>]

Operands

0 ≤ dma ≤ 127
ARP = 0 or 1

Execution

(PC) + 1 → TOS
(ACC(11-0)) → PC
(dma) → pma
Modify AR(ARP) and ARP as specified
(TOS) → PC

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	1	1	1	0	1	0	Data Memory Address						
Indirect:	0	1	1	1	1	1	0	1	1	See Section 4.1						

Description

The TBLW instruction transfers a word in data memory to program memory. The data memory address is specified by the instruction, and the program memory address is specified by the lower 12 bits of the accumulator. A read from data memory is followed by a write to program memory to complete the instruction. The contents of the lowest stack location are lost when using TBLW.

Note that the TBLW and OUT instructions use the same external signals and thus cannot be distinguished when writing to program memory addresses 0 through 7.

Words 1

Cycles 3

Example

```
TBLW   DAT5    ;(DP = 0)
or
TBLW   *       ;If current auxiliary register contains 5.
```

	Before Instruction		After Instruction
Data Memory 5	4339h	Data Memory 5	4339h
Program Memory 8	306h	Program Memory 8	4339h
ACC	8h	ACC	8h
Stack	34h 23h 11h 97h	Stack	34h 23h 11h 11h

Exclusive-OR with Low Accumulator XOR

Syntax

Direct: [<label>] XOR <dma>
Indirect: [<label>] XOR {*|*+|*-}[,<next ARP>]

Operands

0 ≤ dma ≤ 127
ARP = 0 or 1

Execution

(PC) + 1 → PC
(ACC(15-0)).XOR.dma → ACC(15-0)
(ACC(31-16)) → ACC(31-16)

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	1	1	0	0	0	0	Data Memory Address						
Indirect:	0	1	1	1	1	0	0	0	1	See Section 4.1						

Description

The low half of the accumulator is exclusive-ORed with the contents of the addressed data memory location. The upper half of the accumulator is not affected by this instruction.

The XOR instruction is useful for toggling or setting bits of a word for high-speed control. In addition, the one's complement of a word can be found by exclusive-ORing it with all ones.

Words

1

Cycles

1

Example

```
XOR    DAT127  ;(DP = 0)
or
XOR    *       ;If current auxiliary register contains 127.
```

	Before Instruction		After Instruction
Data Memory 127	0F0F0h	Data Memory 127	0F0F0h
ACC	12345678h	ACC	1234A688h

Syntax [<label>] ZAC

Operands None

Execution
(PC) + 1 → PC
0 → ACC

Encoding

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1

Description The contents of the accumulator are replaced with zero.

Words 1

Cycles 1

Example

```
ZAC
```

	Before Instruction		After Instruction
ACC	0A5A5A5A5h	ACC	0h

ZALH Zero Low Accumulator and Load High Accumulator ZALH

Syntax

Direct: [<label>] ZALH <dma>
Indirect: [<label>] ZALH {*|*+|*-}[,<next ARP>]

Operands 0 ≤ dma ≤ 127
ARP = 0 or 1

Execution (PC) + 1 → PC
0 → ACC(15-0)
(dma) → ACC(31-16)

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	0	0	1	0	1	0	Data Memory Address						
Indirect:	0	1	1	0	0	1	0	1	1	See Section 4.1						

Description ZALH loads a data memory value into the high-order half of the accumulator. The low-order bits of the accumulator are zeroed.

ZALH is useful for 32-bit arithmetic operations.

Words 1

Cycles 1

Example

```
ZALH     DAT3     ;(DP = 0)
```

or

```
ZALH     *        ;If current auxiliary register contains 3.
```

	Before Instruction		After Instruction
Data Memory 3	3F01h	Data Memory 3	3F01h
ACC	77FFFFh	ACC	3F010000h

ZALS Zero Accumulator, Load Low Accumulator with Sign-Extension Suppressed ZALS

Syntax

Direct: [<label>] ZALS <dma>
Indirect: [<label>] ZALS {*|*+|*-}[,<next ARP>]

Operands 0 ≤ dma ≤ 127
ARP = 0 or 1

Execution (PC) + 1 → PC
0 → ACC(31-16)
(dma) → ACC(15-0)

Encoding

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct:	0	1	1	0	0	1	1	0	0	Data Memory Address						
Indirect:	0	1	1	0	0	1	1	0	1	See Section 4.1						

Description The contents of the addressed data memory location are loaded into the 16 low-order bits of the accumulator. The upper half of the accumulator is zeroed. The data is treated as a 16-bit unsigned number rather than a two's-complement number. Therefore, there is no sign-extension with this instruction.

ZALS is useful for 32-bit arithmetic operations.

Words 1

Cycles 1

Example

```
ZALS    DAT1    ;(DP = 0)
or
ZALS    *       ;If current auxiliary register contains 1.
```

	Before Instruction		After Instruction
Data Memory 1	0F7FFh	Data Memory 1	0F7FFh
ACC	7FF00033h	ACC	0F7FFh

Section 5

Software Applications

The use of various key software-related processor and instruction set features along with assembly language coding examples is explained in this section. TMS320C1x (first-generation TMS320) instructions are tailored to digital signal processing tasks, providing a single-cycle multiply, scaling, convolution, overflow management, and many other features. There is also instruction set support for logical and arithmetic operations.

More information about specific applications can be found in the book, *Digital Signal Processing Applications with the TMS320 Family* (literature number SPRA012A). The DSP Software Library contains the major DSP routines and application algorithms presented in the applications book. The TMS320 DSP Bulletin Board Service provides access to code updates and new application reports as they become available. See Appendix E for information about the software library and bulletin board.

Major topics discussed in this section are listed below and on the next page.

- Processor Initialization (Section 5.1 on page 5-3)
- Interrupt Management (Section 5.2 on page 5-7)
  Interrupt service routines
  $\overline{\text{BIO}}$ polling
  Context switching
- Program Control (Section 5.3 on page 5-16)
  Software stack expansion
  Subroutine calls
  Addressing and loop control with auxiliary registers
  Computed GOTOs
- Memory Management (Section 5.4 on page 5-23)
  Moving data
  Moving constants into data memory
- Logical and Arithmetic Operations (Section 5.5 on page 5-29)
  Bit manipulation
  Overflow management
  Scaling
  Convolution operations
  Multiplication, division, and addition
  Floating-point arithmetic

- Application-Oriented Operations (Section 5.6 on page 5-42)
  Companding
  FIR/IIR filtering
  Adaptive filtering
  Fast Fourier Transforms (FFT)
  PID control
  Selftest routines.

5.1 Processor Initialization

Prior to the execution of a digital signal processing algorithm, it is necessary to initialize the processor. Generally, initialization takes place anytime the processor is reset.

When reset is activated by applying a low level to the $\overline{\text{RS}}$ (reset) input for a minimum of five cycles, the TMS320C1x terminates program execution and forces the program counter (PC) to zero. Program memory location 0 normally contains a B (branch) instruction in order to direct program execution to the system initialization routine following the reset. The hardware reset also initializes various registers and status bits.

After reset, the processor should be initialized through software. The initialization routine should set up operational modes, memory pointers, interrupts, and the remaining functions necessary to meet system requirements. This section describes how to configure the TMS320C1x devices after reset and provides code for processor initialization.

5.1.1 TMS32010/C10/C15/E15 Initialization

To configure the TMS32010/C10/C15/E15 processor after reset, the following internal functions should be initialized:

- Interrupt structure
- Overflow mode control (OVM)
- Auxiliary registers and auxiliary register pointer (ARP)
- Data memory page pointer (DP).

Note that the OVM (overflow mode) bit, INTM (interrupt mode) bit, auxiliary register pointer (ARP), and data memory page pointer (DP) are not initialized by reset.

Example 5-1 shows coding for initializing the TMS32010/C10/C15/E15 to the following machine state, in addition to the initialization performed during the hardware reset:

- Interrupt enabled
- Overflow mode (OVM) disabled
- Data memory page pointer (DP) set to zero
- Auxiliary register pointer (ARP) set to zero
- Internal memory filled with zeros.

Example 5-1. TMS32010/C10/C15/E15 Processor Initialization

```
       .title   'PROCESSOR INITIALIZATION'
       .def     RESET,INT
       .ref     ISR
*
* PROCESSOR INITIALIZATION.
* RESET AND INTERRUPT VECTOR SPECIFICATION.
*
      .text
RESET  B    INIT    ; RS BEGINS PROCESSING HERE
INT    B    ISR     ; INT BEGINS PROCESSING HERE
*
* THE BRANCH INSTRUCTION AT PROGRAM MEMORY LOCATION 0 DIRECTS
* EXECUTION TO BEGIN HERE FOR RESET PROCESSING THAT INITIAL-
* IZES THE PROCESSOR. WHEN RESET IS APPLIED, THE FOLLOWING
* CONDITIONS ARE ESTABLISHED FOR THE STATUS REGISTER:
*
*      OV OVM INTM 12 11 10 9 ARP 7 6 5 4 3 2 DP
* ST:  0   X   1    1  1  1 1  X  1 1 1 1 1 1 X
*
INIT   ROVM         ; DISABLE OVERFLOW MODE
       LDPK 0       ; POINT DP TO DATA PAGE 0
       LARK 0,255   ; SET LOOP COUNT FOR DATA MEM INIT TO
                    ; 143 FOR 32010 AND 255 FOR 320C15/17
*
* INTERNAL DATA MEMORY INITIALIZATION.
*
       ZAC          ; CLEAR THE ACCUMULATOR
       LARP 0       ; USE AR0 FOR POINTER AND LOOP CONTROL
LOOP   SACL *       ; CLEAR DATA MEMORY
       BANZ LOOP    ; CHECK IF DONE AND DECREMENT AR0
*
* THE PROCESSOR IS INITIALIZED. THE REMAINING APPLICATION-
* DEPENDENT PART OF THE SYSTEM SHOULD NOW BE INITIALIZED.
*
       EINT         ; ENABLE ALL INTERRUPTS
```

5.1.2 TMS320C17E17 Initialization

To configure the TMS320C17/E17 after reset, the following internal functions must be initialized:

- Interrupt structure
- Serial-port framing-pulse generation selection
- Serial-port connection
- Companding hardware
- Serial-port clock
- Auxiliary register pointer
- Data memory page pointer
- Overflow mode.

Two of the I/O ports are dedicated to the serial port and companding hardware, the operation of which is determined by the 32 bits of the system control register. Table 5-1 lists the control register bits with brief definitions.

Table 5-1. Control Register Bit Definitions

CR BIT #	DEFINITION
	PORT 0
CR3 - CR0	Interrupt flags
CR7 - CR4	Interrupt mask bits
CR8	Port 1 configuration control
CR9	External framing enable for serial port transfers
CR10	XF external logic output flag latch
CR11	Serial port companding mode select
CR13 - CR12	Companding hardware enable
CR14	A-law/µ-law conversion select
CR15	Serial clock (SCLK) control
PORT 1	
CR23 - CR16	Frame counter modulus
CR27 - CR24	Serial clock (SCLK) prescale control (divide ratios)
CR28	FR pulse-width control
CR30 - CR29	I/O control
CR31	Reserved for future expansion (set to 0)

Example 5-2 shows coding for initializing the TMS320C17/E17 serial-port and companding hardware for interface to a codec. The following machine state is loaded:

- Set the lower control register bit 8 (CR8) to enable port 1 to access the upper control register. To insure safe system operation, SCLK should be left as an input to the device (CR15 set to logic 1). This prevents any invalid serial-port timing during the initialization routine. The value loaded into the lower control register to accomplish this is 0B988h.

- The upper control register is set as follows:
  - Long FR pulse (variable data-rate selected)
  - SCLK divide ratio of 10
  - FR frequency at SCLK/256 for an 8-kHz framing pulse
  - The value 7CFEh loaded into the upper control register.

  Note that the data operand of the upper control register is set at 7CFEh. This selects two's-complement companding for the serial port and 16-bit length coprocessor mode (i.e., for interface to 16-bit processors). When two's-complement companding is used, there must be at least one instruction between an OUT instruction to the serial port transmit register and an IN instruction from the serial port receive register.

- The lower control register is then configured as follows:
  - Interrupt flags cleared
  - Active FR interrupt enabled. (The FR interrupt flag will be generated independent of the enable condition to the serial port.)
  - Port transfers enabled by active FR
  - Serial companding mode selected (see Section 5.6.1)
  - Companding hardware enabled
  - µ-law conversion selected
  - SCLK selected as an output
  - The value 3888h now loaded into the lower control register.

Note that the interrupt flags are flip-flops. Writing a one to an interrupt flag clears it and sets the corresponding flag to zero; i.e., a write to the flags affects the clear or reset input of the flip-flops.

Example 5-2. TMS320C17/E17 Processor Initialization

```
* A BRANCH INSTRUCTION AT PROGRAM MEMORY LOCATION 0 DIRECTS
* PROCESSOR EXECUTION HERE.  THE CONTROL REGISTER VALUES ARE
* STORED IN ROM STARTING AT LOCATION 4.  THESE VALUES ARE
* THEN READ INTO RAM FOR THE OUT INSTRUCTIONS TO THE CONTROL
* REGISTER.  MEMORY LOCATIONS SET1-SET3 AND ONE ARE LOCATED
* ON RAM PAGE 1. THE PROGRAM MEMORY LOCATION HAS A BRANCH TO
* THE INTERRUPT SERVICE ROUTINE.
*
       .def  RESET,INT,INIT
       .ref  ISR
*
ONE    .set  1      ; CONSTANT ONE
SET1   .set  2      ; LOWER CONTROL REGISTER
SET2   .set  3      ; UPPER CONTROL REGISTER
SET3   .set  4      ; LOWER CONTROL REGISTER
*
* PROCESSOR INITIALIZATION.
* RESET AND INTERRUPT VECTOR SPECIFICATION.
*
      .text
RESET  B     INIT   ; RS BEGINS PROCESSING HERE
INT    B     ISR    ; INT BEGINS PROCESSING HERE
TABLE  .word 0B988h ; CONTROL REGISTER DATA
       .word 7CFEh
       .word 3888h
*
INIT   DINT         ; DISABLE INTERRUPTS
       SOVM         ; SET OVERFLOW MODE
       LARP  0      ; USE AUXILIARY REGISTER 0
       LDPK  1      ; WORK IN RAM PAGE 1
       LACK  1      ; ACC = 1
       SACL  ONE    ; STORE 1 IN MEMORY LOCATION ONE
       LACK  TABLE  ; START AT LOCATION 4
       TBLR  SET1   ; READ VALUE 0B988h TO RAM
       ADD   ONE,0  ; INCREMENT ADDRESS
       TBLR  SET2   ; READ VALUE 1CFEh TO RAM
       ADD   ONE,0  ; INCREMENT ADDRESS
       TBLR  SET3   ; READ VALUE 3888h TO RAM
       OUT   SET1,0 ; CONFIGURE LOWER CONTROL REGISTER
       OUT   SET2,1 ; CONFIGURE UPPER CONTROL REGISTER
       OUT   SET3,0 ; CONFIGURE LOWER CONTROL REGISTER
       LDPK  0      ; RESET RAM PAGE TO 0
*
* THE PROCESSOR IS INITIALIZED.  THE REST OF THE SYSTEM THAT
* IS APPLICATION-DEPENDENT SHOULD BE INITIALIZED BEFORE THE
* EINT INSTRUCTION.
*
       EINT         ; ENABLE INTERRUPTS
```

5.2 Interrupt Management

The interrupt function allows the current process to be suspended in order to perform a more critical function. On the TMS32010/C10/C15/E15, processor execution may be suspended on a high-priority basis by using the $\overline{\text{INT}}$ pin. Otherwise, a lower priority interrupt can be serviced by using a software ($\overline{\text{BIO}}$) polling technique.

The TMS320C17/E17 has four interrupts maskable via the system control register. These interrupts are synchronized and multiplexed into the master interrupt circuitry and have the same priority. Software polling techniques are used to determine which input caused the interrupt when multiple interrupts are enabled.

Processing in the interrupt service routine (ISR) must assure that the processor context is saved before and during execution and restored when the routine is finished. Descriptions and examples of how to implement interrupt service routines, $\overline{\text{BIO}}$ polling, and context switching are provided in this section.

5.2.1 TMS32010/C10/C15/E15 Interrupt Service Routines

The TMS32010/C10 and TMS320C15/E15 devices provide one maskable interrupt ($\overline{\text{INT}}$). By using the $\overline{\text{INT}}$ pin, the processor's execution can be suspended at any point in the program except after a multiply instruction. The instruction following the MPY and MPYK instructions is always executed.

Interrupt processing on the TMS32010/C10/C15/E15 begins as follows:

1) The EINT (enable interrupt) instruction is executed, which sets the INTM (interrupt mode) bit to 0 so that an interrupt can be received.
2) When an interrupt occurs, the INTF (interrupt flag) bit is set to 1.

As interrupt servicing begins, the following sequence occurs automatically:

1) The interrupt is acknowledged, which clears the INTF (interrupt flag) bit to 0.
2) The INTM (interrupt mode) bit is set to 1 to disable further interrupts.
3) The current PC is pushed onto TOS (top of stack).
4) The new PC is set to 2.

During servicing of the interrupt, the following operations are commonly performed by the user in software:

1) Program memory address 2 will either have a service routine to save the context of the machine or a branch to the interrupt service routine.
2) The interrupt service routine is executed. The context of the machine can be saved and the source of the interrupt serviced. Then, the context is restored and the interrupts enabled prior to returning from the interrupt routine.
3) The EINT (enable interrupt) instruction is executed, which sets the INTM (interrupt mode) bit to 0.
4) The RET instruction is executed.

The hardware interrupt can be masked at critical points in the program with the DINT instruction. This sets the INTM (disable interrupt mode) bit to logic one. If an interrupt occurs while INTM equals one, the interrupt will not be serviced until the interrupts are enabled again. However, the INTF (interrupt flag) is set to one, and the interrupt is held pending. The interrupt will be serviced when the INTM bit is set to zero by executing the EINT instruction. If an interrupt is pending when an enable interrupt operation occurs, the interrupt is serviced after the execution of the instruction following the EINT instruction. This allows for a return instruction to be executed before an interrupt is acknowledged.

An interrupt-driven analog input channel can be implemented using the technique described and shown in Example 5-3. However, multiple-level data buffering will impact system I/O overhead. Analog systems supported by first-generation TMS320 devices usually have information bandwidths of less than 20 kHz. The desired sample rate can be generated by dividing the CLKOUT signal from the TMS320. It is advisable to provide at least a one-level data buffer to ensure the integrity of the data read by the processor. If an 8-kHz sample rate is used (for example), the system must then respond to an analog interrupt every 125 µs. The percentage of I/O overhead incurred by this arrangement can be computed by determining the number of clock cycles that the TMS320 will spend in the interrupt routine servicing each sample and dividing by the number of clock cycles available between each sample. Example 5-3 shows a typical interrupt service routine. Note that the memory location flag (FLAG) contains a 1-bit flag to indicate that the required number of samples have been received.

Example 5-3. TMS32010/C10/C15/E15 Interrupt Service Routine

```
* THIS ROUTINE SERVICES AN EXTERNAL INTERRUPT. IT MAY BE
* LOCATED AT PROGRAM MEMORY LOCATION 2, OR A BRANCH AT
* LOCATION 2 DIRECTS PROGRAM EXECUTION HERE. THE ROUTINE
* READS DATA FROM AN EXTERNAL DEVICE (A/D CONVERTER). THE
* NUMBER OF SAMPLES OBTAINED ARE STORED IN MEMORY LOCATION
* COUNT. LIMIT IS THE NUMBER OF SAMPLES NEEDED. MEMORY
* LOCATION ONE CONTAINS THE CONSTANT 1. STATUS IS ALWAYS
* STORED ON DATA PAGE 1 WHEN USING DIRECT MEMORY ADDRESSING.
* ASSUME AR0 POINTS TO THE NEXT EMPTY LOCATION IN THE SAMPLE
* BUFFER.
*
ADC    .set 0         ; ASSIGN PA0 TO A/D CONVERTER
STATUS .set 0         ; STATUS REGISTER STORAGE ON PAGE 1
ACCL   .set 1         ; ASSIGN MEM LOCATION TO SAVE STATUS/ACC
ACCH   .set 2
SAMP   .set 3         ; STORE INPUT DATA HERE
COUNT  .set 4         ; COUNT # OF SAMPLES HERE
FLAG   .set 5         ; ASSIGN MEM LOCATION TO FLAG
LIMIT  .set 32        ; ASSIGN TOTAL # OF SAMPLES REQUIRED
*
       .text
ISR    SST  STATUS    ; SAVE STATUS
       LDPK 1         ; USE DATA PAGE 1
       SACL ACCL      ; SAVE ACCUMULATOR LOW
       SACH ACCH      ; SAVE ACCUMULATOR HIGH
       LARP 0         ; USE AR0
       IN   *-,ADC    ; READ FROM ADC
       LAC  COUNT     ; LOAD SAMPLE COUNTER
       ADD  ONE       ; INCREMENT
       SACL COUNT     ; STORE UPDATED COUNT
       LACK LIMIT
       SUB  COUNT     ; CHECK IF LIMIT EXCEEDED
       BGZ  OK
DONE   LACK 1
       SACL FLAG      ; YES --> SET FLAG
OK     ZALH ACCH      ; RESTORE ACCUMULATOR HIGH
       ADDS ACCL      ; RESTORE ACCUMULATOR LOW
       LST  STATUS    ; RESTORE STATUS
       EINT           ; ENABLE SUBSEQUENT INTERRUPTS
       RET
```

If the processor is using a 20-MHz clock, the number of available cycles between each sample is 625. The overhead required to service this system is 18/625 = 2.9 percent. This overhead burden can be reduced by using a FIFO (first in, first out) to buffer the data. In this case, the TMS320 need only be interrupted when the buffer has filled. If a 16-level FIFO is used in the example above, this interrupt will occur every 2 ms, and the overhead burden will be reduced to about 0.5 percent.

If two different kinds of devices are being serviced by the same interrupt routine, the $\overline{\text{BIO}}$ pin can be used to determine which device needs to be serviced (see Section 5.2.3 for $\overline{\text{BIO}}$ polling).

5.2.2 TMS320C17/E17 Interrupt Service Routines

The TMS320C17/E17 has four maskable interrupts: $\overline{\text{EXINT}}$, $\overline{\text{FSR}}$, $\overline{\text{FSX}}$, and FR. The interrupts are maskable via the system control register bits CR7-CR4. Bits CR3-CR0 serve as the interrupt flags for the four interrupts. An active signal on any of these pins sets the corresponding interrupt flag to one. Since all four interrupts activate a single master interrupt flag, the interrupt service routine (ISR) should poll all four interrupt flags and check for the corresponding interrupt source. The ISR may also need to poll the individual mask bits (CR7-CR4) before recognizing the interrupt flag.

Interrupt processing on the TMS320C17/E17 begins as follows:

1) The EINT (enable interrupt) instruction is executed, which sets the INTM (interrupt mode) bit to 0 so that an interrupt can be received.
2) When an interrupt occurs, the INTF (interrupt flag) bit is set to 1.

As interrupt servicing begins, the following sequence occurs automatically:

1) The interrupt is acknowledged, which clears the INTF (interrupt flag) bit to 0.
2) The INTM (interrupt mode) bit is set to 1 to disable further interrupts.
3) The current PC is pushed onto TOS (top of stack).
4) The new PC is set to 2.

During servicing of the interrupt, the following operations are commonly performed by the user in software:

1) Program memory address 2 will either have a service routine to save the context of the machine or a branch to the interrupt service routine.
2) The interrupt service routine is executed. The context of the machine may be saved and restored later if required. The following can be used to select which interrupt to service:
   a) Use software polling techniques to determine which one of the four flags has been set in the control register.
   b) Check for corresponding mask bits before proceeding (optional).
   c) Clear that flag (reset by writing a 1) and service the source of that flag. There must be an interval of at least four clock cycles after the flag has been set before clearing it. On the $\overline{\text{EXINT}}$ flag, the interrupt source must have been taken away for four cycles before the interrupt flag can be cleared.

All interrupts are synchronized and multiplexed into the master interrupt circuitry and have the same priority. However, interrupt priorities in polling the interrupt flags can be established by the user. The ISR should clear the interrupt flag before executing an EINT instruction or enabling the interrupts. Note that writing a one to an interrupt flag will clear it, i.e., set the corresponding flag to zero. If the interrupt condition persists when an attempt is made to clear the flag, that interrupt flag will remain set. This condition is only applicable to $\overline{\text{EXINT}}$ or its equivalent in coprocessor port mode. In the coprocessor mode on the TMS320C17/E17, the $\overline{\text{BIO}}$ and $\overline{\text{EXINT}}$ lines cannot be driven externally, but are reserved for transfers to/from the coprocessor port. An example interrupt service routine for a system with three active interrupts enabled is given in Example 5-4. Polling is also included in the code example.

Example 5-4. TMS320C17/E17 Interrupt Service Routine

```
* THIS ROUTINE MAY BE LOCATED AT PROGRAM MEMORY LOCATION 2,
* OR A BRANCH INSTRUCTION AT LOCATION 2 DIRECTS PROGRAM
* EXECUTION HERE.  MEMORY LOCATION ONE CONTAINS THE
* CONSTANT 1. STATUS IS ALWAYS STORED ON DATA PAGE 1 WHEN
* DIRECT MEMORY ADDRESSING IS USED.
*
* RECV IS THE SERVICE ROUTINE FOR THE RECEIVE INTERRUPT.
* XINT IS THE SERVICE ROUTINE FOR THE EXTERNAL INTERRUPT.
* TRANS IS THE SERVICE ROUTINE FOR THE TRANSMIT INTERRUPT.
*
       .def  ISR,RECV,XINT
       .ref  TRANS
*
STATUS .set  0
ACCL   .set  1          ; ASSIGN MEM LOCATION TO SAVE STATUS/ACC
ACCH   .set  2
RBUF   .set  3          ; STORE RECEIVE DATA HERE
CREG   .set  4          ; TEMP LOCATION TO STORE CONTROL REG
*
       .text
ISR    SST   STATUS     ; SAVE STATUS
       LDPK  1          ; USE DATA PAGE 1
       SACL  ACCL       ; SAVE LOW ACCUMULATOR
       SACH  ACCH       ; SAVE HIGH ACCUMULATOR
*
* THIS ROUTINE CHECKS FOR THREE ACTIVE INTERRUPTS OCCURRING
* AND SERVICES THEM ACCORDINGLY. IT IS ASSUMED THAT ONE OF
* THREE IS THE SOURCE OF THE INTERRUPT. AFTER AN INTERRUPT
* FLAG IS SET, IT MUST BE RESET BY THE INTERRUPT SERVICE
* ROUTINE TO AVOID BEING INTERRUPTED AGAIN ON THE RETURN
* FROM THE SUBROUTINE.
*
       IN    CREG,PA0   ; READ LOWER CONTROL REGISTER
       LAC   ONE,0      ; LOAD INT INTERRUPT MASK
       AND   CREG       ; INT FLAG SET?
       BNZ   XINT       ; GO TO INT SERVICE ROUTINE
       LAC   ONE,2      ; LOAD FSX INTERRUPT MASK
       AND   CREG       ; FSX FLAG SET?
       BNZ   TRANS      ; GO TO TRANSMIT SERVICE ROUTINE
*
* INTERRUPT MUST BE FSR.
*
RECV   LACK  0Fh        ; SET ALL INTERRUPT FLAGS IN CREG
       OR    CREG
       SACL  CREG
       LACK  0Bh        ; ZERO ALL INTERRUPT FLAGS EXCEPT FSR
       XOR   CREG
       SACL  CREG
       IN    RBUF,PA1   ; READ REC DATA FROM PORT 1
*
* RESTORE STATUS.
*
RESTOR OUT   CREG,PA0   ; RESTORE CNTL REG; CLEAR INTERRUPTS
       ZALH  ACCH       ; RESTORE HIGH ACCUMULATOR
       ADDS  ACCL       ; RESTORE LOW ACCUMULATOR
       LST   STATUS     ; RESTORE STATUS
       EINT             ; ENABLE INTERRUPTS
       RET
*
```

```
* INTERRUPT MUST BE COPROCESSOR EXINT.
*
XINT IN   CPBUF,PA5 ; READ LATCH DATA FROM PORT 5
     LACK 0Fh       ; SET ALL INTERRUPT FLAGS IN CREG
     OR   CREG
     SACL CREG
     LACK 0Eh       ; ZERO ALL INTERRUPT FLAGS EXCEPT EXINT
     XOR  CREG
     SACL CREG
     B    RESTOR    ; BRANCH TO RESTORE STATUS
```

5.2.3 $\overline{\text{BIO}}$ Polling

A low priority interrupt can be serviced by using $\overline{\text{BIO}}$ polling. The BIOZ instruction can be used to poll (or test) the $\overline{\text{BIO}}$ pin to see if a device needs to be serviced. This method allows a critical loop or set of instructions to be executed without a variation in execution time. Because the test for the $\overline{\text{BIO}}$ pin occurs at defined points in the program, context saves are minimal.

The $\overline{\text{BIO}}$ pin can be used to monitor the status of a peripheral. If the FIFO (first in, first out) full status line is connected to the $\overline{\text{BIO}}$ pin, the FIFO is serviced only when the FIFO is full. In the following code segment, the FIFO contains 16 data words. The $\overline{\text{BIO}}$ pin is tested after each time-critical function has been executed.

```
        BIOZ   SKIP
        CALL   SERVE
SKIP     .
         .
         .
```

The subroutine does not have to save the registers or the status, because a new procedure will be executed after the device is serviced, as shown below.

```
SERVE   LARK   AR0,15
        LARK   AR1,TABLE
LOOP    LARP   1
        IN     *+,PA0,AR0
        BANZ   LOOP
        RET
```

The FIFO must be serviced before another word is input or data may be lost. This fact determines the frequency at which the polling must take place.

5.2.4 Context Switching

Context switching, commonly required when processing a subroutine call or interrupt, may be quite extensive or simple, depending on system requirements such as the use made of the stack or auxiliary registers. Unless the interrupt service routine (ISR) is a simple I/O handler, the processing in the ISR generally must assure that the processor context is preserved during execution. The context must be saved before executing the routine itself and restored when the routine is finished. A common routine may be used to secure the context of the processor during interrupt processing.

The TMS320C1x program counter is stored automatically on the hardware stack. If there is any important information in the other TMS320C1x registers, such as the status or auxiliary registers, these must be saved by user software. A stack in data memory, identified by an auxiliary register, is useful for storing the machine state when processing interrupts.

During an interrupt, all registers except the P register can be saved and restored directly. However, the TMS320C1x devices have hardware protection against servicing an interrupt between an MPY or MPYK instruction and the following instruction. For this reason, it is advisable to follow the MPY and MPYK instructions with LTA, LTD, PAC, APAC, or SPAC instructions that transfer data from the P register to the accumulator.

Examples of saving and restoring the state of the TMS320C1x processor are given in Example 5-5 and Example 5-6. Auxiliary register 1 (AR1) is used in both examples as the stack pointer. As the stack grows, it expands into lower memory addresses. The registers saved are the ST status register, accumulator (ACC), P register, T register, all four levels of the hardware stack, and auxiliary registers AR0 and AR1.

The routines in Example 5-5 and Example 5-6 are protected against interrupts, allowing context switches to be nested. This is accomplished by the use of the MAR *- and MAR *+ instructions at the beginning of the context save and context restore routines, respectively. Note that the last instruction of the context save decrements AR1 while the context restore is completed with an additional increment of AR1. This prevents the loss of data if a context save or restore routine is interrupted.

Example 5-5. Context Save

```
            .title    'CONTEXT SAVE'
            .def      SAVE
            .text
      *
      * CONTEXT SAVE ON SUBROUTINE CALL OR INTERRUPT. ASSUME THAT
      * AR1 IS THE STACK POINTER AND AR1 = 128.
      *
      SAVE  LARP AR1      ; CHANGE POINTER TO AR1   AR1 = 128
            MAR  *-       ;                         AR1 = 127
      *
      * SAVE THE STATUS REGISTER.
      *
            SST  *-       ; ST  --> (127),          AR1 = 126
      *
      * SAVE THE ACCUMULATOR.
      *
            SACH *-       ; ACCH --> (126),         AR1 = 125
            SACL *-       ; ACCL --> (125),         AR1 = 124
      *
      * SAVE THE P REGISTER.
      *
      * THE P REGISTER CANNOT BE EASILY RESTORED FROM MEMORY. ON
      * TMS320C1X DEVICES, IT IS ASSUMED THAT THE MPY AND MPYK
      * INSTRUCTIONS HAVE BEEN FOLLOWED BY AN APAC, PAC, SPAC,
      * LTA, OR LTD INSTRUCTION. HENCE, SAVING THE ACCUMULATOR
      * HAS ALSO SAVED THE P REGISTER.
      *
      * SAVE THE T REGISTER.
      *
            MPYK 1        ; T --> P
            PAC           ; T --> ACC
            SACL *-       ; T --> (124),            AR1 = 123
      *
      * SAVE ALL FOUR LEVELS OF THE HARDWARE STACK.
      *
            POP           ; TOS      --> ACC,
            SACL *-       ; TOS  (4) --> (123),     AR1 = 122
            POP           ; STACK(3) --> ACC,
            SACL *-       ; STACK(3) --> (122),     AR1 = 121
            POP           ; STACK(2) --> ACC,
            SACL *-       ; STACK(2) --> (121),     AR1 = 120
            POP           ; BOS  (1) --> ACC,
            SACL *-       ; BOS  (1) --> (120),     AR1 = 119
      *
      * SAVE AUXILIARY REGISTERS.
      *
            SAR  AR0,*-   ; AR0 --> (119),          AR1 = 118
            SAR  AR1,*-   ; AR1 --> (118),          AR1 = 117
      *
      * SAVE IS COMPLETE.
```

Example 5-6. Context Restore

```
       .title   'CONTEXT RESTORE'
       .def     RESTOR
       .text
*
* CONTEXT RESTORE AT THE END OF A SUBROUTINE OR INTERRUPT.
* ASSUME THAT AR1 IS THE STACK POINTER AND AR1 = 117.
*
RESTOR LARP AR1     ; CHANGE POINTER TO AR1, AR1 = 117
       MAR  *+      ;                        AR1 = 118
*
* RESTORE AUXILIARY REGISTERS.
*
       LAR  AR1,*+ ; (118) --> AR1,          AR1 = 119
       LAR  AR0,*+ ; (119) --> AR0,          AR0 = 120
*
* RESTORE ALL FOUR LEVELS OF THE HARDWARE STACK.
*
       ZALS *+      ; (120) --> ACC,         AR1 = 121
       PUSH         ; (120) --> BOS  (1),
       ZALS *+      ; (121) --> ACC,         AR1 = 122
       PUSH         ; (121) --> STACK(2),
       ZALS *+      ; (122) --> ACC,         AR1 = 123
       PUSH         ; (122) --> STACK(3),
       ZALS *+      ; (123) --> ACC,         AR1 = 124
       PUSH         ; (123) --> TOS  (4),
*
* RESTORE THE T REGISTER.
*
       LT   *+      ; (124) --> T,           AR1 = 125
*
* RESTORE THE ACCUMULATOR.
*
       ZALS *+      ; (125) --> ACCL,        AR1 = 126
       ADDH *+      ; (126) --> ACCH,        AR1 = 127
*
* RESTORE THE STATUS REGISTER.
*
       LST  *+      ; (127) -> ST,           AR1 = 128
*
* RESTORE IS COMPLETE.
*
       EINT         ; ENABLE INTERRUPTS
       RET          ; RETURN TO CALLING ROUTINE
```

5.3 Program Control

To facilitate the use of the TMS320C1x in general-purpose high-speed processing, a variety of instructions are provided for software stack expansion, implementation of subroutine calls, addressing and loop control with auxiliary registers, and external branch control. Descriptions and examples of how to use these features are given in this section.

5.3.1 Software Stack Expansion

The TMS320C1x has a 12-bit Program Counter (PC) and a four-level hardware stack for PC storage. Provisions have been made on the TMS320C1x for extending the hardware stack into data memory. This is useful for deep subroutine nesting or stack overflow protection.

The hardware stack is accessible via the accumulator using the PUSH and POP instructions. The PUSH instruction pushes the 12 LSBs of the accumulator onto the top of stack (TOS). The POP instruction pops the TOS into the 12 LSBs of the accumulator. Following the POP instruction, the TOS can be moved into data memory by storing the low-order accumulator word (SACL instruction). This allows expansion of the stack into the data RAM. From data RAM, it can easily be copied into off-chip program RAM using the TBLW instruction. In this way, the stack can be expanded to very large levels.

When the stack has four values stored on it and one or more values are to be put on the stack before any other values are popped off, a subroutine can be used to perform software stack expansion. Such a routine is illustrated in Example 5-7. In this example, the main program stores the stack starting location in memory in the auxiliary register and indicates to the subroutine whether to push data from memory onto the stack or pop data from the stack to memory. If a zero is loaded into the accumulator before calling the subroutine, the subroutine pushes data from memory to the stack. If a one is loaded into the accumulator, the subroutine pops data from the stack to memory.

A CALL instruction should be used to initiate execution of the software stack expansion routine. Since the CALL instruction uses the stack to save the program counter, the subroutine pops this value into the accumulator and saves it in a memory location. Then at the end of the subroutine, this value is reloaded into the accumulator, and the main program is reentered using the RET instruction. This prevents the calling routine program counter from being stored into a memory location. The subroutine in Example 5-7 uses the BANZ (branch on auxiliary register not zero) instruction to control all of its loops.

Example 5-7. Software Stack Expansion

```
* THIS ROUTINE EXPANDS THE STACK WHILE LETTING THE MAIN
* PROGRAM DETERMINE WHERE TO STORE THE STACK CONTENTS OR
* FROM WHERE TO RECOVER THEM.
*
LOC1    .set 0
*
        .text
STACK   LARK AR1,3       ; LOAD COUNTER
        LDPK 1           ; USE PAGE 1
        BNZ  PO          ; IF POPD IS NEEDED, GOTO PO
        POP              ; LOAD PC INTO ACCUMULATOR
        SACL LOC1        ; STORE PC AT MEM LOCATION LOC1
*
        LARP 0           ; USE AR0
        LAC  *+,AR1      ; LOAD ACCUMULATOR INTO MEMORY
        PUSH             ; PUT MEMORY ON STACK
        BANZ P           ; BRANCH TO P UNTIL STACK IS FULL
        LAC  LOC1        ; LOAD PC INTO ACCUMULATOR
        PUSH             ; PUT RETURN ADDRESS ON STACK
        RET              ; RETURN TO MAIN PROGRAM
*
PO      POP              ; LOAD PC INTO ACCUMULATOR
        SACL LOC1        ; SAVE PC INTO MEMORY
        LARP AR0         ; USE AR0
        MAR  *-          ; ALIGN STACK POINTER
*
PO1     LARP 0           ; USE AR0
        POP              ; PUT STACK IN ACCUMULATOR
        SACL *-,0,AR1    ; STORE STACK IN MEMORY
        BANZ PO1         ; BRANCH TO PO1 UNTIL SAVED
        MAR  *+          ; REALIGN STACK POINTER
        LAC  LOC1        ; LOAD ACCUMULATOR WITH PC
        PUSH             ; PUT RETURN ADDRESS ON STACK
        RET              ; RETURN TO MAIN PROGRAM
```

5.3.2 Subroutine Calls

When a subroutine call is made using the CALL or CALA instruction, the current contents of the program counter are stored on the top of the stack. At the end of the subroutine, a RET (return from subroutine) instruction pops the top of the stack to the program counter. The program then resumes execution at the instruction following the subroutine call.

In two circumstances, a level of stack must be reserved for the machine's use. First, the TBLR and TBLW instructions use one level of stack. Second, when interrupts are enabled, the PC is saved on the stack during the interrupt routine. If a system is designed to use both interrupts and a TBLR or TBLW instruction, only two levels of stack are available for nesting subroutine calls.

Subroutine calls can be nested deeper than two levels if the return address is removed from the stack and saved in data memory. The POP instruction moves the top of stack (TOS) into the accumulator and pops the stack up one level. The return address can then be stored in data memory until the end of the subroutine when it is put back into the accumulator. The PUSH instruction pushes the stack down one level and then moves the accumulator onto the TOS. Therefore, when the RET instruction is executed, the PC is updated with the return address. This procedure allows a second subroutine to be called inside the first subroutine without using another level of stack.

The POP and PUSH instructions can also be used to pass arguments to a subroutine. The .word directives following the subroutine call can be used to create a list of constants and/or variables to be passed to the subroutine. After the subroutine is called, the TOS points to the list of arguments following the CALL instruction. By moving the argument pointer from the TOS to the accumulator, the list of arguments can be read into data memory using the TBLR instruction. Between each TBLR instruction, the accumulator must be incremented by one to point to the next argument in the list. To create the return address, the argument pointer is incremented past the last element in the argument list. The PUSH instruction moves the return address onto the TOS, and the RET instruction updates the PC. Example 5-8 illustrates a call that passes two arguments to a subroutine.

Example 5-8. Two Arguments Passed to a Subroutine

```
* CLEAR BITS
*
* THIS ROUTINE CLEARS THE BITS OF A DATA WORD DESIGNATED BY
* A MASK. THE BITS SET TO ONE IN THE MASK INDICATE THE BITS
* IN THE DATA WORD TO BE CLEARED. ALL OTHER BITS REMAIN
* UNCHANGED. LOCATION ONE CONTAINS THE CONSTANT 1. MINUS
* CONTAINS A MASK INVERTER, -1 OR 0FFFFh. TWO ARGUMENTS ARE
* PASSED TO THIS SUBROUTINE. THE CALLING SEQUENCE IS AS
* FOLLOWS:
*
*       CALL   CBITS
*       .word  VALUE     ; 1ST ARGUMENT = ADDRESS OF DATA WORD
*       .word  0081h     ; 2ND ARGUMENT = MASK
*
STATUS .set    0         ; STORE STATUS REGISTER HERE
XR0    .set    126       ; TEMPORARY LOCATIONS
XR1    .set    127
*
CBITS  SST     STATUS    ; SAVE STATUS
       LDPK    0         ; USE DATA PAGE 0
       SAR     AR0,XR0   ; SAVE AR0 IN TEMPORARY LOCATION
*
       POP               ; GET ADDRESS OF 1ST ARGUMENT IN ACC
       TBLR    XR1       ; STORE 1ST ARGUMENT IN TEMP LOCATION
       LAR     AR0,XR1   ; PUT 1ST ARGUMENT INTO AR0
       ADD     ONE       ; POINT TO 2ND ARGUMENT
       TBLR    XR1       ; 2ND ARGUMENT = MASK
       ADD     ONE       ; POINT TO RETURN ADDRESS
       PUSH              ; PUT RETURN ADDRESS ON TOS
*
       LARP    0
       LAC     XR1       ; LOAD MASK INTO ACCUMULATOR
       XOR     MINUS     ; INVERT MASK
       AND     *         ; CLEAR BITS
       SACL    *         ; STORE MODIFIED VALUE
*
       LAR     AR0,XR0   ; RESTORE AR0
       LDPK    1         ; USE DATA PAGE 1
       LST     STATUS    ; RESTORE STATUS REGISTER
       RET               ; RETURN TO MAIN PROGRAM
```

Hardware stack allocation involves allocating the usage of the various stack levels for interrupts, subroutine calls, pipelined instructions, and the emulator (XDS). The TMS320C1x disables all interrupts when taking an interrupt trap. If interrupts are enabled more than one instruction before the return of the interrupt service routine, the routine can also be interrupted, thus using another level of the hardware stack. This should be taken into consideration when managing the use of the stack.

When nesting subroutine calls, each call uses a level of the stack. The number of levels used by interrupts must be considered as well as the depth of the nesting of subroutines. Two possible allocations of the hardware stack levels are:

- 1 level reserved for interrupt service routines (ISR)
- 3 levels available for subroutine calls.

or:

- 1 level reserved for interrupt service routines (ISR)
- 2 levels available for subroutine calls
- 1 level available for TBLR/TBLW instructions.

5.3.3 Addressing and Loop Control with Auxiliary Registers

The two auxiliary registers on the TMS320C1x can be used either as pointers for indirect addressing or as loop counters. In the indirect addressing mode, the auxiliary register pointer (ARP) is used to determine which auxiliary register is selected. The LARP instruction sets the ARP equal to the value of the immediate operand. The value of the ARP can also be changed in the indirect addressing mode; the ARP is updated after the instruction has been executed.

The contents of the auxiliary register are interpreted as a data memory address when the indirect addressing mode is used. A sequential list of data can easily be accessed in the indirect mode by using the autoincrement/decrement feature of the auxiliary registers. The auxiliary register can also be used as a 9-bit counter (see Section 3.4.5). The MAR (modify auxiliary register and pointer) instruction allows the auxiliary register selected by the ARP to be incremented or decremented without implementing any other operation in parallel.

Three instructions (LARK, LAR, and SAR) either load or store a value into an auxiliary register, independent of the value of the ARP. The first operand in each of these instructions determines which auxiliary register is to be either loaded or stored. This operand does not affect the value of the ARP for subsequent instructions.

Example 5-9 illustrates using an auxiliary register in the indirect addressing mode to input data into a block of memory.

Example 5-9. Auxiliary Register Indirect Addressing

```
* THIS ROUTINE USES AN AUXILIARY REGISTER IN THE INDIRECT
* ADDRESSING MODE TO INPUT DATA INTO A BLOCK OF MEMORY.
*
        LARK AR0,DATBLK ; INIT AR0 AS A POINTER TO DATBLK
*                       ; (AREA OF 8 WORDS IN DATA MEMORY)
        LARP 0          ; SELECT AR0
        LACK 8          ; INIT ACCUMULATOR AS A COUNTER
*
LOOP    IN   *+,PA0     ; INPUT DATA
        SUB  ONE        ; DECREMENT COUNTER (ONE = VALUE 1)
        BNZ  LOOP       ; REPEAT UNTIL COUNT = 0
```

An auxiliary register can also be used as a loop counter. The BANZ instruction tests and then decrements the auxiliary register selected by ARP. Because the test for zero occurs before the auxiliary register is decremented, the value loaded into the auxiliary register must be one less than the number of times the loop should be executed. The maximum number of loops that can be counted is 512, because only 9 bits of each auxiliary register are implemented as counters. A routine that inputs data and calculates a sum while the auxiliary register is used to count the number of loops is shown in Example 5-10. The accumulator contains the result.

Example 5-10. Auxiliary Register Loop Counting

```
* THIS ROUTINE USES AN AUXILIARY REGISTER TO COUNT THE
* NUMBER OF LOOPS.
*
        LARK AR0,3      ; INITIALIZE AR0 AS A COUNTER
        LARP 0          ; SELECT AR0
        ZAC             ; CLEAR ACCUMULATOR
*
LOOP    IN   DATA1,PA2  ; INPUT DATA VALUE
        ADD  DATA1      ; ADD DATA TO ACCUMULATOR
        BANZ LOOP       ; REPEAT LOOP FOUR TIMES
```

Both indirect addressing and loop counting can be performed at the same time to implement loops efficiently. If the data block is defined to start at location 0 in data memory, the same auxiliary register that is counting the number of loops can also be the pointer for indirect addressing, as shown below. Note that data locations 0 through 7 are loaded with input data.

```
        LARK AR0,7        ; AR0 POINTS TO END OF DATA BLOCK
LOOP    IN   *,PA2        ; INPUT DATA VALUE
        BANZ LOOP         ; REPEAT LOOP 8 TIMES
```

The data block does not have to start at zero if one auxiliary register is used for counting and the other register is used as a pointer. Example 5-11 illustrates how both auxiliary registers can be used at once.

Example 5-11. Auxiliary Register Pointing and Loop Counting

```
* THIS ROUTINE USES ONE AUXILIARY REGISTER FOR POINTING AND
* THE OTHER REGISTER FOR LOOP COUNTING.
*
        LARK AR0,7        ; INITIALIZE AR0 AS A COUNTER
        LARK AR1,DATBLK   ; AR0 POINTS TO START OF DATBLK
*                         ; (DATA MEMORY AREA)
        ZAC               ; CLEAR ACCUMULATOR
*
LOOP    LARP 1            ; POINT TO AR1
        ADD  *+,AR0       ; CALCULATE SUM OF DATA IN BLOCK
*                         ; POINT TO AR0
        BANZ LOOP         ; REPEAT LOOP 8 TIMES
```

5.3.4 Computed GOTOs

Processing may be executed in a time-dependent (interrupt-driven) or process-dependent (user-selected) way. Selecting the processing mode may depend on the result of a particular computation. A simple computed GOTO can be programmed in the TMS320C1x by using the CALA instruction. This instruction uses the contents of the accumulator as the direct address of the call. The address of the subroutine can be computed from a data value to determine which one of several routines will be executed. The return at the end of each of these routines causes program execution to resume with the instruction following the CALA command. Note that the CALA instruction uses a level of stack, because it is an indirect subroutine call, not just an indirect branch.

Example 5-12 illustrates how to compute a call to one of several routines. The subroutines are defined first, and then·a table of branches to each subroutine is created. The main part of the program inputs a data value of 0, 1, or 2. The appropriate address in the table is calculated in the accumulator. An indirect subroutine call causes the proper branch in the table to be executed.

Example 5-12. Computed GOTO

```
* THIS ROUTINE COMPUTES AND EXECUTES A SUBROUTINE CALL.
*
ONE    .set  126         ; STORE CONSTANT 1
VALUE  .set  127         ; VALUE READ FROM PORT 4
*
SUB1   IN    DAT1,PA0    ; INPUT DATA VALUE FROM PORT 0
       RET
*
SUB2   IN    DAT1,PA1    ; INPUT DATA VALUE FROM PORT 1
       RET
*
SUB3   IN    DAT1,PA2    ; INPUT DATA VALUE FROM PORT 2
       RET
*
TBL1   B     SUB1        ; CREATE TABLE OF BRANCHES TO EACH
       B     SUB2        ; SUBROUTINE DEFINED
       B     SUB3
*
START  LDPK  0           ; USE PAGE 0
       LACK  1           ; ACC = 1
       SACL  ONE         ; STORE 1 IN LOCATION ONE
       LT    ONE         ; LOAD T REGISTER WITH VALUE OF 1
       MPYK  TBL1        ; GET ADDRESS OF TABLE
       PAC
       IN    VALUE,PA4   ; INPUT DATA VALUE OF 0, 1, OR 2
       LT    VALUE       ; LOAD T REG WITH VALUE FROM PA4
       MPYK  2           ; CALCULATE OFFSET
       APAC
       CALA              ; GO TO DESIGNATED SUBROUTINE
       LAC   DAT1        ; RETURN HERE AFTER SUBROUTINE
        .
        .
        .
```

5.4 Memory Management

The TMS320C1x has a modified Harvard architecture in which the program and data memories reside in two separate spaces. Therefore, the next instruction fetch can occur while the current instruction is fetching data and executing the operation. The concept of the Harvard architecture increases the speed of the device, but it requires the use of instructions to transfer a word between data memory and program memory.

Data memory consists of 144/256 words of 16-bit on-chip RAM with all nonimmediate data operands residing within this RAM. Program memory consists of 1.5K/4K words of 16-bit on-chip ROM with 1524/4000 words reserved for program use. Only those devices with EPROM capability can access all 4096 words. Since there is no microprocessor mode of operation on the TMS320C17/E17, all program memory resides within the on-chip ROM.

The TMS320C1x uses three forms of data memory addressing: direct, indirect, and immediate. Direct addressing uses the seven bits of the instruction word concatenated with the data page pointer to form the data memory address. Indirect addressing uses the lower eight bits of the auxiliary registers as the data memory address. Immediate addressing uses part of the instruction word for data rather than data RAM.

The structure of the TMS320C1x memory map can vary for each application (see Section 3.4.4 for memory maps). Instructions are provided for moving data and for moving constants into data memory. Explanations and examples are provided in this section.

5.4.1 Moving Data

The DMOV (data move) instruction allows a data word to be written into the next higher memory location in a single cycle without affecting the accumulator. If variables are placed in consecutive locations, a DMOV instruction can be used to move each of the variables before the next calculation is performed. For example, when implementing a digital filter, the variables in the equation represent the inputs and outputs at discrete times. This type of data structure is typically implemented as a shift register when the data at time t is shifted to the position previously occupied by the data at time t-1. If consecutive addresses in data memory correspond to consecutive time increments, then shifts can be accomplished simply by using the DMOV instruction to move the data item at location d to that corresponding to d+1.

The LTD instruction combines the data move operation with the LTA (load T register and accumulate previous product) instruction operations, performing the three operations in parallel. The operand of the instruction is loaded into the T register; the operand is also written into the next higher memory location; and the P register is added to the accumulator. When using the LTD instruction, the order of the multiply and accumulate operations becomes important because the data is being moved while the calculation is being performed. The oldest input variable must be multiplied by its constant and loaded into the accumulator first. Then the input, which is one time-unit delay less, is multiplied and accumulated. This process is repeated until the entire equation has been computed.

Example 5-13 illustrates the use of the LTD instruction to move input variables in memory as the results are calculated.

Example 5-13. Moving Data Using the LTD Instruction

```
* THE FOLLOWING EQUATION WILL BE IMPLEMENTED TO DEMONSTRATE
* THE USE OF THE LTD INSTRUCTION. AT THE END OF THE SUB-
* ROUTINE, LOCATION X1 IS AVAILABLE TO INPUT THE NEW SAMPLE.
*
*               Y = A*X3 + B*X2 + C*X1
*
* WHERE A, B, C, X1, X2, AND X3 ARE VALUES STORED AT THESE
* ADDRESSES.
*
X1      .set 0          ; USE THESE MEMORY LOCATIONS
X2      .set 1
X3      .set 2
Y       .set 3
A       .set 127
B       .set 126
C       .set 125
*
START   ZAC             ; CLEAR ACCUMULATOR
        LDPK 0          ; USE PAGE 0
        LT   X3
        MPY  A          ; P = A*X3
        LTD  X2         ; T = X2, X2 --> X3, ACC = A*X3
        MPY  B          ; P = B*X2
        LTD  X1         ; T = X1, X1 --> X2, ACC = A*X3 + B*X2
        MPY  C          ; P = C*X1
        APAC            ; ACC = A*X3 + B*X2 + C*X1
        SACH Y,1        ; Y = ACCH
```

The table below illustrates the effect on data memory after execution of the code in Example 5-13.

Data Memory	Before Code Execution	After Code Execution
0h	X1	X1
1h	X2	X1
2h	X3	X2

The DMOV feature is useful in implementing filters and convolution algorithms.

5.4.2 Moving Constants into Data Memory

Most signal processors have a separate memory space for storing constants. By allowing communication between data and program memory, the TMS320C1x is able to incorporate a constant memory capability with its program memory, thus allowing an efficient use of memory space. The portion of memory not used for storing constants is available for use as program space.

Five immediate instructions provide an efficient way to execute operations using constants. The LARP instruction changes the auxiliary register pointer, and the LDPK instruction changes the data page pointer. The LACK, LARK, and MPYK instructions allow constants to be used in calculations. LACK and LARK both require an unsigned operand with a magnitude no greater than eight bits. The MPYK instruction allows a 13-bit signed number as an operand.

A 16-bit value can be moved from program memory to data memory using the TBLR instruction. TBLR requires that the program memory address (the source) be in the accumulator, while the data memory address (the destination) is obtained from the operand of the instruction. This instruction is commonly used to look up values in a table in program memory. The address of the value in the table is computed in the accumulator before executing the instruction. TBLR then moves the value into data memory. TBLR is a three-cycle instruction and, therefore, takes longer than an immediate instruction. However, it has more flexibility since it operates on 16-bit constants.

Sometimes it is convenient to store data operands in program ROM or external memory, and then read them into the on-chip RAM as they are needed. Two means are available for doing this. First, the TBLR (table read) instruction can be used to transfer data from on-chip program ROM to on-chip data RAM. Second, off-chip data RAM can be addressed via the IN and OUT instructions. With some extra hardware, the IN and OUT instructions can be used to read and write from data RAM to large amounts of external storage addressed as a peripheral.

Data may also be transferred from data memory to program memory by means of the TBLW instruction. The IN and OUT instructions can be used to transfer data between the on-chip data memory and the I/O space (see Section 6.1).

Note that the TBLW (table write) instruction should not be used on the TMS320C17/E17 since this instruction transfers data from on-chip data RAM to external memory. The TMS320C17/E17 does not directly interface to external memory since the port address bits (PA2-PA0) are the only address lines external to the device.

Example 5-14 illustrates bringing the cosine value of a variable into data memory using the TBLR instruction. Note that if the address of COSINE is greater than 255, the address can be loaded into the accumulator by loading the T register with a one, multiplying by the constant COSINE, and transferring it from the P register into the accumulator.

Example 5-14. Moving a Constant into Data Memory Using the TBLR Instruction

```
* THIS ROUTINE USES THE TBLR INSTRUCTION TO BRING THE COSINE
* VALUE OF A VARIABLE INTO DATA MEMORY. A TABLE CONTAINING
* THE COSINE VALUES IS FIRST CREATED IN PROGRAM MEMORY.
*
COSINE       DATA
       .
       .
       .
START  IN    X,PAO
       LACK  COSINE      ; LOAD TABLE ADDRESS
       ADD   X           ; CALCULATE PROGRAM MEMORY ADDRESS
       TBLR  COSX        ; MOVE VALUE INTO DATA MEMORY
```

The following table shows the effect on data memory after the TBLR instruction has been executed in Example 5-14.

Program Memory	Before TBLR Execution	After TBLR Execution
COSINE + X	02FFh	02FFh
Data Memory		
COSX	71F2h	02FFh

Another method for transferring data from program memory into data memory uses the TBLR instruction. By using the TBLR instruction, a calculated, rather than predetermined, location of data in program memory may be specified for transfer. A routine using this approach is shown in Example 5-15.

Example 5-15. Moving Program Memory to Data Memory with TBLR

```
* THIS ROUTINE USES THE TBLR INSTRUCTION TO MOVE DATA VALUES
* FROM PROGRAM MEMORY INTO DATA MEMORY. BY USING THIS ROUTINE,
* THE PROGRAM MEMORY LOCATION IN THE ACCUMULATOR FROM WHICH
* DATA IS TO BE MOVED TO A SPECIFIC DATA MEMORY LOCATION CAN
* BE SPECIFIED. ASSUME THAT THE ACCUMULATOR CONTAINS THE
* ADDRESS IN PROGRAM MEMORY FROM WHICH TO TRANSFER THE DATA.
*
TABLE  LARP  1           ; USE AR1
       LARK  AR1,63      ; START FROM ADDRESS 63
*
LOOP   TBLR  *           ; MOVE DATA INTO DATA RAM
       BANZ  LOOP        ; TRANSFER 64 VALUES
       RET               ; RETURN TO CALLING PROGRAM
```

In cases where systems require that temporary storage be allocated in the program memory, TBLW can be used to transfer data from internal data memory to external program memory. The code in Example 5-16 demonstrates how this may be accomplished.

Example 5-16. Moving Internal Data Memory to Program Memory with TBLW

```
* THIS ROUTINE USES THE TBLW INSTRUCTION TO MOVE DATA VALUES
* FROM INTERNAL DATA MEMORY TO EXTERNAL PROGRAM MEMORY. THE
* CALLING ROUTINE MUST SPECIFY THE DESTINATION PROGRAM MEMORY
* ADDRESS IN THE ACCUMULATOR. ASSUME THAT THE ACCUMULATOR
* CONTAINS THE ADDRESS IN PROGRAM MEMORY INTO WHICH THE DATA
* IS TRANSFERRED.
*
TABLE  LARK  AR1,63   ; LOAD LOOP COUNT OF 64
       LARK  AR0,DAT1 ; LOAD STARTING ADDRESS
*
LOOP   LARP  AR0      ; USE AR0
       TBLW  *+,AR1   ; MOVE DATA TO EXTERNAL PROGRAM RAM
       BANZ  LOOP     ; DECREMENT AND CHECK IF DONE
       RET            ; RETURN TO CALLING PROGRAM
```

After the execution of the TBLW instruction, the following effect has occurred on program memory:

Program Memory	Before TBLW Execution	After TBLW Execution
PROG1	0FF10h	1234h
Data Memory		
DAT1	1234h	1234h

The IN and OUT instructions are used to transfer data between the data memory and the I/O space, as shown in Example 5-17 and Example 5-18.

Example 5-17. Moving Data from I/O Space into Data Memory with IN

```
* THIS ROUTINE USES THE IN INSTRUCTION TO MOVE DATA VALUES
* FROM THE I/O SPACE INTO DATA MEMORY. DATA ACCESSED FROM
* I/O PORT 7 IS TRANSFERRED TO SUCCESSIVE MEMORY LOCATIONS
* ON DATA PAGE 0.
*
INPUT  LARK  AR0,32       ; SET UP LOOP COUNT
       LARK  AR1,DAT1     ; SET UP DESTINATION ADDRESS
*
LOOP   LARP  AR1          ; USE AR1
       IN    *+,PA7,AR0   ; MOVE DATA INTO DATA RAM
       BANZ  LOOP         ; DECREMENT AND CHECK IF DONE
       RET                ; RETURN TO CALLING PROGRAM
```

Example 5-18. Moving Data from Data Memory to I/O Space with OUT

```
* THIS ROUTINE USES THE OUT INSTRUCTION TO MOVE DATA VALUES
* FROM THE DATA MEMORY TO THE I/O SPACE. DATA IS TRANSFERRED
* TO I/O PORT 7 FROM SUCCESSIVE MEMORY LOCATIONS ON DATA
* PAGE 0.
*
OUTPUT LARK  AR0,32       ; SET UP LOOP COUNT
       LARK  AR1,DAT1     ; SET UP STARTING ADDRESS
*
LOOP   LARP  AR1          ; USE AR1
       OUT   *+,PA7,AR0   ; MOVE DATA INTO I/O SPACE
       BANZ  LOOP         ; DECREMENT AND CHECK IF DONE
       RET                ; RETURN TO CALLING PROGRAM
```

5.5 Logical and Arithmetic Operations

Although the TMS320C1x instruction set is oriented toward digital signal processing, the same fundamental operations of a general-purpose processor, such as bit manipulation, logical and arithmetic operations, logical and arithmetic shifts, and overflow management, are included. Explanations and examples of how to use instructions for scaling, convolution operations, fixed-point multiplication/division/addition, and floating-point arithmetic are also included in this section.

The contents of the accumulator may be stored in data memory using the SACH and SACL instructions or stored in the stack by using the PUSH instruction. The accumulator may be loaded from data memory using the ZALH, ZALS, and LAC instructions, which zero the accumulator before loading the data value. The ZAC instruction zeroes the accumulator. POP can be used to restore the accumulator contents from the stack. The accumulator is also affected by the execution of the ABS instruction, which replaces the contents of the accumulator with its absolute value.

5.5.1 Bit Manipulation

A specified bit of a word from data memory can either be set, cleared, or tested. Such bit manipulations are accomplished by using the hardware shifter and the logic instructions, AND, OR, and XOR. In Example 5-19, operations on single bits are performed on the data word VALUE. In this and the following example, data memory location ONE contains the value 1 and MINUS contains the value -1 (all bits set).

Example 5-19. Single-Bit Manipulation

```
*
* CLEAR BIT 5 OF DATA MEMORY LOCATION VALUE. MEMORY LOCATION
* ONE CONTAINS CONSTANT 1. MEMORY LOCATION MINUS CONTAINS -1
* OR 0FFFFh.
*
        LAC  ONE,5   ; ACC = 00000020h
        XOR  MINUS   ; INVERT ACCUMULATOR; ACC = 0000FFDFh
        AND  VALUE   ; BIT 5 OF VALUE IS ZEROED
        SACL VALUE
*
* SET BIT 12 OF VALUE.
*
        LAC  ONE,12  ; ACC = 00001000h
        OR   VALUE   ; BIT 12 OF VALUE
        SACL VALUE
*
* TEST BIT 3 OF VALUE.
*
        LAC  ONE,3   ; ACC = 00000008h
        AND  VALUE   ; TEST BIT 3 OF VALUE
        BZ   BIT3Z   ; BRANCH TO BIT3Z IF BIT IS CLEAR
```

More than one bit can be set, cleared, or tested at one time if the necessary mask exists in data memory. In Example 5-20, the six low-order bits in the word VALUE are cleared if MASK contains the value 63.

Example 5-20. Multiple-Bit Manipulation

```
*
* CLEAR LOWER SIX BITS OF VALUE. MEMORY LOCATION MASK
* CONTAINS THE MASK TO CLEAR THE BITS. MEMORY LOCATION
* MINUS CONTAINS -1 OR 0FFFFh.
*
        LAC   MASK    ; ACC = 0000003Fh
        XOR   MINUS   ; INVERT ACCUMULATOR; ACC = 0000FFC0h
        AND   VALUE   ; CLEAR LOWER SIX BITS
        SACL  VALUE
```

5.5.2 Overflow Management

The TMS320C1x has two features that can be used to handle overflow management. These include the branch on overflow conditions and accumulator saturation (overflow mode). These features provide several options for overflow protection within an algorithm.

A program can branch to an error handler routine on an overflow of the accumulator by using the BV (branch on overflow) instruction. This instruction can be performed after any ALU operation that may cause an accumulator overflow.

The overflow mode is a feature useful for DSP applications. This mode simulates the saturation effect characteristic of analog systems. When enabled, any overflow in the accumulator results in the accumulator contents being replaced with the largest positive value (7FFFFFFFh) if the overflowed number is positive, or the largest negative value (80000000h) if negative. The overflow mode is controlled by the OVM bit of the status register and can be changed by the SOVM (set overflow mode), ROVM (reset overflow mode), or LST (load status register) instructions. Overflows can be detected in software by testing the OV (overflow) bit in the status register. When a branch is used to test the overflow bit, OV is automatically reset. Note that the OV bit does not function as a carry bit. It is set only when the absolute value of a number is too large to be represented in the accumulator, and it is not reset except by specific instructions. The overflow mode feature affects all arithmetic operations in the ALU.

In Example 5-21, the accumulator saturates to 7FFFFFFFh or the largest positive value. The BV instruction also clears the OV bit.

Example 5-21. Overflow Management

```
* THE ACCUMULATOR WILL SATURATE TO THE HIGHEST POSITIVE VALUE
* WHEN OVERFLOW OCCURS. THE ACCUMULATOR CONTAINS 7FFFF423h.
* MEMORY LOCATION A CONTAINS 74EDh. MEMORY LOCATION B
* CONTAINS 67AFh.
*
        SOVM           ; SET OVERFLOW MODE
        LT    A        ; T = 74EDh
        MPY   B        ; P = 2F5B4903h
        APAC           ; ACC = 7FFFFFFFh
        BV    OVRFLW   ; CHECK OV BIT
*                      ; BRANCH TO OVERFLOW HANDLING ROUTINE
```

The effect on the accumulator before and after the code execution is shown as follows:

	Before Code Execution	After Code Execution
ACC	7FFFF423h	7FFFFFFFh

5.5.3 Scaling

Scaling the data coming into the accumulator or already in the accumulator is useful in signal processing algorithms. This is frequently necessary in adaptation or other algorithms that must compute and apply correction factors or normalize intermediate results. Scaling and normalizing are implemented on the TMS320C1x via shifts of data on the incoming path to the accumulator.

There are two types of shifts: logical and arithmetic. A logical shift is implemented by filling the empty bits to the left of the MSB with zeros, regardless of the value of the MSB. An arithmetic shift fills the empty bits to the left of the MSB with ones if the MSB is one, or with zeros if the MBS is zero. The second type of bit padding is referred to as sign extension.

Data can be left-shifted 0 to 16 bits when the accumulator is loaded, and left-shifted 0, 1, or 4 bits when storing from the accumulator using the SACH instruction. These shifts can be used for loading numbers into the high 16 bits of the accumulator and renormalizing the result of a multiply. The incoming left shift of 0 to 16 bits is supplied in the instruction itself. Left shifts of data fetched from data memory are available for loading the accumulator (LAC), adding to the accumulator (ADD), and subtracting from the accumulator (SUB). When data is left-shifted 16 bits, the ZALH, ADDH, and SUBH instructions are used. The left-shift of 0, 1, or 4, available with the SACH instruction, is used to shift out the extra sign bits when fractional multiplication is used (see Section 5.5.5).

The hardware shift, which is built into the ADD, SUB, and LAC instructions, performs an arithmetic left-shift on a 16-bit word. This feature can also be used to perform right-shifts. A right-shift of n is implemented by performing a left-shift of 16-n and saving the upper word of the accumulator. Example 5-22 performs an arithmetic right-shift of 7 on a 16-bit number in the accumulator.

Example 5-22. Arithmetic Right-Shift

```
SACL  TEMP     ; MOVE NUMBER TO MEMORY
LAC   TEMP,9   ; SHIFT LEFT (16-7)
SACH  TEMP     ; SAVE HIGH WORD IN MEMORY
LAC   TEMP     ; RETURN NUMBER BACK TO ACCUMULATOR
```

The effect on the accumulator before and after the code execution is shown as follows:

	Before Code Execution	After Code Execution
ACC	0FFFFA452h	0FFFFFF48h

A logical right-shift of 4 on a 32-bit number stored in the accumulator is shown in Example 5-23. The 32-bit results of the shift are then stored in data memory. In this example, the accumulator initially contains the hexadecimal number, 9D84C1B2h. The variables, SHIFTH and SHIFTL, will receive the high word (09D8h) and low word (4C1Bh) of the shifted results.

Example 5-23. Logical Right-Shift

```
*
* SHIFT THE LOWER WORD. MEMORY LOCATION MINUS CONTAINS -1
* OR 0FFFFh.
*
      SACH SHIFTH      ; SHIFTH = 9D84h     INITIAL VALUES
      SACL SHIFTL      ; SHIFTL = 0C1B2h
      LAC  SHIFTL,12   ; ACC = 0FC1B2000h
      SACH SHIFTL      ; SHIFTL = 0FC1Bh
      LAC  MINUS,12    ; ACC = 0FFFFF000h
      XOR  MINUS       ; ACC = 0FFFF0FFFh
      AND  SHIFTL      ; ACC = 00000C1Bh
*
* SHIFT THE UPPER WORD.
*
      ADD  SHIFTH,12   ; ACC = 0F9D84C1Bh
      SACL SHIFTL      ; SHIFTL = 4C1Bh     FINAL LOW VALUE
      SACH SHIFTH      ; SHIFTH = 0F9D8h
      LAC  MINUS,12    ; ACC = 0FFFFF000h
      XOR  MINUS       ; ACC = 0FFFF0FFFh
      AND  SHIFTH      ; ACC = 000009D8h
      SACL SHIFTH      ; SHIFTH = 09D8h     FINAL HIGH VALUE
```

The accumulator is affected before and after the code execution as follows:

	Before Code Execution	After Code Execution
ACC	9D84C1B2h	09D84C1Bh

An arithmetic right-shift of 4 can be implemented using the same routine as shown above, except with the last four lines omitted.

5.5.4 Convolution Operations

Many DSP applications must perform convolution operations or other operations similar in form. These operations require data to be shifted or delayed. The DMOV and LTD instructions can perform the needed data moves for convolution.

The data move function is used for on-chip data memory. It allows a word to be copied from the currently addressed data memory location in on-chip RAM to the next higher location while the data from the addressed location is being operated upon (e.g., by the CALU). The data move and the CALU operation are performed in the same cycle. The data move function is useful in implementing algorithms, such as convolutions and digital filtering, where data is being passed through a time window. It models the z^{-1} delay operation encountered in those applications.

5.5.5 Multiplication

The TMS320C1x hardware multiplier normally performs two's-complement 16-bit by 16-bit multiplies and produces a 32-bit result in a single processor cycle. To multiply two operands, one operand must be loaded into the T register. The second operand is moved by the multiply instruction to the multiplier, which then produces the product in the P register. Before another multiply can be performed, the contents of the P register must be moved to the accumulator. By pipelining multiplies and P-register moves, most multiply operations can be performed with a single instruction.

Computation on the TMS320C1x is based on a fixed-point two's-complement representation of numbers. Each 16-bit number is evaluated with a sign bit, i integer bits, and 15-i fractional bits. Thus, the number

```
0 0000010 10100000
         |
         ├─ binary point
```

has a value of 2.625. This particular number is said to be represented in a Q8 format (8 fractional bits). Its range is between -128 (1000000000000000) and 127.996 (0111111111111111). The fractional accuracy of a Q8 number is about 0.004 (one part in 2^8 or 256).

Although particular situations (e.g., a combination of dynamic range and accuracy requirements) must use mixed notations, it is more common to work entirely with fractions represented in a Q15 format or integers in a Q0 format. This is especially true for signal processing algorithms where multiply and accumulate operations are dominant. The result of a fraction times a fraction remains a fraction, and the result of an integer times an integer remains an integer. No overflows are possible.

Q format is a number representation commonly used when performing operations on noninteger numbers. In Q format, the Q number (15 in Q15) denotes how many bits are located to the right of the binary point. A 16-bit number in Q15 format, therefore, has an assumed binary point immediately to the right of the most significant bit. Since the most significant bit constitutes the sign of the number, then numbers represented in Q15 may take on values from +1 (represented by +0.99997...) to -1.

A wide variety of situations may be encountered when multiplying two numbers. Three of these situations are provided in Example 5-24, Example 5-25, and Example 5-26.

Example 5-24. Fraction × Fraction (Q15 × Q15 = Q30)

```
                    0100000000000000   = 0.5  in Q15
                 ×  0100000000000000   = 0.5  in Q15
 ----------------------------------
00 01000000000000   0000000000000000   = 0.25 in Q30
  |
  |- binary point
```

Two sign bits remain after the multiply. Generally, a single-precision (16-bit) result is saved, rather than maintaining the full intermediate precision. The upper half of the result does not contain a full 15 bits of fractional precision since the multiply operation actually creates a second sign bit. In order to recover that precision, the product must he shifted left by one bit, as shown in the following code excerpt:

```
LT      OP1     ; OP1 = 4000h (0.5 in Q15)
MPY     OP2     ; OP2 = 4000h (0.5 in Q15)
PAC
SACH    ANS,1   ; ANS = 2000h (0.25 in Q15)
```

The MPYK instruction provides a multiply by a 13-bit signed constant. In fractional notation, this means that a Q15 number can be multiplied by a Q12 number. The resulting number must be left-shifted by four bits to maintain full precision.

```
LT      OP1     ; OP1 = 4000h (0.5  in Q15)
MPYK    2048    ; OP2 = 0800h (0.5  in Q12)
PAC
SACH    ANS,4   ; ANS = 2000h (0.25 in Q15)
```

Example 5-25. Integer × Integer (Q0 × Q0 = Q0)

```
                   0000000000010001  =  17 in Q0
                ×  1111111111111011  =  -5 in Q0
 ---------------------------------
1111111111111111   1111111110101011  = -85 in Q0
                                    |
                                    |- binary point
```

In this case, the extra sign bits do not change the result, and the desired product is entirely in the lower half of the product, as shown in the following program:

```
LT      OP1     ; OP1 = 0011h ( 17 in Q0)
MPY     OP2     ; OP2 = 0FFFBh ( -5 in Q0)
PAC
SACH    ANS     ; ANS = 0FFABh (-85 in Q0)
```

Example 5-26. Mixed Notation (Q14 × Q14 = Q28)

```
                      0110000000000000  = 1.50  in Q14
                    × 0011000000000000  = 0.75  in Q14
------------------------------------------
0001  001000000000    0000000000000000  = 1.125 in Q28
   └─ binary point
```

The maximum magnitude of a Q14 number is just under two. Thus, the maximum magnitude of the product of two Q14 numbers is four. Two integer bits are required to allow for this possibility, leaving a maximum precision for the product of 13 bits. In general, the following rule applies: The product of a number with i integer bits and f fractional bits and a second number with j integer bits and g fractional bits will be a number with (i+j) integer bits and (f+g) fractional bits. The highest precision possible for a 16-bit representation of this number will have (i+j) integer bits and (15-i-j) fractional bits.

If the physical system being modelled is well understood, the precision with which the number is modelled can be increased. For example, if it is known that the above product can be no more than 1.8, the product can be represented as a Q14 number rather than the theoretical worst case of Q13, shown in the following program:

```
LT      OP1       ; OP1 = 6000h (1.5   in Q14)
MPY     OP2       ; OP2 = 3000h (0.75  in Q14)
PAC
SACH    ANS,1     ; ANS = 2400h (1.125 in Q13)
```

The techniques illustrated in the previous three examples all truncate the result of the multiplication to the desired precision. The error generated as a result can be as much as minus one full LSB. This is true whether the truncated number is positive or negative. It is possible to implement a simple rounding technique to reduce this potential error by a factor of two, as shown in the code sequence of Example 5-27. The maximum error generated in this example is plus one-half LSB whether ANS is positive or negative.

Example 5-27. Rounding Technique for Multiplication

```
LT      OP1
MPY     OP2       ; OP1 * OP2
PAC
ADD     ONE,14    ; ROUND UP
SACH    ANS,1
```

A common operation in DSP algorithms is the summation of products. The contents of the P register are added to the accumulator, and two values simultaneously read and multiplied. A data memory value is multiplied by a program memory value. Example 5-28 shows an implementation of multiplies and accumulates using the LTA-MPY instruction pair.

Example 5-28. Multiply and Accumulate Using the LTA-MPY Instruction Pair

```
*                 CLOCK    TOTAL CLOCK   PROGRAM     TOTAL PROGRAM
*                 CYCLES    CYCLES       MEMORY         MEMORY
*
    ZAC            1                      1
    LT    D1       1                      1
    MPY   C1       1                      1
    LTA   D2       1                      1
    MPY   C2       1                      1
     .
     .              2N                     2N
     .
    LTA   DN       1                      1
    MPY   CN       1                      1
    APAC           1        2 + 2N        1              2 + 2N
```

5.5.6 Division

Binary division is the inverse of multiplication. Multiplication consists of a series of shift and add operations, while division can be broken into a series of subtracts and shifts. Although the first-generation TMS320 does not have an explicit divide instruction, it is possible to implement an efficient flexible divide capability using the conditional subtract instruction, SUBC. SUBC implements binary division in the same manner as is commonly done in long division. Given a 16-bit positive dividend and divisor, the repetition of the SUBC command 16 times produces a 16-bit quotient in the low accumulator and a 16-bit remainder in the high accumulator. With each SUBC, the divisor is left-shifted 15 bits and subtracted from the accumulator. For each subtract not producing a negative answer, a one is put in the LSB of the quotient and then shifted. For each subtract producing a negative answer, the accumulator is simply left-shifted. The shifting of the remainder and quotient after each subtract produces the separation of the quotient and remainder in the low and high halves of the accumulator. The similarities between long division and the SUBC method of division are shown in Figure 5-1 where 33 is divided by 5.

```
LONG DIVISION:

                    0000000000000110      Quotient
0000000000000101 )0000000000100001
                        -101
                         110
                        -101
                          11              Remainder


SUBC METHOD:

32  HIGH ACC       LOW ACC       0              COMMENT
|               | |               |
0000000000000000  0000000000100001  (1)  Dividend is loaded into ACC. The
             -10  1000000000000000       divisor is left-shifted 15 and sub-
------------------------------------     tracted from ACC. The subtraction
             -10  0111111111011111       is negative, so discard the result
|               | |               |      and shift left the ACC one bit.
0000000000000000  0000000001000010  (2)  2nd subtract produces negative
             -10  1000000000000000       answer, so discard result and shift
------------------------------------     ACC (dividend) left.
             -10  0111111110111110

                 •                                 •
                 •                                 •
                 •                                 •
|               | |               |
0000000000000100  0010000000000000  (14) 14th SUBC command. The result
             -10  1000000000000000       is positive. Shift result left and
------------------------------------     replace LSB with '1'.
0000000000000001  1010000000000000

|               | |               |
0000000000000011  0100000000000001  (15) Result is again positive. Shift
             -10  1000000000000000       result left and replace LSB with '1'.
------------------------------------
0000000000000000  1100000000000001

|               | |               |
0000000000000001  1000000000000011  (16) Last subtract. Negative answer, so
             -10  1000000000000000       discard result and shift ACC left.
------------------------------------
               -  1111111111111101

------------------------------------
0000000000000011  0000000000000110       Answer reached after 16 SUBC
                                         instructions.
|   REMAINDER   | |   QUOTIENT    |
```

Figure 5-1. Long Division and SUBC Division

The condition of the divisor, less than the shifted dividend, is determined by the sign of the result. The only restriction for the use of the SUBC instruction is that both the dividend and divisor MUST be positive. Thus, the sign of the quotient must be determined and the quotient computed using the absolute value of the dividend and divisor. In addition, when implementing a divide algorithm, it is important to know if the quotient can be represented as a fraction and the degree of accuracy to which the quotient is to be computed. Each of these considerations can affect how the SUBC instruction is used (see Example 5-29 and Example 5-30). Note that the next instruction after SUBC cannot use the accumulator.

Example 5-29. Using SUBC Where Numerator < Denominator

```
* THIS ROUTINE DIVIDES TWO BINARY, TWO'S-COMPLEMENT NUMBERS
* OF ANY SIGN WHERE THE NUMERATOR IS LESS THAN THE
* DENOMINATOR.
*
*                   BEFORE        AFTER
*                INSTRUCTION  INSTRUCTION
*
* NUMERA              21            21
* DENOM               42            42
* QUOT                 0           0.5
*                               (0.1 0 0)
*
DIV     LARP  0
        LT    NUMERA   ; GET SIGN OF QUOTIENT
        MPY   DENOM
        PAC
        SACH  TEMSGN   ; SAVE SIGN OF QUOTIENT
        LAC   DENOM
        ABS
        SACL  DENOM    ; MAKE DENOMINATOR POSITIVE
        ZALH  NUMERA   ; ALIGN NUMERATOR
        ABS            ; MAKE NUMERATOR POSITIVE
        LARK  0,14
*
* IF DIVISOR AND DIVIDEND ARE ALIGNED, DIVISION CAN START
* HERE.
*
KPDVNG  SUBC  DENOM    ; 15-CYCLE DIVIDE LOOP
        BANZ  KPDVNG
*
        SACL  QUOT
        LAC   TEMSGN
        BGEZ  DONE     ; DONE IF SIGN IS POSITIVE
        ZAC
        SUB   QUOT
        SACL  QUOT     ; NEGATE QUOTIENT IF NEGATIVE
DONE    RET            ; RETURN TO MAIN PROGRAM
```

Example 5-30. Using SUBC Where Accuracy of Quotient Specified

```
* THIS ROUTINE DIVIDES TWO BINARY, TWO'S-COMPLEMENT NUMBERS
* OF ANY SIGN, SPECIFYING THE FRACTIONAL ACCURACY OF THE
* QUOTIENT (FRAC).
*
*                 BEFORE       AFTER
*               INSTRUCTION  INSTRUCTION
*
* NUMERA          11            11
* DENOM            8             8
* FRAC             3             3
* QUOT            17           1.375
*                             (1.0 1 1)
*
DN1     LT    NUMERA   ; GET SIGN OF QUOTIENT
        MPY   DENOM
        PAC
        SACH  TEMSGN   ; SAVE SIGN OF QUOTIENT
        LAC   DENOM
        ABS
        SACL  DENOM    ; MAKE DENOMINATOR POSITIVE
        LACK  15
        ADD   FRAC
        SACL  FRAC     ; COMPUTE LOOP COUNT
        LAC   NUMERA   ; ALIGN NUMERATOR
        ABS            ; MAKE NUMERATOR POSITIVE
        LAR   0,FRAC
*
* IF DIVISOR AND DIVIDEND ARE ALIGNED, DIVISION CAN START
* HERE.
*
KPDVNG  SUBC  DENOM    ; 16 + FRAC CYCLE DIVIDE LOOP
        BANZ  KPDVNG
*
        SACL  QUOT
        LAC   TEMSGN
        BGEZ  DONE     ; DONE IF SIGN IS POSITIVE
        ZAC
        SUB   QUOT
        SACL  QUOT     ; NEGATE QUOTIENT IF NEGATIVE
DONE    RET            ; RETURN TO MAIN PROGRAM
```

5.5.7 Addition

Both operands in division must be represented in the same Q format. Enough room must be allowed in the result to accommodate bit growth or there must be some preparation to handle overflows. If the operands are only 16 bits long, the result may have to be represented as a double-precision number. Example 5-31 and Example 5-32 illustrate two approaches to adding 16-bit numbers.

Example 5-31. Maintaining 32-Bit Results

```
        LAC   OP1      ; Q15
        ADD   OP2      ; Q15
        SACH  ANSHI    ; HIGH-ORDER 16 BITS OF RESULT
        SACL  ANSLO    ; LOW-ORDER 16 BITS OF RESULT
```

Example 5-32. Adjusted Binary Point to Maintain 16-Bit Results

```
LAC   OP1,15   ; Q14 NUMBER IN ACCH
ADD   OP2,15   ; Q14 NUMBER IN ACCH
SACH  ANS      ; Q14
```

Double-precision operands present a more complex problem since actual arithmetic overflows or underflows may occur. The BV (branch on overflow) instruction can be used to check for the occurrence of these conditions. A second technique is the use of saturation mode operations, which will saturate the result of overflowing accumulations to the most positive or most negative number. Both techniques, however, result in a loss of precision. The best technique involves a thorough understanding of the underlying physical process and care in selecting number representations.

5.5.8 Floating-Point Arithmetic

Although the TMS320C1x devices are fixed-point 16/32-bit microprocessors, they can also perform floating-point computations. Using the floating-point single-precision standard proposed by the IEEE, the TMS320C1x can perform a floating-point multiplication in 8.4 µs and a floating-point addition in 17.2 µs. For a detailed discussion of floating-point arithmetic and TMS320 source code, refer to "Floating-Point Arithmetic with the TMS32010," an application report in the book, *Digital Signal Processing Applications with the TMS320 Family* (literature number SPRA012A).

Floating-point numbers are often represented on microprocessors in a two-word format of mantissa and exponent. The mantissa is stored in one word. The exponent, the second word, indicates how many bit positions from the left the binary point is located. If the mantissa is 16 bits, a 4-bit exponent is sufficient to express the location of the binary point. Because of its 16-bit word size, the 16/4-bit floating-point format functions most efficiently on the TMS320C1x.

Operations in the TMS320C1x central ALU are performed in two's-complement fixed-point notation. To implement floating-point arithmetic, operands must be converted to fixed point for arithmetic operations, and then converted back to floating point. Conversion to floating-point notation is performed by normalizing the input data (i.e., shifting the MSB of the data word into the MSB of the internal memory word). The exponent word then indicates how many shifts are required. To multiply two floating-point numbers, the mantissas are multiplied and the exponents added. The resulting mantissa must be renormalized. (Since the input operands are normalized, no more then one left shift is required to normalize the result.)

Floating-point addition or subtraction requires shifting the mantissa so that the exponents of the two operands match. The difference between the exponents is used to left-shift the lower power operand before adding. Then, the output of the add must be renormalized.

Instructions useful in floating-point operations are the LAC, LACK, ADD, and SUB instructions. The mantissas are often used in Q15 format. Q format is a number representation commonly used when performing operations on non-integer numbers. In Q format, the Q number (15 in Q15) denotes how many digits are located to the right of the binary point. A 16-bit number in Q15 format, therefore, has an assumed binary point immediately to the right of the most significant bit. Since the most significant bit constitutes the sign of the number, then numbers represented in Q15 may take on values from +1 (represented by +0.99997...) to -1.

5.6 Application-Oriented Operations

The TMS320C1x has been designed to provide efficient implementations of many common digital signal processing algorithms. Its features provide solutions to numerically intensive problems usually characterized by multiply and accumulate operations. Some device-specific features that aid in the implementation of specific algorithms on the TMS320C1x include companding, filtering, Fast Fourier Transforms (FFT), and PID control. These applications require I/O performed either in parallel or serial.

5.6.1 Companding

In the area of telecommunications, one of the primary concerns is the I/O bandwidth in the communications channel. One way to minimize this bandwidth is by companding (COMpress/exPAND). Companding is defined by two international standards, A-law and μ-law, both based on the compression of the equivalent of 13 bits of dynamic range into an 8-bit code. The standard employed in the United States and Japan is μ-law companding. The European standard is referred to as A-law companding. Detailed descriptions and code examples of μ-law and A-law companding are presented in "Companding Routines for the TMS32010/TMS32020," an application report included in the book, *Digital Signal Processing Applications with the TMS320 Family* (literature number SPRA012A).

The technique of companding allows the digital sample information corresponding to a 13-bit dynamic range to be transmitted as 8-bit data. For processing in the TMS320C1x, it is necessary to convert the 8-bit logarithmic data to a 16-bit linear format. Prior to output, the linear result must be converted to the compressed or companded format. On the TMS32010/C10/C15, companding must be performed in software using conversion routines. On-chip companding hardware on the TMS320C17/E17 implements these functions.

Software routines for μ-law and A-law companding, flowcharts, companding algorithms, and detailed descriptions are provided in the application report on companding routines in the book, *Digital Signal Processing Applications with the TMS320 Family* (literature number SPRA012A). The algorithm space and time requirements for μ-law and A-law companding on the TMS32010/C10/C15/E15 are given in Table 5-2.

Table 5-2. Program Space and Time Requirements for μ-/A-Law Companding

FUNCTION	WORDS OF MEMORY		PROGRAM CYCLES		TIME REQD†
	Program	Data	Initialization	Loop‡	μs
μ-Law:					
Compression	105	13	17	40	8.0
Expansion	46	8	6	23	4.6
A-Law:					
Compression	97	11	14	36	7.2
Expansion	48	7	4	25	5.0

†Assuming initialization
‡Worst case

Four modes are available for the on-chip companding hardware operation on the TMS320C17/E17: serial encode, serial decode, parallel encode, and parallel decode. The companding hardware converts between two's-complement or sign-magnitude format and the companded format.

In the serial encode mode, transmitted data is encoded according to either μ-law or A-law format. In the serial decode mode, received data is decoded to a linear format according to the specified companding law.

In the parallel modes, either the encoder or decoder is enabled, and then data written to port 1 is compressed or expanded. To convert sign-magnitude or two's-complement linear PCM to 8-bit log PCM, the encoder is enabled for parallel operation, and the sample is written to port 1. An IN instruction from port 1 returns the linear PCM value. To convert an 8-bit log PCM to a sign-magnitude or two's-complement linear PCM, the decoder is enabled for parallel operation, and the 8-bit sample is written to port 1. The expanded linear value is returned on the IN instruction from port 1. Note that when the conversion mode selected converts a two's-complement value, there must be one instruction cycle between the OUT and IN instructions. Care should be taken to have one OUT-IN instruction sequence to port 1 for each data sample, because the execution of two OUT instructions to port 1 in succession pushes the first sample into the transmit register TR1, preventing access for read purposes. OUT instructions to port addresses 2 through 7 will not affect serial-port operations.

When the companding hardware converts to sign-magnitude data, it must be converted to two's-complement notation for computation in the microcomputer. Sign-magnitude notation consists of a sign bit in the MSB: a zero indicating a positive value, and a one indicating a negative number. All bits between the sign bit and the MSB of the data value are set to zero. For conversions between μ-law and sign-magnitude linear PCM, the hexadecimal value 1FFFh represents the most positive value of 8191 and the value 9FFFh represents the most negative value of -8191. For conversions between A-law and sign-magnitude linear PCM, the hexadecimal value 0FFFh represents the most positive value of 4095 and the value 8FFFh represents the most negative alue of -4095.

Conversion between sign-magnitude and two's-complement data for μ-law encoding and decoding is implemented with the code shown in Example 5-33 and Example 5-34, respectively. Conversion between two's-complement and sign-magnitude data for A-law encoding and decoding is implemented with the code shown in Example 5-35 and Example 5-36, respectively. Note that both TMS320C17/E17 devices feature hardware companding logic that can operate in either μ-law or A-law format with either sign-magnitude or two's-complement numbers.

Example 5-33. Two's-Complement to Sign-Magnitude for μ-Law Encoding

```
* THIS ROUTINE CONVERTS A TWO'S-COMPLEMENT NUMBER TO 14-BIT
* SIGN-MAGNITUDE FORMAT AND ADDS THE BIAS OF 33 FOR MU-LAW
* ENCODING. MEMORY LOCATION 1 CONTAINS THE VALUE 1 AND
* MEMORY LOCATION 2 (BIAS) CONTAINS +33.
*
OUTPUT
       LAC  SAMPLE      ; GET THE LINEAR DATA FOR OUTPUT
       BGEZ POSOUT      ; IF POSITIVE, CHECK POS MAX VALUE
       ABS              ; IF NEGATIVE, CHECK ABSOLUTE VALUE
       ADD  BIAS        ; ADD IN THE BIAS OF 21h
       ADD  ONE,15      ; SET THE SIGN BIT NEGATIVE
       SACL SAMPLE      ; HOLD FOR LATER
       SUB  NEGMAX      ; COMPARE TO NEGATIVE MAX = 9FFFh
       BLEZ DONE        ; IF WITHIN MAX, THEN SEND IT
       LAC  NEGMAX      ; ELSE, LOAD THE VALUE WITH THE
       SACL SAMPLE      ; LARGEST NEGATIVE IN RANGE
       B    DONE        ; AND SEND IT
POSOUT ADD  BIAS        ; ADD IN THE BIAS OF 21h
       SACL SAMPLE      ; AND SAVE IT
       SUB  POSMAX      ; COMPARE TO POSITIVE MAX = 1FFFh
       BLEZ DONE        ; IF WITHIN MAX, THEN SEND IT
       LAC  POSMAX      ; ELSE, LOAD THE VALUE WITH THE
       SACL SAMPLE      ; LARGEST POSITIVE VALUE IN RANGE
DONE   OUT  SAMPLE,PA1  ; AND SEND IT TO ENCODER
* ...  CONTINUE CODE HERE
```

Example 5-34. Sign-Magnitude to Two's-Complement for μ-Law Decoding

```
* THIS ROUTINE CONVERTS A 14-BIT SIGN-MAGNITUDE NUMBER TO
* TWO'S-COMPLEMENT NOTATION AND REMOVES THE BIAS OF 33 FOR
* MU-LAW DECODING.  MEMORY LOCATION 1 CONTAINS THE VALUE 1
* AND MEMORY LOCATION 2 (BIAS) CONTAINS 33.
*
INPUT
       IN   SAMPLE,PA1  ; READ INPUT FROM SERIAL PORT; DECODE
       LAC  SAMPLE      ; MOVE INPUT TO ACCUMULATOR
       SUB  BIAS        ; REMOVE BIAS VALUE
       BGEZ POS         ; IF POSITIVE, THEN SAVE IT
       ADD  ONE,15      ; ELSE, DELETE SIGN BIT BY CARRY
       SACL SAMPLE      ; SAVE MAGNITUDE VALUE
       ZAC              ; NEGATE THE INPUT BY
       SUB  SAMPLE      ; SUBTRACTING FROM ZERO AND SAVE
POS    SACL SAMPLE      ; FULLY EXPANDED LINEAR DATA
* ...  CONTINUE CODE HERE
```

Example 5-35. Two's-Complement to Sign-Magnitude for A-Law Encoding

```
* THIS ROUTINE CONVERTS A TWO'S-COMPLEMENT NUMBER TO 13-BIT
* SIGN-MAGNITUDE NOTATION FOR A-LAW ENCODING.  MEMORY
* LOCATION 1 CONTAINS THE VALUE 1.
*
OUTPUT
        LAC  SAMPLE      ; GET THE LINEAR DATA FOR OUTPUT
        BGEZ POSOUT      ; IF POSITIVE, CHECK POS MAX VALUE
        ABS              ; IF NEGATIVE, CHECK NEG MAX VALUE
        ADD  ONE,15      ; SET THE SIGN BIT NEGATIVE
        SACL SAMPLE      ; HOLD FOR LATER
        SUB  NEGMAX      ; COMPARE TO NEGATIVE MAX = 8FFFh
        BLEZ DONE        ; IF WITHIN MAX, THEN SEND IT
        LAC  NEGMAX      ; ELSE, LOAD THE VALUE WITH THE
        SACL SAMPLE      ; LARGEST NEGATIVE IN RANGE
        B    DONE        ; AND SEND IT
POSOUT  SACL SAMPLE      ; SAVE IT
        SUB  POSMAX      ; COMPARE TO POSITIVE MAX = 0FFFh
        BLEZ DONE        ; IF WITHIN MAX, THEN SEND IT
        LAC  POSMAX      ; ELSE, LOAD THE VALUE WITH THE
        SACL SAMPLE      ; LARGEST POSITIVE VALUE IN RANGE
DONE    OUT  SAMPLE,PA1 ; AND SEND IT TO ENCODER
* ...   CONTINUE CODE HERE
```

Example 5-36. Sign-Magnitude to Two's-Complement for A-Law Decoding

```
* THIS ROUTINE CONVERTS A 13-BIT SIGN-MAGNITUDE NUMBER TO
* TWO'S-COMPLEMENT NOTATION FOR A-LAW ENCODING.  MEMORY
* LOCATION 1 CONTAINS THE VALUE 1.
*
INPUT
        IN   SAMPLE,PA1 ; READ INPUT FROM SERIAL PORT; DECODE
        LAC  SAMPLE      ; MOVE INPUT TO ACCUMULATOR
        BGEZ POS         ; IF POSITIVE, THEN SAVE IT
        ADD  ONE,15      ; ELSE, DELETE SIGN BIT BY CARRY
        SACL SAMPLE      ; SAVE MAGNITUDE VALUE
        ZAC              ; NEGATE THE INPUT BY
        SUB  SAMPLE      ; SUBTRACTING FROM ZERO AND SAVE
POS     SACL SAMPLE      ; FULLY EXPANDED LINEAR DATA
* ...   CONTINUE CODE HERE
```

5.6.2 FIR/IIR Filtering

Digital filters are a common requirement for digital signal processing systems. The filters fall into two basic categories: Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters. For either category of filter, the coefficients of the filter (weighting factors) may be fixed or adapted during the course of the signal processing. The theory and implementation of digital filters has been presented and discussed in an application report, "Implementation of FIR/IIR Filters with the TMS32010/TMS32020," included in the book, *Digital Signal Processing Applications with the TMS320 Family* (literature number SPRA012A).

IIR filters benefit from the fast instruction cycle time of the TMS320C1x. IIR filters typically require fewer multiply/accumulates. Correspondingly, the amount of data memory for samples and coefficients is not usually the limiting factor. Because of sensitivity to quantization of the coefficients themselves, IIR filters are usually implemented in cascaded second-order sections. This translates to code consisting of LTD-MPY instruction pairs. Example 5-37 provides an implementation of a second-order IIR filter.

Example 5-37. Implementing an IIR Filter

```
*
* THE FOLLOWING EQUATIONS ARE USED TO IMPLEMENT AN IIR FILTER:
*
*    d(n) = x(n)    + d(n-1)a1 + d(n-2)a2
*    y(n) = d(n)b0 + d(n-1)b1 + d(n-2)b2
*
START  IN    XN,PA0  ; INPUT NEW VALUE XN
       LAC   XN,15   ; LOAD ACCUMULATOR WITH XN
*
       LT    DNM1
       MPY   A1
*
       LTD   DNM2
       MPY   A2
*
       APAC
       SACH  DN,1    ; d(n) = x(n) + d(n-1)a1 + d(n-2)a2
       ZAC
       MPY   B2
*
       LTD   DNM1
       MPY   B1
*
       LTD   DN
       MPY   B0
*
       APAC
       SACH  YN,1    ; y(n) = d(n)b0 + d(n-1)b1 + d(n-2)b2
       OUT   YN,PA1  ; YN IS THE OUTPUT OF THE FILTER
```

FIR filters also benefit from the fast instruction cycle time. In addition, an FIR filter requires many more multiply/accumulates than does the IIR filter with equivalent sharpness at the cutoff frequencies and with distortion and attenuation in the passbands and stopbands. The TMS320C1x devices help solve this problem by making longer filters feasible to implement. The TMS320C15/E15/C17/E17 has expanded data memory of 256 words, thus allowing additional coefficients and samples to be stored for longer-length filters. Example 5-34 provides an implementation of a fourth-order (4 taps) FIR filter. Each tap consists of a LTD-MPY instruction pair, uses two data memory locations, and takes two instruction cycles to execute.

Example 5-34. Implementing an FIR Filter

```
*      *
* THE FOLLOWING EQUATION IS USED TO IMPLEMENT AN FIR FILTER:
*
*    y(n)=[Ax(n-1)+Cx(n-3)+Dx(n-4)]* 2**-16
*
START  IN    X1,PA0  ; INPUT SAMPLE
       ZAC
*
       LT    X4      ; x(n-4)
       MPY   D
*
       LTD   X3      ; ACC=Dx4; x(n-4))=x(n-3)
       MPY   C
*
       LTD   X2      ; ACC=Dx4+Cx3; x(n-3))=x(n-2)
       MPY   B
*
       LTD   X1      ; ACC=Dx4+Cx3+Bx2; x(n-2))=x(n-1)
       MPY   A
*
       APAC          ; ACC=Dx4+Cx3+Bx2+Ax1
       SACH  Y,1
       OUT   Y,PA1   ; OUTPUT RESULTS
       B     START
```

The implementation of the FIR filter using straightline code was shown in Example 5-34. For longer-length FIR filters, straightline code may require larger program memory size. Depending on system constraints, the designer may choose to reduce program memory size by using looped code. However, straightline code will run much faster than looped versions. The design trade-off should be carefully considered by the design engineer.

5.6.3 Adaptive Filtering

With FIR or IIR filtering, the filter coefficients may be fixed or adapted. If the coefficients are adapted or updated with time, then another factor impacts the computational capacity. This factor is the requirement to adapt each of the coefficients, usually with each sample. A means of adapting the coefficients is the Least-Mean-Square (LMS) algorithm given by the following equation:

$$b_k(i+1) = b_k(i) + 2B\ e(i)\ x(i-k)$$

$$\text{where } e(i) = x(i) - y(i)$$

$$\text{and } y(i) = \sum_{k=0}^{N-1} b_k\ x(i-k)$$

Quantization errors in the updated coefficients can be minimized if the result is obtained by rounding rather than truncating. For each coefficient in the filter at a given point in time, the factor 2B e(i) is a constant. This factor can then be computed once and stored in the T register for each of the updates. Thus, the computational requirement has become one multiply/accumulate plus rounding. The adaptation of each coefficient is five instructions corresponding to five clock cycles. This is shown in the instruction sequence as follows:

```
LARK   AR0,LASTAP   ; POINT TO DATA SAMPLE
LARK   AR1,COEFFD   ; POINT TO COEFFICIENTS
LARP   AR0
LT     ERRF         ; errf = 2B*e(i)
  .
  .
  .
MPY    *-,AR1       ; P = 2B*e(i)*X(i-0)
ZALH
APAC                ; b0(i+1) = b0(1) + P
ADD    ONE,15       ; ROUND
SACH   *+,0,AR0     ; STORE b0(i+1)
  .
  .
  .
```

Example 5-39 shows a routine to filter a signal and update the coefficients. The total execution time of the routine is 30 + 7n where n is the filter length. Data and program memory requirements are 5 + 2n words and 28 + 7n words, respectively. The filter length for adaptive filters is restricted both by execution time and memory. There is obviously more processing to be completed per sample due to the adaptation, and the size of the on-chip data RAM limits the number of coefficients and data samples that can be stored.

Another routine on adaptive filtering is discussed in the book, *Digital Signal Processing Applications with the TMS320 Family* (literature number SPRA012A); see application report, "Digital Voice Echo Canceller with a TMS32020."

Example 5-39. 32-Tap Adaptive Filter

```
        .title  'ADAPTIVE FILTER'
        .def    ADPFIR
        .def    X,Y
*
* THIS 32-TAP ADAPTIVE FILTER USES PAGE 0 FOR COEFFICIENTS
* AND DATA SAMPLES. THE NEWEST INPUT SHOULD BE IN MEMORY
* LOCATION X WHEN CALLED. THE OUTPUT WILL BE IN MEMORY
* LOCATION Y WHEN RETURNED.
*
ONE     .set 120            ; CONSTANT ONE
BETA    .set 121            ; ADAPTATION CONSTANT * 2
ERR     .set 122            ; SIGNAL ERROR
ERRF    .set 123            ; ERROR FUNCTION
Y       .set 124            ; FILTER OUTPUT
X       .set 125            ; NEWEST DATA SAMPLE
FRSTAP  .set  32            ; NEXT NEWEST DATA SAMPLE
LASTAP  .set  63            ; OLDEST DATA SAMPLE
COEFFD  .set   0            ; START OF COEFFICIENT TABLE
*
* FINITE IMPULSE  RESPONSE (FIR) FILTER.
*
        .text
ADPFIR  LDPK 0              ; USE DATA PAGE 0
        LARK AR1,COEFFD     ; LOAD POINTER FOR COEFF TABLE
        LARK AR0,LASTAP     ; LOAD POINTER FOR DATA SAMPLES
        MPYK 0              ; CLEAR THE P REGISTER
        LAC  ONE,14         ; LOAD OUTPUT ROUNDING BIT
        LARP AR0
*
* DO 32 TAPS.
*
FIR     LT   *-,AR1         ; LOAD T REG WITH OLDEST SAMPLE
        MPY  *+,AR0         ; MULTIPLY WITH LAST COEFFICIENT
*
        LTD  *-,AR1         ; LOAD NEXT SAMPLE
        MPY  *+,AR0         ; MULTIPLY WITH NEXT COEFFICIENT
*
        LTD  *-,AR1         ; LOAD NEXT SAMPLE
        MPY  *+,AR0         ; MULTIPLY WITH NEXT COEFFICIENT
         .
         .
         .
        LTD  *-,AR1         ; LOAD LAST SAMPLE
        MPY  *+,AR0         ; MULTIPLY WITH LAST COEFFICIENT
*
        APAC
        SACH Y,1            ; STORE FILTER OUTPUT
        ZAC
        SUB  Y              ; ACC = -y(i)
        ADD  X              ; ADD THE NEWEST INPUT
        SACL ERR            ; err(i) = x(i) - y(i)
*
* LMS ADAPTATION OF FILTER COEFFICIENTS.
*
        LT   ERR
        MPY  BETA
        PAC                 ; errf(i) = 2*beta*err(i)
        ADD  ONE,14         ; ROUND THE RESULT
        SACH ERRF,1
        LAC  X
        SACL FRSTAP         ; INCLUDE NEWEST SAMPLE
```

```
*
        LARK  AR0,LASTAP  ; POINT TO DATA SAMPLE
        LARK  AR1,COEFFD  ; POINT TO COEFFICIENTS
*
        LT    ERRF        ; KEEP ERRF IN T REGISTER
*
ADAPT   MPY   *-,AR1      ; P = 2*beta*err(i)*x(i-31)
        ZALH  *
        APAC              ; b31(i+1) = b31(i) + P
        ADD   ONE,15      ; ROUND
        SACH  *+,0,AR0    ; STORE b31(i+1)
*
        MPY   *-,AR1      ; P = 2*beta*err(i)*x(i-30)
        ZALH  *
        APAC              ; b30(i+1) = b30(i) + P
        ADD   ONE,15      ; ROUND
        SACH  *+,0,AR0    ; STORE b30(i+1)
*
        MPY   *-,AR1      ; P = 2*beta*err(i)*x(i-29)
        ZALH  *
        APAC              ; b29(i+1) = b29(i) + P
        ADD   ONE,15      ; ROUND
        SACH  *+,0,AR0    ; STORE b29(i+1)
          .
          .
          .
        MPY   *-,AR1      ; P = 2*beta*err(i)*x(i-0)
        ZALH  *
        APAC              ; b0(i+1) = b0(i) + P
        ADD   ONE,15      ; ROUND
        SACH  *+,0,AR0    ; STORE b0(i+1)
*
        RET               ; RETURN TO MAIN PROGRAM
```

5.6.4 Fast Fourier Transforms (FFT)

Fourier transforms are another important tool often used in digital signal processing systems. The purpose of the transform is to convert information from the time domain to the frequency domain. The inverse Fourier transform converts information back to the time domain from the frequency domain. Implementations of Fourier transforms that are computationally efficient are known as Fast Fourier Transforms (FFTs). The theory and implementation of FFTs has been discussed in the book, *DFT/FFT and Convolution Algorithms,* by Burrus and Parks, published by John Wiley and Sons. The book also contains a large number of sample TMS32010 and FORTRAN programs to implement DFT/FFT algorithms. The TMS320C1x reduces the execution time of all FFTs by virtue of its single-cycle instruction time.

Example 5-36 consists of some of the macros used in the implementation of FFTs. Example 5-37 provides the code for an 8-point DIT (decimation in time) FFT. The code has been structured into a number of macro calls, including a macro for bit reversal.

Example 5-40. FFT Macros

```
COMBO   $MACRO R1,I1,R2,I2,R3,I3,R4,I4
*
* CALCULATE PARTIAL TERMS FOR R3, R4, I3, AND I4.
*
        LAC    :R3:,14   ;ACC  := (1/4)(R3)
        ADD    :R4:,14   ;ACC  := (1/4)(R3+R4)
        SACH   :R3:,1    ;R3   := (1/2)(R3+R4)
        SUB    :R4:,15   ;ACC  := (1/4)(R3+R4)-(1/2)(R4)
        SACH   :R4:,1    ;R4   := (1/2)(R3-R4)
        LAC    :I3:,14   ;ACC  := (1/4)(I3)
        ADD    :I4:,14   ;ACC  := (1/4)(I3+I4)
        SACH   :I3:,1    ;I3   := (1/2)(I3+I4)
        SUB    :I4:,15   ;ACC  := (1/4)(I3+I4)-(1/2)(I4)
        SACH   :I4:,1    ;I4   := (1/2)(I3-I4)
*
* CALCULATE PARTIAL TERMS FOR R2, R4, I2, AND I4.
*
        LAC    :R1:,14   ;ACC  := (1/4)(R1)
        ADD    :R2:,14   ;ACC  := (1/4)(R1+R2)
        SACH   :R1:,1    ;R1   := (1/2)(R1+R2)
        SUB    :I3:,15   ;ACC  := (1/4)(R1+R2)-(1/2)(R2)
        ADD    :I4:,15   ;ACC  := (1/4)[(R1-R2)+(I3-I4)]
        SACH   :R2:      ;R2   := (1/4)[(R1-R2)+(I3-I4)]
        SUBH   :I4:      ;ACC  := (1/4)[(R1-R2)-(I3-I4)]
        DMOV   :R4:      ;I4   := R4 = (1/2)(R3-R4)
        SACH   :R4:      ;R4   := (1/4)[(R1-R2)-(I3-I4)]
        LAC    :I1:,14   ;ACC  := (1/4)(I1)
        ADD    :I2:,14   ;ACC  := (1/4)(I1+I2)
        SACH   :I1:,1    ;I1   := (1/2)(I1+I2)
        SUB    :I2:,15   ;ACC  := (1/4)(I1+I2)-(1/2)(I2)
        SUB    :I4:,15   ;ACC  := (1/4)[(I1-I2)-(I3-I4)]
        SACH   :I2:      ;I2   := (1/4)[(I1-I2)-(I3-I4)]
        ADDH   :I4:      ;ACC  := (1/4)[(I1-I2)+(I3-I4)]
        SACH   :I4:      ;I4   := (1/4)[(I1-I2)+(I3-I4)]
*
* CALCULATE PARTIAL TERMS FOR R1, R3, I1, AND I3
*
        LAC    :R1:,15   ;ACC  := (1/4)(R1+R2)
        ADD    :R3:,15   ;ACC  := (1/4)[(R1+R2)+(R3+R4)]
        SACH   :R1:      ;R1   := (1/4)[(R1+R2)+(R3+R4)]
        SUBH   :R3:      ;ACC  := (1/4)[(R1+R2)-(R3+R4)]
        SACH   :R3:      ;R3   := (1/4)[(R1+R2)-(R3+R4)]
        LAC    :I1:,15   ;ACC  := (1/4)(I1+I2)
        ADD    :I3:,15   ;ACC  := (1/4)[(I1+I2)+(I3+I4)]
        SACH   :I1:      ;I1   := (1/4)[(I1+I2)+(I3+I4)]
        SUBH   :I3:      ;ACC  := (1/4)[(I1+I2)-(I3+I4)]
        SACH   :I3:      ;I3   := (1/4)[(I1+I2)-(I3+I4)]
        $END
*
* MACRO FOR INPUT BIT REVERSAL.
*
BITREV  $MACRO PR,PI,QR,QI
        ZALH   :PR:
        ADDS   :QR:
        SACL   :PR:
        SACH   :QR:
        ZALH   :PI:
        ADDS   :QI:
        SACL   :PI:
        SACH   :QI:
        $END
```

```
*
ZERO    $MACRO PR,PI,QR,QI
*
* CALCULATE Re(P+Q) AND Re(P-Q)
*
        LAC    :PR:,15   ;ACC  := (1/2)(PR)
        ADD    :QR:,15   ;ACC  := (1/2)(PR+QR)
        SACH   :PR:      ;PR   := (1/2)(PR+QR)
        SUBH   :QR:      ;ACC  := (1/2)(PR+QR)-(QR)
        SACH   :QR:      ;QR   := (1/2)(PR-QR)
*
* CALCULATE Im(P+Q) AND Im(P-Q)
*
        LAC    :PI:,15   ;ACC  := (1/2)(PI)
        ADD    :QI:,15   ;ACC  := (1/2)(PI+QI)
        SACH   :PI:      ;PR   := (1/2)(PI+QI)
        SUBH   :QI:      ;ACC  := (1/2)(PI+QI)-(QI)
        SACH   :QI:      ;QR   := (1/2)(PI-QI)
        $END
*
PIBY4   $MACRO PR,PI,QR,QI,W
*
        LT     :W:       ;T REG:= W=COS(PI/4)=SIN(PI/4)
        LAC    :QI:,14   ;ACC  := (1/4)(QI)
        SUB    :QR:,14   ;ACC  := (1/4)(QI-QR)
        SACH   :QI:,1    ;QI   := (1/2)(QI-QR)
        ADD    :QR:,15   ;ACC  := (1/4)(QI+QR)
        SACH   :QR:,1    ;QR   := (1/2)(QI+QR)
        LAC    :PR:,14   ;ACC  := (1/4)(PR)
        MPY    :QR:      ;P REG:= (1/4)(QI+QR)*W
        APAC             ;ACC  := (1/4)[PR+(QI+QR)*W]
        SACH   :PR:,1    ;PR   := (1/2)[PR+(QI+QR)*W]
        SPAC             ;ACC  := (1/4)(PR)
        SPAC             ;ACC  := (1/4)[PR-(QI+QR)*W]
        SACH   :QR:,1    ;QR   := (1/2)[PR-(QI+QR)*W]
        LAC    :PI:,14   ;ACC  := (1/4)(PI)
        MPY    :QI:      ;P REG:= (1/4)(QI-QR)*W
        APAC             ;ACC  := (1/4)[PI+(QI-QR)*W]
        SACH   :PI:,1    ;PI   := (1/2)[PI+(QI-QR)*W]
        SPAC             ;ACC  := (1/4)(PI)
        SPAC             ;ACC  := (1/4)[PI-(QI-QR)*W]
        SACH   :QI:,1    ;QI   := (1/2)[PI-(QI-QR)*W]
        $END
*
PIBY2   $MACRO PR,PI,QR,QI
*
* CALCULATE Re(P+jQ) AND Re(P-jQ)
*
        LAC    :PI:,15   ;ACC  := (1/2)(PI)
        SUB    :QR:,15   ;ACC  := (1/2)(PI-QR)
        SACH   :PI:      ;PI   := (1/2)(PI-QR)
        ADDH   :QR:      ;ACC  := (1/2)(PI-QR)+(QR)
        SACH   :QR:      ;QR   := (1/2)(PI+QR)
*
* CALCULATE Im(P+jQ) AND Im(P-jQ)
*
```

```
                LAC     :PR:,15    ;ACC    := (1/2)(PR)
                ADD     :QI:,15    ;ACC    := (1/2)(PR+QI)
                SACH    :PR:       ;PR     := (1/2)(PR+QI)
                SUBH    :QI:       ;ACC    := (1/2)(PR+QI)-(QI)
                DMOV    :QR:       ;QR     --> QI
                SACH    :QR:       ;QR     := (1/2)(PR-QI)
                $END
*
PI3BY4          $MACRO PR,PI,QR,QI,W
*
                LT      :W:        ;T REG := W=COS(PI/4)=SIN(PI/4)
                LAC     :QI:,14    ;ACC    := (1/4)(QI)
                SUB     :QR:,14    ;ACC    := (1/4)(QI-QR)
                SACH    :QI:,1     ;QI     := (1/2)(QI-QR)
                ADD     :QR:,15    ;ACC    := (1/4)(QI+QR)
                SACH    :QR:,1     ;QR     := (1/2)(QI+QR)
                LAC     :PR:,14    ;ACC    := (1/4)(PR)
                MPY     :QI:       ;P REG := (1/4)(QI-QR)*W
                APAC               ;ACC    := (1/4)[PR+(QI-QR)*W]
                SACH    :PR:,1     ;PR     := (1/2)[PR+(QI-QR)*W]
                SPAC               ;ACC    := (1/4)(PR)
                SPAC               ;ACC    := (1/4)[PR-(QI-QR)*W]
                MPY     :QR:       ;P REG := (1/4)(QI+QR)*W
                SACH    :QR:,1     ;QR     := (1/2)[PR-(QI-QR)*W]
                LAC     :PI:,14    ;ACC    := (1/4)(PI)
                SPAC               ;ACC    := (1/4)[PI-(QI+QR)*W]
                SACH    :PI:,1     ;PI     := (1/2)[PI-(QI+QR)*W]
                APAC               ;ACC    := (1/4)(PI)
                APAC               ;ACC    := (1/4)[PI+(QI+QR)*W]
                SACH    :QI:,1     ;QI     := (1/2)[PI+(QI+QR)*W]
                $END
```

Example 5-41. An 8-Point DIT FFT

```
* THIS ROUTINE IMPLEMENTS AN 8-POINT DIT FFT. ASSUME THAT
* TWIDDLE FACTOR = W VALUE STORED IN MEMORY LOCATION W.
*
X0R     .set    00
X0I     .set    01
X1R     .set    02
X1I     .set    03
X2R     .set    04
X2I     .set    05
X3R     .set    06
X3I     .set    07
X4R     .set    08
X4I     .set    09
X5R     .set    10
X5I     .set    11
X6R     .set    12
X6I     .set    13
X7R     .set    14
X7I     .set    15
W       .set    16
WVALUE  .set    5A82h      ; VALUE FOR SIN(45) OR COS(45)
*
* INITIALIZE FFT PROCESSING. ASSUME TWIDDLE FACTOR =
* W VALUE STORED IN MEMORY LOCATION W.
*
        .text
FFT     ROVM               ; RESET OVERFLOW MODE
        LDPK    0          ; SET DATA PAGE POINTER TO 0
*
* BIT-REVERSED INPUT SAMPLES.
*
        BITREV  X1R,X1I,X4R,X4I
        BITREV  X3R,X3I,X6R,X6I
*
* FIRST AND SECOND STAGES COMBINED WITH DIVIDE-BY-4
* INTERSTAGE SCALING.
*
        COMBO   X0R,X0I,X1R,X1I,X2R,X2I,X3R,X3I,
        COMBO   X4R,X4I,X5R,X5I,X6R,X6I,X7R,X7I.
*
* THIRD STAGE WITH DIVIDE-BY-2 INTERSTAGE SCALING.
*
        ZERO    X0R,X0I,X4R,X4I
        PIBY4   X1R,X1I,X5R,X5I,W
        PIBY2   X2R,X2I,X6R,X6I
        PI3BY4  X3R,X3I,X7R,X7I,W
```

5.6.5 PID Control

Control systems are concerned with regulating a process and achieving a desired behaviour or output from the process. A control system consists of three main components: sensors, actuators, and a controller. Sensors measure the behavior of the system. Actuators supply the driving force to ensure the desired behaviour. The controller generates actuator commands corresponding to the error conditions observed by the sensors and the control algorithms programmed in the controller. The controller typically consists of an analog or digital processor.

Analog control systems are usually based on fixed components and are not programmable. They are also limited to using single-purpose characteristics of the error signal, such as P (proportional), I (integral), and D (derivative) or their combination. These limitations, along with other disadvantages o analog systems such as component aging and temperature drift, are causing digital control systems to increasingly replace analog systems in most control applications.

Digital control systems that use a microprocessor/microcontroller are able to implement more sophisticated algorithms of modern control theory, such as state models, deadbeat control, state estimation, optimal control, and adaptive control. Digital control algorithms deal with the processing of digital signals and are similar to DSP algorithms. The TMS320C1x instruction set can therefore be used very effectively in digital control systems.

The most commonly used algorithm in both analog and digital control systems is the PID (Proportional, Integral, and Derivative) algorithm. The classical PID algorithm is given by

$$u(t) = K_p\, e(t) + K_i \int edt + K_d\, de/dt$$

The PID algorithm must be converted into a digital form for implementation on a microprocessor. Using a rectangular approximation for the integral, the PID algorithm can be approximated as

$$u(n) = u(n\text{-}1) + K_1\, e(n) + K_2\, e(n\text{-}1) + K_3\, e(n\text{-}2)$$

This algorithm is implemented in Example 5-42.

Example 5-38. PID Control

```
        .title   'PID CONTROL'
        .def     PID
*
* THIS ROUTINE IMPLEMENTS A PID ALGORITHM.
*
UN      .set 0       ; OUTPUT OF CONTROLLER
E0      .set 1       ; LATEST ERROR SAMPLE
E1      .set 2       ; PREVIOUS ERROR SAMPLE
E2      .set 3       ; OLDEST ERROR SAMPLE
K1      .set 4       ; GAIN CONSTANT
K2      .set 5       ; GAIN CONSTANT
K3      .set 6       ; GAIN CONSTANT
*
* ASSUME DATA PAGE 0 IS SELECTED.
*
        .text
PID     IN    E0,PA0 ; READ NEW ERROR SAMPLE
        LAC   UN     ; ACC = u(n-1)
        LT    E2     ; LOAD T REG WITH OLDEST SAMPLE
        MPY   K2     ; P = K2*e(n-2)
        LTD   E1     ; ACC = u(n-1)+K2*e(n-2)
        MPY   K1     ; P = K1*e(n-1)
        LTD   E0     ; ACC = u(n-1)+K1*e(n-1)+K2*e(n-2)
        MPY   K0     ; P = K0*e(n)
        APAC         ; ACC = u(n-1)+K0*e(n)+K1*e(n-1)
*                    ;       +K2*e(n-2)
        SACH  UN,1   ; STORE OUTPUT
        OUT   UN,PA1 ; SEND IT
```

The PID loop takes 13 cycles to execute or 2.6 μs at a 20-MHz clock rate. The TMS320 can also be used to implement more sophisticated algorithms such as state modeling, adaptive control, state estimation, Kalman filtering, and optimal control. Other functions that can be implemented are noise filtering, stability analysis, and additional control loops.

5.6.6 Selftest Routines

A selftest program can effectively perform incoming quality verification or be used as a powerup device verification tool. Texas Instruments has developed a selftest program to check out the functionality of a TMS320C1x device before branching to the user code. This program is not intended to provide a means of logic debug but rather to indicate device pass/fail from which it can be determined whether or not the TMS320C1x is still functional.

When designing a DSP device, Texas Instruments runs very thorough patterns through the logic to test all the stages. In these patterns, worst-case conditions and transitions are forced in order to verify logic design prior to manufacturing. Likewise, the speed and electrical specifications are thoroughly tested. In production manufacturing, every TMS320C1x is tested to meet the functionality, speed, and power specifications of the device before it is shipped. The drive levels and loading of lines are checked at full speed and over varying temperature.

The 460-word selftest program for the TMS320C1x exercises most of the on-chip resources of the device with a minimal amount of external circuitry. Note that this code is intended for testing on-chip resources and will not exercise the external interface lines.

Example 5-43 contains a small portion of this selftest program, which checks out the ALU section. The ALU test is designed to validate the basic operation of the circuit. It consists of a series of subtests to verify addition and subtraction operations of both halves of the 32-bit operation as well as carry and overflow calculations, absolute value, and SUBC operation. A failure in any of these tests will set the error code in the accumulator to 100Xh where X is the number of the subtest that has failed.

Other sections of this selftest check the auxiliary registers, on-chip data RAM, on-chip program ROM (longitudinal redundancy test), status register and branches, pre- and post-scaling shifters, multiplier, and the instruction set.

An applications brief is available which discusses the code segments that comprise the TMS320C1x selftest program as well as how to link and execute this code. The applications brief and selftest code are available via the TMS320 DSP Bulletin Board Service (see Appendix E).

Example 5-43. Selftest Routine

```
* THIS PROGRAM EXECUTES AN INTERNAL SELFTEST OF THE TMS320C1X
* MICROCOMPUTER ALU. A FAILURE IN ANY OF THESE TESTS WILL SET
* THE ERROR CODE IN THE ACCUMULATOR TO 100Xh WHERE X IS THE
* NUMBER OF THE SELFTEST THAT HAS FAILED.
*
*
* RESET AND INTERRUPT VECTORS.
*
BEGIN  B     START  ; RESET SOFT VECTOR
       B     INTRPT ; INTERRUPT SOFT VECTOR
*
* REQUIRED DATA VALUES FOR TEST PROGRAMS.
*
       .word 0FFFFh ; RAM TEST PATTERN 1
       .word 0AAAAh ; RAM TEST PATTERN 2
       .word 5555h  ; RAM TEST PATTERN 3
       .word 0h     ; RAM TEST PATTERN 4
*
* PROGRAM INITIALIZATION DP = 0 AND DISABLE INTERRUPTS.
*
       .text
START               ; START INITIALIZATION ROUTINE
       LDPK  0      ; START IN ZERO DATA PAGE
       DINT         ; DISABLE EXTERNAL INTERRUPTS
*
* ARITHMETIC LOGIC UNIT TEST.
*
ALU
       LACK  1      ; GET INCREMENT VALUE
       SACL  8      ; STORE IT IN REG8
       LACK  4      ; POINT ACC TO PATTERNS TABLE
       TBLR  4      ; PUT TABLE VALUE IN REG4
       ADD   8      ; INCREMENT TABLE ADDRESS
       TBLR  5      ; PUT TABLE VALUE IN REG5
       ADD   8      ; INCREMENT TABLE ADDRESS
       TBLR  6      ; PUT TABLE VALUE IN REG6
       ADD   8      ; INCREMENT TABLE ADDRESS
       TBLR  7      ; PUT TABLE VALUE IN REG7
       LACK  10h    ; SET ERROR CODE VALUE
       SACL  2      ; STORE CODE IN REG2
*
ALU1   ZAC          ; CLEAR OUT ACCUMULATOR
       ADDS  5      ; ADD IN 0AAAAh PATTERN
       AND   5      ; AND WITH 0AAAAh PATTERN
       OR    6      ; OR WITH 5555h PATTERN
       SUBS  4      ; SUBTRACT -1 FROM PATTERN
       BZ    ALU2   ; IF ACC CLEARED, GO TO NEXT TEST
*
       LACK  1      ; IF NOT, THEN SET TEST 1 CODE
       ADD   2,8    ; ADD IN ERROR CODE
       B     ERROR  ; EXIT TO ERROR ROUTINE
*
ALU2   ZALH  5      ; ADD HIGH THE 0AAAAh PATTERN
       ADDH  6      ; SUBTRACT HIGH THE 5555h PATTERN
       SACH  0      ; SAVE THE VALUE
       ZALH  0      ; RESTORE THE VALUE
       ABS          ; TAKE ABSOLUTE VALUE
       SUBH  8      ; SUBTRACT HIGH 10000h
       BZ    ALU3   ; IF ACC CLEARED, GO TO NEXT TEST
```

```
*
        LACK   2       ; IF NOT, THEN SET TEST 2 CODE
        ADD    2,8     ; ADD IN ERROR CODE
        B      ERROR   ; EXIT TO ERROR ROUTINE
*
ALU3    LAC    4,12    ; LOAD ACC WITH 0FFFFF000h PATTERN
        ADD    8,12    ; ADD 00001000h TO IT
        BZ     ALU4    ; IF ACC CLEARED, GO TO NEXT TEST
*
        LACK   3       ; IF NOT, THEN SET TEST 3 CODE
        ADD    2,8     ; ADD IN ERROR CODE
        B      ERROR   ; EXIT TO ERROR ROUTINE
*
ALU4    ADD    4       ; LOAD ACC WITH 0FFFFFFFFh PATTERN
        ABS            ; TAKE ABSOLUTE VALUE
        SUB    8       ; SUBTRACT 00000001h
        BZ     ALU5    ; IF ACC CLEARED, GO TO NEXT TEST
*
        LACK   4       ; IF NOT, THEN SET TEST 4 CODE
        ADD    2,8     ; ADD IN ERROR CODE
        B      ERROR   ; EXIT TO ERROR ROUTINE
*
ALU5    LACK   40h     ; GET DIVISOR = 64
        SACL   0       ; SAVE IN REG0
        LACK   0FFh    ; GET DIVIDEND = 255
        SUBC   0       ; 1ST STAGE OF DIVIDE
        NOP            ; REQUIRED NOP
        SUBC   0       ; 2ND STAGE OF DIVIDE
        NOP            ; REQUIRED NOP
          .
          .
          .
        SUBC   0       ; 16TH STAGE OF DIVIDE
        NOP            ; REQUIRED NOP
        SACH   1       ; SAVE REMAINDER
        SACL   2       ; SAVE QUOTIENT
        LACK   3       ; GET QUOTIENT COMPARISON MASK
        XOR    2       ; COMPARE WITH CALCULATED ANSWER
        BZ     ALU6    ; IF ACC CLEARED, GO TO NEXT TEST
*
        LACK   5       ; IF NOT, THEN SET TEST 5 CODE
        ADD    2,8     ; ADD IN ERROR CODE
        B      ERROR   ; EXIT TO ERROR ROUTINE
*
ALU6    LACK   3Fh     ; GET REMAINDER COMPARISON MASK
        XOR    1       ; COMPARE WITH ANSWER
        BZ     STATUS  ; IF ACC CLEARED, GO TO NEXT TEST
*
        LACK   6       ; IF NOT, THEN SET TEST 6 CODE
        ADD    2,8     ; ADD IN ERROR CODE
        B      ERROR   ; EXIT TO ERROR ROUTINE
```

Section 6

Hardware Applications

Information and examples on interfacing a TMS320C1x (first-generation TMS320) digital signal processor with external devices are presented in this section. The examples given are general enough in nature that they may be easily adapted to fit a particular system requirement.

The following buses, ports, and control signals provide system interface to the TMS320C1x processor:

- 12-bit address bus (A11-A0)
- 16-bit data bus (D15-D0)
- 3-bit port address bus
- Memory control signals (MC/$\overline{\mathrm{MP}}$ or MC/$\overline{\mathrm{PM}}$)
- Reset ($\overline{\mathrm{RS}}$)
- Interrupt ($\overline{\mathrm{INT}}$) and branch control ($\overline{\mathrm{BIO}}$)
- Enable signals ($\overline{\mathrm{DEN}}$, $\overline{\mathrm{MEN}}$, and $\overline{\mathrm{WE}}$)
- External flag (XF))
- Serial port clock (SCLK)
- Serial port receive/transmit channel inputs/outputs (DR/DX)
- Serial port framing inputs and output ($\overline{\mathrm{FSR}}$, $\overline{\mathrm{FSX}}$, and FR)
- Coprocessor port read/write signals ($\overline{\mathrm{RD}}$/$\overline{\mathrm{WR}}$)
- Coprocessor latch signals ($\overline{\mathrm{TBLF}}$/$\overline{\mathrm{RBLE}}$).

Major hardware applications discussed in this section are listed below.

- Expansion Memory Interface (Section 6.1 on page 6-2)
  - Program ROM expansion
  - Data RAM expansion
- Codec Interface (Section 6.2 on page 6-6)
- A/D and D/A Interface (Section 6.3 on page 6-8)
- I/O Ports (Section 6.4 on page 6-10)
- Coprocessor Interface (Section 6.5 on page 6-11)
- System Applications (Section 6.6 on page 6-13)
  - 2400 bps modem
  - Speech synthesis system
  - Voice store-and-forward message system.

6.1 Expansion Memory Interface

The TMS320C1x can be interfaced to a wide variety of memory and I/O devices. The TMS32010/C10 and TMS320C15/E15 devices can be interfaced to up to 4K words of external program memory. Expansion of program memory is accomplished directly through the use of the $\overline{\text{MEN}}$ (memory enable) and $\overline{\text{WE}}$ (write enable) control lines, with memory accesses occurring in a single cycle.

6.1.1 Program ROM Expansion

Twelve TMS32010 output pins (A11-A0) are available for addressing external memory. They contain either the buffered outputs of the program counter or the I/O port address.

Read operations are performed on external memory either during opcode or operand fetches or during the execution of a TBLR (table read) instruction. Write operations have no effect on the circuit. When a read operation occurs, an address is placed on the address bus, and the $\overline{\text{MEN}}$ (memory enable) strobe is generated by driving $\overline{\text{MEN}}$ low to enable external memory. The instruction word is then transferred to the TMS32010 via the 16-bit data bus.

A memory address being placed on the bus becomes valid following a maximum delay (t_{d1}) from the falling edge of CLKOUT. The combined delay of:

$$t_{d1} + t_{a(A)} + t_{su(D)} = \text{minimum cycle time } t_{c(C)}$$

where $t_{a(A)}$ = memory access time of EPROM from address valid
$t_{su(D)}$ = setup time form data bus valid prior to CLKOUT↓

serves as the timing constraint used when calculating $t_{c(C)}$.

When only external program ROM is required, a minimum system can consist of a TMS320C10/C15 and up to 4K words of external program memory (TMS27C292), as shown in Figure 6-1. The $\overline{\text{MEN}}$ signal and the address (A11-A0) and data (D15-D0) lines on the TMS320C10/C15/E15 are connected directly to the TMS27C292 memories, and no address decoding is required. These memories are a pair of TMS27C292 4K x 8 ROMs by Texas Instruments, configured in parallel for a direct 16-bit interface to the TMS320C10/C15/E15.

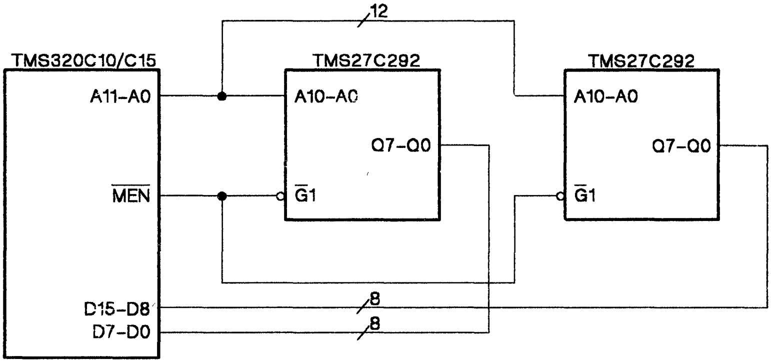


Figure 6-1. Minimum Program ROM Expansion

An inexpensive system with minimal chip-count is possible when using the TMS320C10-14. The usage of an EPROM, interfaced to a TMS320C10-14, in external program memory allows the implementation of 4K words of non-volatile program memory along with the added flexibility in reprogramming, thus, providing for system development, future program expansion, and/or upgrade modification. Single-cycle memory access using a direct memory interface requires no additional external interface logic.

On the TMS320C10-14, t_{d1} with a maximum value of 50 ns and $t_{su(D)}$ with a minimum value of 50 ns are both constants; therefore, $t_{a(A)}$ is the only remaining variable used in determining the minimum clock cycle time of the system. For the circuit shown in Figure 6-2 (with $t_{a(A)}$ = 170 ns), inserting these values into the equation yields $t_{c(C)}$ min = 270 ns.

In Figure 6-2, a pair of Texas Instruments TMS2732A-17 4K x 8 EPROM memories are configured in parallel for a direct 16-bit interfacing with TMS320C10-14. These EPROMs display a 170-ns access time. However, other EPROMs may be used with access times best suited to a particular application as long as the TMS320C10-14 clock frequency has been selected to allow for the access time of the EPROMs chosen.

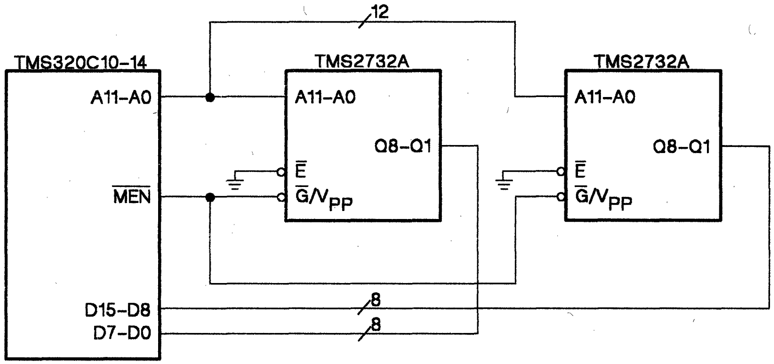


Figure 6-2. EPROM Interface to the TMS320C10-14

Contention for the data bus is not a concern in this memory configuration. Therefore, the Ē (chip enable) pin for the EPROM pair has been tied to ground to avoid unnecessary switching transients that could be induced if the chip enables were toggled upon memory access.

6.1.2 Data RAM Expansion

No direct memory expansion is provided on the TMS320C1x. However, if RAM is used for external program memory, this memory can be used to store data information, accessed using the TBLR and TBLW instructions. These instructions, however, take three cycles to execute.

If larger memory or faster memory accesses are required, an alternative memory expansion scheme using I/O ports can be implemented for a TMS320C1x device. In this case, additional RAM can be used to supplement internal data memory, and can be accessed in only two cycles using the IN and OUT instructions. If RAM is to be used for program memory, additional logic must be included to distinguish between an I/O write (OUT) and a program memory write (TBLW).

Figure 6-3 provides an example of external data memory expansion. The design consists of up to 16K words of static RAM (IMS1420), addressed by the lower 14 bits of a 16-bit counter (74ALS193). In the case of the IMS1420s, the address of the data to be accessed is loaded into the counter by implementing an OUT instruction to port 0. This loads the data bus into the counters. Memory can then be read from or written to sequentially by doing an IN or OUT instruction to port 1. The MSB in the counters determines whether the memory address is incremented (MSB = 0) or decremented (MSB = 1) after a read or write of data memory. Memory continues to be addressed sequentially until new data is loaded into the counters.

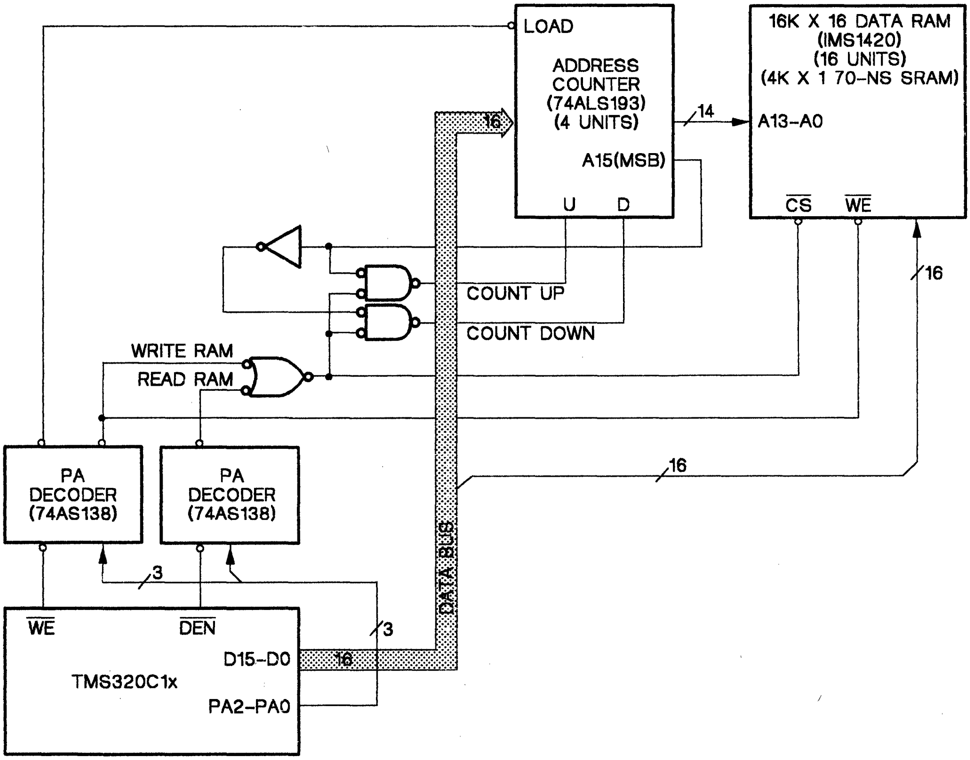


Figure 6-3. Data RAM Expansion

Dynamic memories may also be used; however, these devices may impose additional constraints on the system designer. For example, some memory cycle times may not allow consecutive IN/OUT/IN instruction sequences. Memory refresh must also be considered. Since the TMS320C1x does not implement "wait" states, memory refresh must be generated transparent to the processor.

For additional information regarding interfacing to TMS320C1x devices, refer to *Digital Signal Processing Applications with the TMS320 Family* (literature number SPRA012A).

6.2 Codec Interface

In areas of telecommunications, speech processing, and other applications that require low-cost analog I/O devices, a combo-codec may be useful. A combo-codec consists of nonlinear A/D and D/A converters with antialiasing and smoothing filters and data storage registers. For additional information on combo-codecs, refer to *TCM29C13/C14/C16/C17 Combined Single-Chip PCM Codec and Filter Data Sheet.*

The TMS320C17/E17 is capable of direct interface to serial devices such as combo-codecs, thus reducing chip count and improving system throughput. These TMS320 devices can also compand (COMpress and exPAND) a PCM (Pulse Code Modulation) data stream, acquired by the codec, through the use of on-chip companding hardware.

Figure 6-4 shows the TMS320C17/E17 interfaced to a TCM29C13 combo-codec to demonstrate direct serial-port interface capability. A standalone full-duplex serial interface is shown, in which the TMS320C17/E17 provides the serial clock for bit transmission. The codec is sampled every 125 μs (8-kHz frequency), at which time an 8-bit PCM byte is exchanged between the two devices. A second codec can also be interfaced to the TMS320C17/E17 with no additional logic or interconnections since these devices implement two independent serial ports.

Timing for the serial interface system is controlled by the serial-port clock (SCLK). SCLK is configured as an output from the TMS320C17/E17, and its frequency is set to 2.048 MHz (see Section 3.9). A 20.48-MHz crystal is input to the TMS320 as its system clock. The SCLK frequency is derived from this system clock by a divide-by-10 in the SCLK prescale control logic, initialized through control register 1. SCLK is connected to CLKR/CLKX on the TCM29C13 to provide the transmit and receive master clock. CLKSEL on the codec is tied to V_{CC} to select the 2.048-MHz master clock mode.

Framing pulses are generated by the TMS320C17/E17 on the FR output pin. The frequency of these pulses is set to 8 kHz by dividing the serial clock (SCLK) by 256. This value is also initialized through control register 1. The short FR framing pulses provide the codec with framing pulses for the fixed data-rate mode. FR is input to both the FSX and FSR inputs on the codec. The FR output causes simultaneous transmit and receive operations from the serial port. The FSX input on the codec causes the device to transmit PCM data on the next eight consecutive positive transitions of the serial-port clock (SCLK). The FSR input on the codec causes the device to receive PCM data on the next eight consecutive negative transitions of the serial-port clock (SCLK). With this timing, the codec transmits and receives one 8-bit PCM sample every 125 μs.

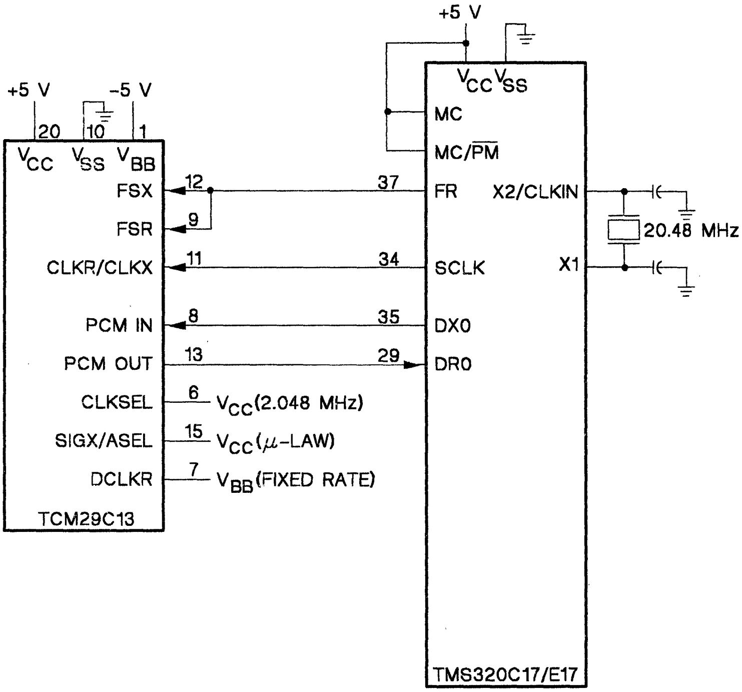


Figure 6-4. Codec Interface for Standalone Serial Operation

The TMS320C17/E17 transmits its PCM sample via the DX0 pin. The sample is received by the TCM29C13 on the PCM IN pin. The TMS320 receives PCM samples on its DR0 pin, which is the output of the PCM OUT pin of the TCM29C13. With this setup, single-channel operation is realized with the TMS320C17/E17. All data transmission occurs on channel 0, requiring one IN instruction from port 1 to receive the PCM sample and one OUT instruction to port 1 to send a sample to the codec.

In the serial interface configuration, μ-255 law companding is selected by setting system control register bit 14 (CR14) to logic 0. The TCM29C13 is put into the μ-law companding mode by connecting the SIGX/ASEL pin to V_{CC}.

Linear A/D and D/A converters may also be interfaced to the TMS320C17/E17 through its parallel ports instead of using the serial port.

6.3 A/D and D/A Interface

The TMS320C10/C15/E15/C17/E17 can be interfaced to A/D (analog-to-digital) and D/A (digital-to-analog) converters to perform the necessary conversions. A minimum of external circuitry is required.

Figure 6-5 shows an interface of the TLC0820 8-bit A/D converter to the TMS320C10/C15/E15/C17/E17. Since the control circuitry of the TLC0820 operates much more slowly than the TMS320C10/C15/E15/C17/E17, it cannot be directly interfaced. All of the logic functions are implemented with one each of the following devices from the 74ALS family of Advanced Low-power Schottky Logic:

74ALS679	12-bit address comparator
74LS74	Dual positive edge-triggered D-type flip-flops
74ALS465	Octal buffer with three-state output
74LS32	Quad two-input OR-gate.

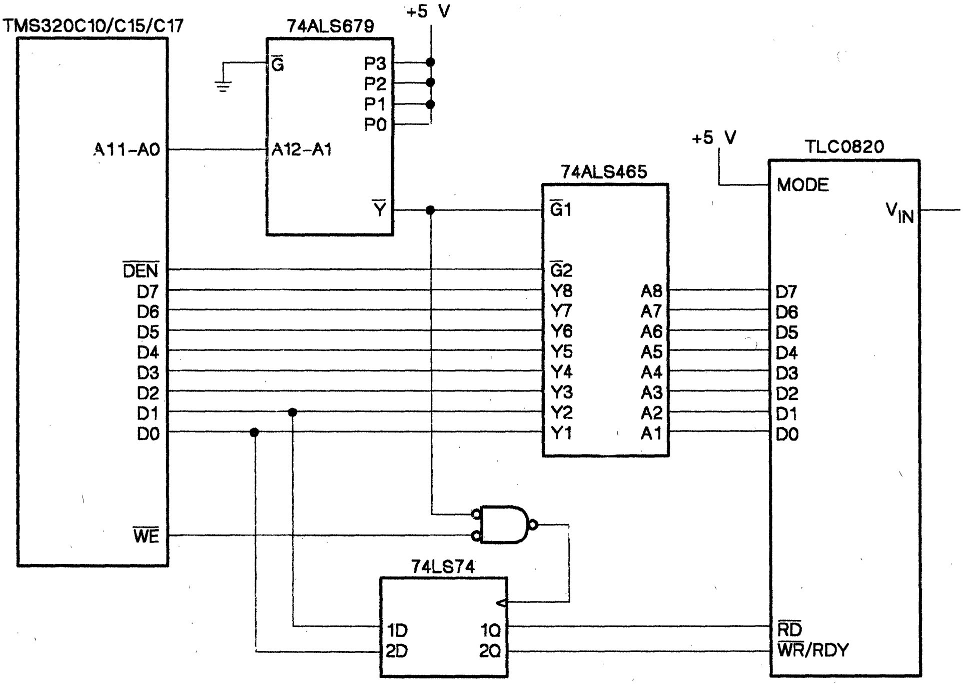
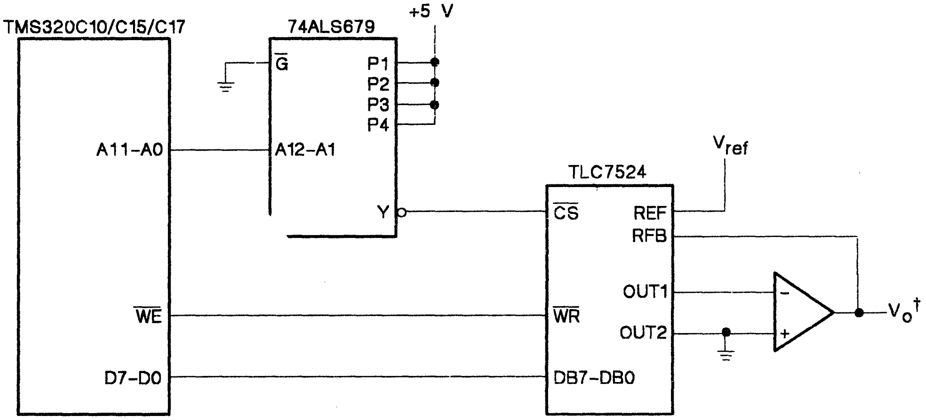


Figure 6-5. A/D Converter to TMS320C10/C15/E15/C17/E17 Interface

An interface of the TLC7524 8-bit D/A converter to the TMS320C10/C15/E15/C17/E17 is shown in Figure 6-6. Due to the high-speed operation of the internal logic circuitry of the TLC7524, the interface to the TMS320C10/C15/E15/C17/E17 requires external logic circuitry to decode the address of the peripheral. Here a 74ALS679 12-bit address comparator is used.



† $V_O = -V_{ref}\frac{D}{256}$, where D = digital input

Figure 6-6. D/A Converter to TMS320C10/C15/E15/C17/E17 Interface

For further information about the A/D and D/A converters shown in the figures, refer to *Linear Circuits Data Book* (literature number SLYD001).

6.4 I/O Ports

The TMS320C1x devices interface to input/output (I/O) devices through the eight 16-bit parallel ports (see Section 3.7 for I/O functions). The I/O space is selected by the $\overline{\mathrm{DEN}}$ signal for reads and the $\overline{\mathrm{WE}}$ signal for writes. Each of the eight I/O ports is addressed by the three LSBs of the address bus with all other address lines held low. The I/O ports share the 16 data lines.

The I/O ports may be used for interfacing external circuitry such as data memory expansion devices (see Section 6.1), A/D and D/A converters, synchronization latches, or memory-mapped peripheral devices. Figure 6-7 shows a circuit that can be used to generate device select lines for each of the individual port writes. A similar circuit may be used to enable I/O port reads.

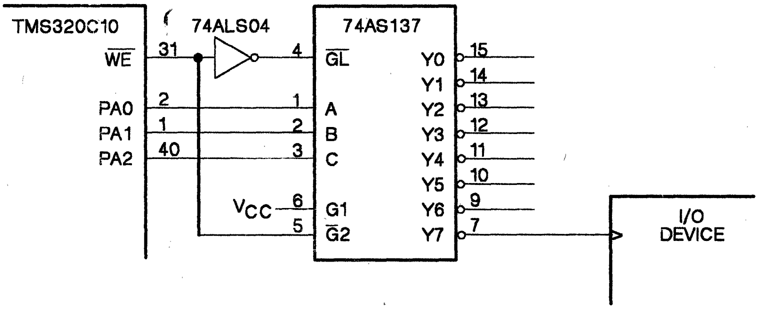


Figure 6-7. I/O Port Interface Circuit

When interfacing the TMS320C1x to slower devices, a handshake interface used in conjunction with the I/O port interface may be desirable. Data to be transferred may be stored in latches to be read by the TMS320C1x at a later time. Handshaking may then be established using the interrupt, $\overline{\mathrm{BIO}}$, and XF (TMS320C17/E17) signals.

6.5 Coprocessor Interface

The TMS320C17/E17 includes an option to use the parallel I/O interface exclusively as a coprocessor interface. This option includes both the buffer logic to communicate between two processors asynchronously, and the protocol logic to protect against poor communication. This port allows the TMS320C17/E17 to act as either a master processor or a slave processor in a multiprocessing system. The circuit also allows data to be transferred as either 8 or 16-bit values.

As a master processor, the TMS320C17/E17 writes to and reads from the coprocessor interface at will. This requires that the slave processor keep the receive buffer full and the transmit buffer empty. Figure 6-8 shows the TMS320C17 as a master processor to a TMS70C42 (8-bit microcomputer). As the internal CPU writes to the coprocessor interface, the $\overline{\text{TBLF}}$ (transmit buffer latch full) signal is driven active low. This signals the TMS70C42 that there is data to be read and that the 8-bit microcomputer must read that data before the next write by the internal CPU. In Figure 6-8, the $\overline{\text{TBLF}}$ signal is tied to an I/O bit on the 8-bit microcomputer so that the microcomputer can poll the signal and act accordingly. This signal could also be tied to an interrupt on the 8-bit microcomputer if this better suited system requirements. When the internal CPU reads its buffer, it signals the 8-bit microcomputer that the read buffer is empty by generating the $\overline{\text{RBLE}}$ (read buffer latch empty) signal. This signals the microcomputer that it must reload the receive latch before the next internal CPU access.

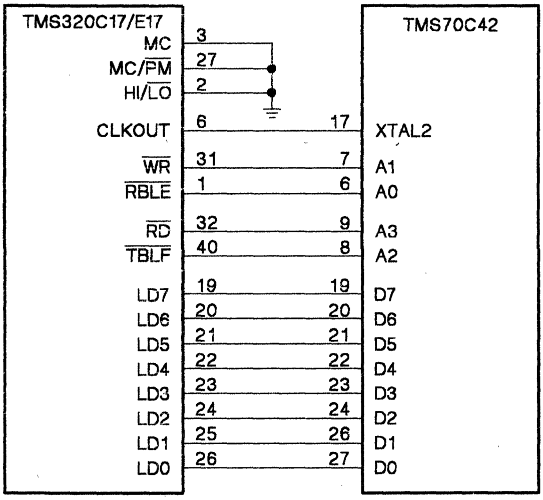


Figure 6-8. TMS320C17/E17 to TMS70C42 Interface

When the TMS320C17/E17 serves as a slave processor, transfer of data is controlled by a master processor. Figure 6-9 shows how TMS320C17/E17 (slave) interfaces with the 16-bit microprocessor TMS320C25 (master). When TMS320C25 writes to TMS320C17/E17, an interrupt signal is sent from the master to the internal CPU of the slave. The CPU must then read the information stored in the coprocessor interface before the next write from the TMS320C25. When the TMS320C25 reads the transfer latch of the coprocessor port, the internal CPU of the slve receives an active low $\overline{\text{BIO}}$ signal. When transferring information to the master processor, the internal CPU monitors the $\overline{\text{BIO}}$ line (using the BIOZ instruction) to determine when it can reload the transmit latch. Note that a wait state may be required when interfacing to the TMS320C25.

To support mixed 8/16-bit operation, the read buffer latch is cleared to 0 when read by the internal CPU.

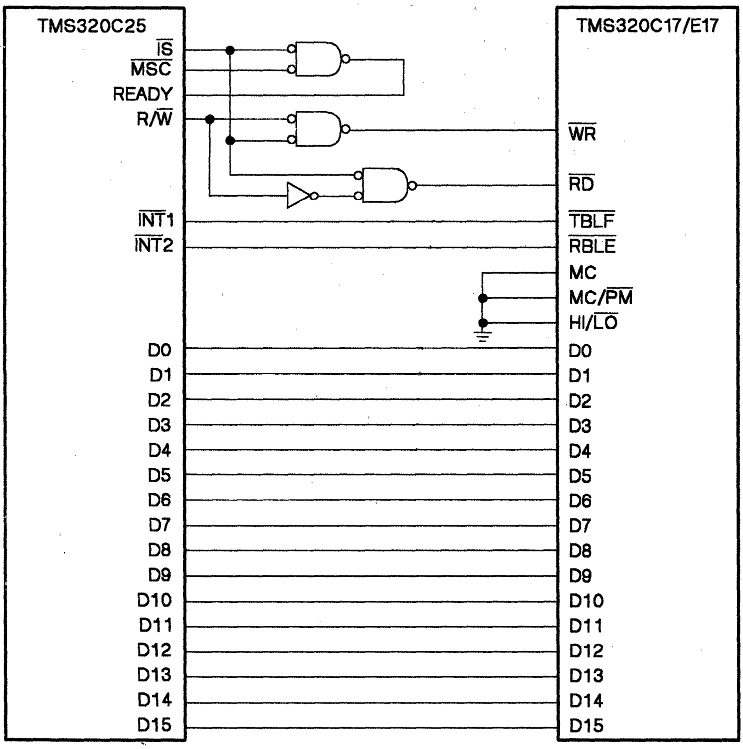


Figure 6-9. TMS320C17/E17 to TMS320C25 Interface

6.6 System Applications

The TMS320C1x devices are commonly used in many system applications. Several of these system applications are presented in this section, in a general form, to illustrate basic approaches to system design using the TMS320C1x. These applications include a 2400 bps modem, a speech synthesis system, and a voice store-and-forward message center.

6.6.1 2400 bps Modem

The implementation of a 2400 bps modem is shown in Figure 6-10. This system implements the functions of a V.22 bis modem using a TMS320A2400 and a TMS70A2400, which are masked ROM versions of the TMS320C17 and TMS7042, respectively. The TMS320A2400 performs all of the signal processing functions, and the TMS70A2400 performs all of the interface protocol and control functions. The remaining system components perform the necessary analog-to-digital (A/D) and digital-to-analog (D/A) conversions as well as the PC bus interfacing, telephone line interfacing, and filtering functions.

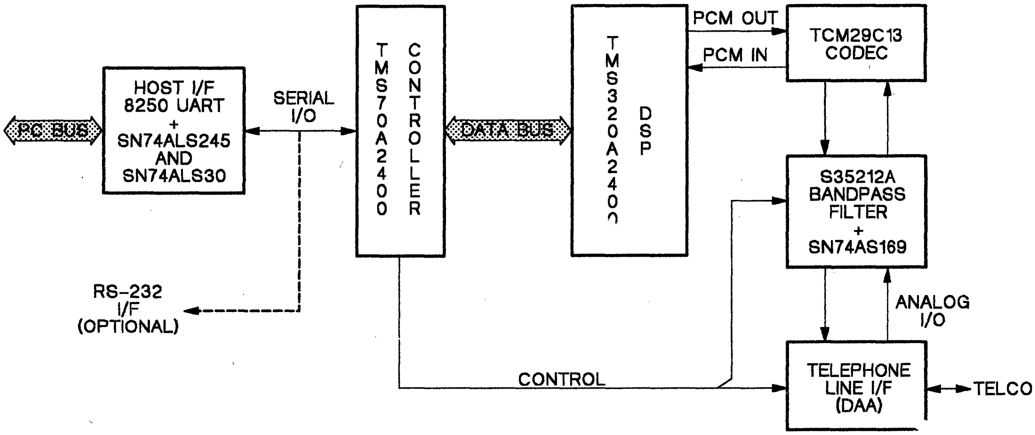


Figure 6-10. 2400 bps Modem

6.6.2 Speech Synthesis System

The system design for speech applications consists of a codec, a digital signal processor supported with program and data memory, a speech data memory, and an optional host processor. A block diagram of this system, shown in Figure 6-11, consists of the following components:

- Codec (TCM29C18)
- Digital signal processor (TMS320C17)
- Speech data ROM (TSP60C20) or EPROM (TMS27C56)
- Microcomputer host (TMS70C42).

The actual speech system is composed of the digital signal processor and the codec. The microcomputer host is used to perform an end-product application that calls upon the speech subsystem when needed, such as in the case of a minicomputer and array processor system. The speech system can be used to perform speech synthesis, vocoding, speech recognition, speaker verification, and DTMF decoding/encoding as well as many other algorithmically intensive applications.

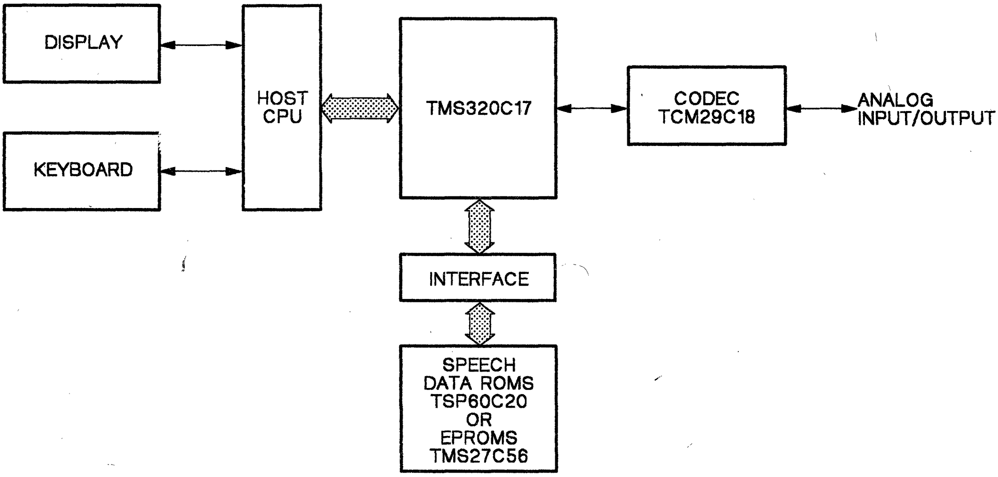


Figure 6-11. Speech Synthesis System

6.6.3 Voice Store-and-Forward Message Center

The voice store-and-forward message center consists of a TMS320C17-based system interfaced to a phone line and a large storage area either on DRAMs or computer disks depending on the application. Some applications of the message center are: voice mail for a computer network, answering machines for home use (see Figure 6-12), and a hand-held battery-operated voice message pad for personal use. Typical algorithms required to perform the task are: half-duplex ADPCM or sub-band coder, LPC synthesis, and DTMF encoder/decoder. A combination of these algorithms will fit into the 4K on-chip program ROM of the TMS320C17, requiring no external data memory. Because the CPU utilization is less than 100 percent when performing any of these tasks, other operations can also be done by the TMS320C17, such as digital volume control, noise filtering, etc. A masked ROM version of the TMS320C17 can provide a cost-effective solution.

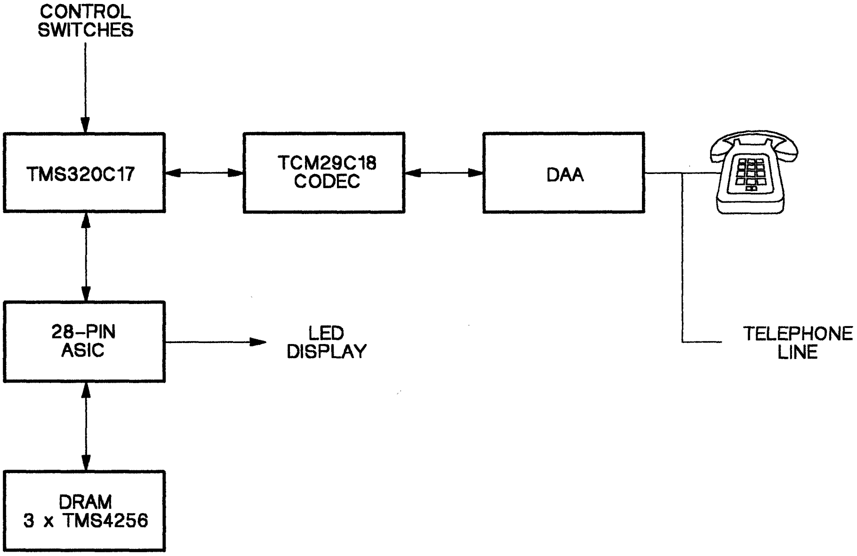


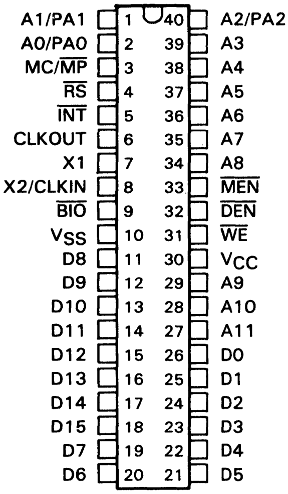
Figure 6-12. Answering Machine

TMS320 FIRST-GENERATION DIGITAL SIGNAL PROCESSORS

JANUARY 1987—REVISED MAY 1989

- 160-ns Instruction Cycle
- 144/256-Word On-Chip Data RAM
- 1.5K/4K-Word On-Chip Program ROM
- 4K-Word On-chip Program EPROM (TMS320E15/E17)
- EPROM Code Protection for Copyright Security
- 4K-Word Total External Memory at Full Speed
- 32-Bit ALU/Accumulator
- 16 × 16-Bit Multiplier with a 32-Bit Product
- 0 to 16-Bit Barrel Shifter
- Eight Input and Eight Output Channels
- Dual-Channel Serial Port (TMS320C17/E17)
- 16-Bit Bidirectional Data Bus with 50-Mbps Transfer Rate
- Single 5-V Supply
- Packaging: 40-Pin DIP, 44-Lead PLCC, and 44-Lead CER-QUAD
- Commercial and Military Versions Available
- NMOS Technology:
  — TMS32010 . . . . 200-ns cycle time
- CMOS Technology:
  — TMS320C10 . . . . 200-ns cycle time
  — TMS320C10-14 . . . . 280-ns cycle time
  — TMS320C10-25 . . . . 160-ns cycle time
  — TMS320C15 . . . . 200-ns cycle time
  — TMS320C15-25 . . . . 160-ns cycle time
  — TMS320E15 (EPROM) . . . . 200-ns cycle time
  — TMS320E15-25 (EPROM) . . 160-ns cycle time
  — TMS320C17 . . . . 200-ns cycle time
  — TMS320E17 (EPROM) . . . . 200-ns cycle time

TMS32010, TMS320C10
N PACKAGE
(TOP VIEW)



A1/PA1 1 40 A2/PA2
A0/PA0 2 39 A3
MC/$\overline{\text{MP}}$ 3 38 A4
$\overline{\text{RS}}$ 4 37 A5
$\overline{\text{INT}}$ 5 36 A6
CLKOUT 6 35 A7
X1 7 34 A8
X2/CLKIN 8 33 $\overline{\text{MEN}}$
$\overline{\text{BIO}}$ 9 32 $\overline{\text{DEN}}$
V_{SS} 10 31 $\overline{\text{WE}}$
D8 11 30 V_{CC}
D9 12 29 A9
D10 13 28 A10
D11 14 27 A11
D12 15 26 D0
D13 16 25 D1
D14 17 24 D2
D15 18 23 D3
D7 19 22 D4
D6 20 21 D5

This data sheet provides complete design documentation for all the first-generation devices of the TMS320 family. This facilitates the selection of the devices best suited for user applications by providing all specifications and special features for each TMS320 member. This data sheet is divided into four major sections: architecture, electrical specifications (NMOS and CMOS), timing diagrams, and mechanical data. In each of these sections, generic information is presented first, followed by specific device information. An index is provided for quick reference to specific information about a device.

TEXAS INSTRUMENTS
POST OFFICE BOX 1443 • HOUSTON, TEXAS 77001

description

The TMS320 family of 16/32-bit single-chip digital signal processors combines the flexibility of a high-speed controller with the numerical capability of an array processor, thereby offering an inexpensive alternative to multichip bit-slice processors. The highly paralleled architecture and efficient instruction set provide speed and flexibility to produce a MOS microprocessor family capable of executing 6.4 MIPS (million instructions per second). The TMS320 family optimizes speed by implementing functions in hardware that other processors implement through microcode or software. This hardware-intensive approach provides the design engineer with processing power previously unavailable on a single chip.

The TMS320 family consists of two generations of digital signal processors. The first generation contains the TMS32010 and its spinoffs, as described in this data sheet. The TMS32020 and TMS320C25 are the second-generation processors, designed for higher performance. Many features are common among the TMS320 processors. Specific features are added in each processor to provide different cost/performance tradeoffs. Software compatibility is maintained throughout the family to protect the user's investment in architecture. Each processor has software and hardware tools to facilitate rapid design.

introduction

The TMS32010, the first NMOS digital signal processor in the TMS320 family, was introduced in 1983. Its powerful instruction set, inherent flexibility, high-speed number-crunching capabilities, and innovative architecture have made this high-performance, cost-effective processor the ideal solution to many telecommunications, computer, commercial, industrial, and military applications. Since that time, the TMS320C10, a low-power CMOS version of the industry-standard TMS32010, and other spinoff devices have been added to the first generation of the TMS320 family.

The TMS32010 microprocessor executes at 20 MHz or 5 MIPS. It is capable of executing a 16 x 16-bit multiply with a 32-bit result in a single instruction cycle. On-chip data RAM of 144 words and on-chip program ROM of 1.5K words are available. Full-speed execution of 4K words of off-chip program memory is also possible.

The TMS320C10 is object-code and pin-for-pin compatible with the TMS32010. It is processed in CMOS technology, achieving a power dissipation less than one-sixth that of the NMOS device. The lower power dissipation makes the TMS320C10 ideal for power-sensitive applications such as digital telephony and portable products. The TMS320C10-25, a 25-MHz version of the TMS320C10, has a 160-ns instruction cycle time and is well suited for high-performance DSP applications. The TMS320C10 is also available in a 280-ns version, the TMS320C10-14. This device provides a low-cost alternative for DSP applications not requiring the maximum operating frequency of the TMS320C10.

The TMS320C15 and TMS320E15 CMOS devices are object-code and pin-for-pin compatible with the TMS32010 and offer expanded on-chip RAM of 256 words and on-chip program ROM or EPROM of 4K words. These devices allow the capability of upgrading performance and reducing power, board space, and system cost without hardware redesign. The TMS320C15/E15 are available in 160-ns versions, the TMS320C15-25 and TMS320E15-25.

introduction (continued)

The TMS320C17 and TMS320E17 also offer expanded on-chip RAM of 256 words and on-chip program ROM or EPROM of 4K words. These devices provide a dual-channel serial interface, on-chip μ-law/A-law companding hardware, and a serial port timer. In addition, a 16-bit coprocessor interface provides a direct communication channel to common 4/8-bit microcomputers (no glue logic required), and minimal logic interface to most common 16/32-bit microprocessors. The devices are object-code compatible with the TMS32010 and processed in CMOS technology.

Table 1 provides an overview of the first generation of TMS320 processors with comparisons of memory, I/O, cycle timing, power, package type, technology, and military support. For specific availability, contact the nearest TI Field Sales Office.

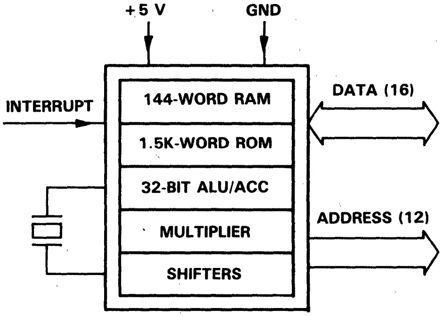
TABLE 1. TMS320 FIRST-GENERATION DEVICE OVERVIEW

DEVICE	MEMORY				I/O†			CYCLE	TYP	PACKAGE		
	ON-CHIP			OFF-CHIP				TIME	POWER	TYPE‡		
	RAM	ROM	EPROM	EXPANSION	SER	PAR	CPX	(ns)	(mW)	DIP	PLCC	CER-QUAD
TMS32010§ (NMOS)	144	1.5K	–	4K	–	8 x 16	–	200	900	40	–	–
TMS320C10§ (CMOS)	144	1.5K	–	4K	–	8 x 16	–	200	165	40	44	–
TMS320C10-14 (CMOS)	144	1.5K	–	4K	–	8 x 16	–	280	140	40	44	–
TMS320C10-25 (CMOS)	144	1.5K	–	4K	–	8 x 16	–	160	200	40	44	–
TMS320C14¶ (CMOS)	256	4K	–	4K	1	7 x 16	–	160	–	–	68	–
TMS320E14¶ (CMOS)	256	–	4K	4K	1	7 x 16	–	160	–	–	–	68
TMS320C15¶ (CMOS)	256	4K	–	4K	–	8 x 16	–	200	225	40	44	–
TMS320C15-25 (CMOS)	256	4K	–	4K	–	8 x 16	–	160	250	40	44	–
TMS320E15¶ (CMOS)	256	–	4K	4K	–	8 x 16	–	200	275	40	–	44
TMS320E15-25 (CMOS)	256	–	4K	4K	–	8 x 16	–	160	325	40	–	44
TMS320C17 (CMOS)	256	4K	–	–	2	6 x 16	YES	200	250	40	44	–
TMS320E17 (CMOS)	256	–	4K	–	2	6 x 16	YES	200	275	40	–	44

† SER = serial; PAR = parallel; CPX = coprocessor interface.
‡ DIP = dual in-line pin; PLCC = plastic-leaded chip carrier;
CER-QUAD = surface mount ceramic-leaded chip carrier.
§ Military version available.
¶ Military version planned; contact nearest TI Field Sales Office for availability.

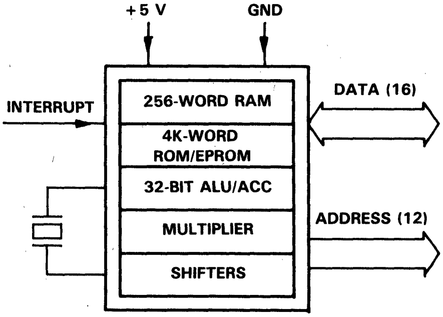
Key Features: TMS32010/C10

- Instruction Cycle Timing:
  - 160 ns (TMS320C10-25)
  - 200 ns (TMS32010/C10)
  - 280 ns (TMS320C10-14)
- 144 Words of On-Chip Data RAM
- 1.5K Words of On-Chip Program ROM
- External Memory Expansion up to 4K Words at Full Speed
- 16 x 16-Bit Multiplier with 32-Bit Product
- 0 to 16-Bit Barrel Shifter
- On-Chip Clock Oscillator
- Single 5-V Supply
- Device Packaging:
  - 40-Pin DIP (all devices)
  - 44-Lead PLCC (CMOS only)
- Technology
  - NMOS: TMS32010
  - CMOS: TMS320C10/C10-14/C10-25

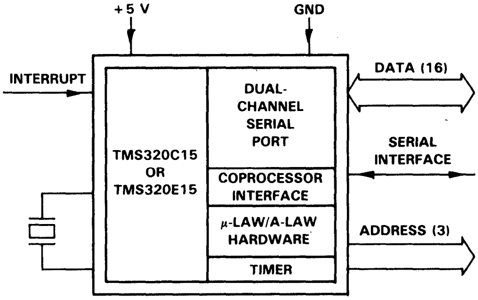


Key Features: TMS320C15/E15

- Instruction Cycle Timing:
  - 160 ns (TMS320C15-25/E15-25)
  - 200 ns (TMS320C15/E15)
- 256 Words of On-Chip Data RAM
- 4K Words of On-Chip Program ROM (TMS320C15/C15-25)
- 4K Words of On-Chip Program EPROM (TMS320E15/E15-25)
- EPROM Code Protection for Copyright Security
- External Memory up to 4K Words at Full Speed
- Object-Code and Pin-For-Pin Compatible with TMS32010
- 16 x 16-Bit Multiplier with 32-Bit Product
- 0 to 16-Bit Barrel Shifter
- On-Chip Clock Oscillator
- Single 5-V Supply
- Device Packaging:
  - 40-Pin DIP (all devices)
  - 44-Lead PLCC (TMS320C15/C15-25)
  - 44-Lead CER-QUAD (TMS320E15/E15-25)
- CMOS Technology



Key Features: TMS320C17/E17



- **Instruction Cycle Timing:**
  – 200 ns (TMS320C17/E17)
- **256 Words of On-Chip Data RAM**
- **4K Words of On-Chip Program ROM (TMS320C17)**
- **4K Words of On-Chip Program EPROM (TMS320E17)**
- **EPROM Code Protection for Copyright Security**
- **Object-Code Compatible with TMS32010**
- **Dual-Channel Serial Port for Full-Duplex Serial Communication**
- **Serial Port Timer for Standalone Serial Communications**
- **On-Chip Companding Hardware for μ-law/A-law PCM Conversions**
- **16-Bit Coprocessor Interface for Common 4/8/16/32-Bit Microcomputers/Microprocessors**
- **Device Packaging:**
  – 40-Pin DIP (all devices)
  – 44-Lead PLCC (TMS320C17)
  – 44-Lead CER-QUAD (TMS320E17)
- **CMOS Technology**

architecture

The TMS320 family utilizes a modified Harvard architecture for speed and flexibility. In a strict Harvard architecture, program and data memory lie in two separate spaces, permitting a full overlap of instruction fetch and execution. The TMS320 family's modification of the Harvard architecture allows transfers between program and data spaces, thereby increasing the flexibility of the device. This modification permits coefficients stored in program memory to be read into the RAM, eliminating the need for a separate coefficient ROM. It also makes available immediate instructions and subroutines based on computed values.

32-bit ALU/accumulator

The TMS320 first-generation devices contain a 32-bit ALU and accumulator for support of double-precision, two's-complement arithmetic. The ALU is a general-purpose arithmetic unit that operates on 16-bit words taken from the data RAM or derived from immediate instructions. In addition to the usual arithmetic instructions, the ALU can perform Boolean operations, providing the bit manipulation ability required of a high-speed controller. The accumulator stores the output from the ALU and is often an input to the ALU. It operates with a 32-bit wordlength. The accumulator is divided into a high-order word (bits 31 through 16) and a low-order word (bits 15 through 0). Instructions are provided for storing the high- and low-order accumulator words in memory.

shifters

Two shifters are available for manipulating data. The ALU barrel shifter performs a left-shift of 0 to 16 places on data memory words loaded into the ALU. This shifter extends the high-order bit of the data word and zero-fills the low-order bits for two's-complement arithmetic. The accumulator parallel shifter performs a left-shift of 0, 1, or 4 places on the entire accumulator and places the resulting high-order accumulator bits into data RAM. Both shifters are useful for scaling and bit extraction.

16 x 16-bit parallel multiplier

The multiplier performs a 16 x 16-bit two's-complement multiplication with a 32-bit result in a single instruction cycle. The multiplier consists of three units: the T Register, P Register, and multiplier array. The 16-bit T Register temporarily stores the multiplicand; the P Register stores the 32-bit product. Multiplier values either come from the data memory or are derived immediately from the MPYK (multiply immediate) instruction word. The fast on-chip multiplier allows the device to perform fundamental operations such as convolution, correlation, and filtering.

data and program memory

Since the TMS320 devices use a Harvard architecture, data and program memory reside in two separate spaces. The first-generation devices have 144 or 256 words of on-chip data RAM and 1.5K or 4K words of on-chip program ROM. On-chip program EPROM of 4K words is provided on the TMS320E15/E17. The EPROM cell utilizes standard PROM programmers and is programmed identically to a 64K CMOS EPROM (TMS27C64).

program memory expansion

The first-generation devices are capable of executing up to 4K words of external memory at full speed for those applications requiring external program memory space. This allows for external RAM-based systems to provide multiple functionality. The TMS320C17/E17 provides no memory expansion capability.

microcomputer/microprocessor operating modes (TMS32010/C10/C15/E15)

The TMS32010/C10 and TMS320C15/E15 devices offer two modes of operation defined by the state of the MC/$\overline{\text{MP}}$ pin: the microcomputer mode (MC/$\overline{\text{MP}}$ = 1) or the microprocessor mode (MC/$\overline{\text{MP}}$ = 0). In the microcomputer mode, on-chip ROM is mapped into the memory space with up to 4K words of external memory available. In the microprocessor mode, all 4K words of memory are external.

interrupts and subroutines

The TMS320 first-generation devices contain a four-level hardware stack for saving the contents of the program counter during interrupts and subroutine calls. Instructions are available for saving the device's complete context. PUSH and POP instructions permit a level of nesting restricted only by the amount of available RAM. The interrupts used in these devices are maskable.

input/output

The 16-bit parallel data bus can be utilized to perform I/O functions in two cycles. The I/O ports are addressed by the three LSBs on the address lines. In addition, a polling input for bit test and jump operations ($\overline{\text{BIO}}$) and an interrupt pin ($\overline{\text{INT}}$) have been incorporated for multitasking.

serial port (TMS320C17/E17)

Two of the I/O ports on the TMS320C17/E17 are dedicated to the serial port and companding hardware. I/O port 0 is dedicated to control register 0, which controls the serial port, interrupts, and companding hardware. I/O port 1 accesses control register 1, as well as both serial port channels, and the companding hardware. The six remaining I/O ports are available for external parallel interfaces.

The dual-channel serial port is capable of full-duplex serial communication and offers direct interface to combo-codecs. Receive and transmit registers that operate with 8-bit data samples are I/O-mapped. Either internal or external framing signals for serial data transfers are selected through the system control register. The serial port clock provides the bit timing for transfers with the serial port, and may be either an input or output. A framing pulse signal provides framing pulses for combo-codec circuits, an 8-kHz sample clock for voice-band systems, or a timer for control applications.

companding hardware (TMS320C17/E17)

On-chip hardware enables the TMS320C17/E17 to compand (COMpress/exPAND) data in either μ-law or A-law format. The companding logic operation is configured via the system control register. Data may be companded in either a serial mode for operation on serial port data (converting between linear and logarithmic PCM) or a parallel mode for computation inside the device. The TMS320C17/E17 allows the hardware companding logic to operate with either sign-magnitude or two's-complement numbers.

coprocessor port (TMS320C17/E17)

The coprocessor port on the TMS320C17/E17 provides a direct connection to most 4/8-bit microcomputers and 16/32-bit microprocessors. The port is accessed through I/O port 5 using IN and OUT instructions. The coprocessor interface allows the device to act as a peripheral (slave) microcomputer to a microprocessor, or as a master to a peripheral microcomputer. In the microcomputer mode, the 16 data lines are used for the 6 parallel 16-bit I/O ports. In the coprocessor mode, the 16-bit parallel port is reconfigured to operate as a 16-bit latched bus interface. For peripheral transfer, an 8-bit or 16-bit length of the coprocessor port can be selected.

instruction set

A comprehensive instruction set supports both numeric-intensive operations, such as signal processing, and general-purpose operations, such as high-speed control. All of the first-generation devices are object-code compatible and use the same 60 instructions. The instruction set consists primarily of single-cycle single-word instructions, permitting execution rates of more than six million instructions per second. Only infrequently used branch and I/O instructions are multicycle. Instructions that shift data as part of an arithmetic operation execute in a single cycle and are useful for scaling data in parallel with other operations.

Three main addressing modes are available with the instruction set: direct, indirect, and immediate addressing.

direct addressing

In direct addressing, seven bits of the instruction word concatenated with the 1-bit data page pointer form the data memory address. This implements a paging scheme in which the first page contains 128 words, and the second page contains up to 128 words.

indirect addressing

Indirect addressing forms the data memory address from the least-significant eight bits of one of the two auxiliary registers, AR0 and AR1. The Auxiliary Register Pointer (ARP) selects the current auxiliary register. The auxiliary registers can be automatically incremented or decremented and the ARP changed in parallel with the execution of any indirect instruction to permit single-cycle manipulation of data tables. Indirect addressing can be used with all instructions requiring data operands, except for the immediate operand instructions.

immediate addressing

Immediate instructions derive data from part of the instruction word rather than from the data RAM. Some useful immediate instructions are multiply immediate (MPYK), load accumulator immediate (LACK), and load auxiliary register immediate (LARK).

instruction set summary

Table 2 lists the symbols and abbreviations used in Table 3, the instruction set summary. Table 3 contains a short description and the opcode for each TMS320 first-generation instruction. The summary is arranged according to function and alphabetized within each functional group.

TABLE 2. INSTRUCTION SYMBOLS

SYMBOL	MEANING
ACC	Accumulator
D	Data memory address field
I	Addressing mode bit
K	Immediate operand field
PA	3-bit port address field
R	1-bit operand field specifying auxiliary register
S	4-bit left-shift code
X	3-bit accumulator left-shift field

TABLE 3. TMS320 FIRST-GENERATION INSTRUCTION SET SUMMARY

ACCUMULATOR INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABS	Absolute value of accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0
ADD	Add to accumulator with shift	1	1	0	0	0	0	← S →				I	← D →						
ADDH	Add to high-order accumulator bits	1	1	0	1	1	0	0	0	0	0	I	← D →						
ADDS	Add to accumulator with no sign extension	1	1	0	1	1	0	0	0	0	1	I	← D →						
AND	AND with accumulator	1	1	0	1	1	1	1	0	0	1	I	← D →						
LAC	Load accumulator with shift	1	1	0	0	1	0	← S →				I	← D →						
LACK	Load accumulator immediate	1	1	0	1	1	1	1	1	1	0	← K →							
OR	OR with accumulator	1	1	0	1	1	1	1	0	1	0	I	← D →						
SACH	Store high-order accumulator bits with shift	1	1	0	1	0	1	1	← X →			I	← D →						
SACL	Store low-order accumulator bits	1	1	0	1	0	1	0	0	0	0	I	← D →						
SUB	Subtract from accumulator with shift	1	1	0	0	0	1	← S →				I	← D →						
SUBC	Conditional subtract (for divide)	1	1	0	1	1	0	0	1	0	0	I	← D →						
SUBH	Subtract from high-order accumulator bits	1	1	0	1	1	0	0	0	1	0	I	← D →						
SUBS	Subtract from accumulator with no sign extension	1	1	0	1	1	0	0	0	1	1	I	← D →						
XOR	Exclusive OR with accumulator	1	1	0	1	1	1	1	0	0	0	I	← D →						
ZAC	Zero accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1
ZALH	Zero accumulator and load high-order bits	1	1	0	1	1	0	0	1	0	1	I	← D →						
ZALS	Zero accumulator and load low-order bits with no sign extension	1	1	0	1	1	0	0	1	1	0	I	← D →						

AUXILIARY REGISTER AND DATA PAGE POINTER INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LAR	Load auxiliary register	1	1	0	0	1	1	1	0	0	R	I	← D →						
LARK	Load auxiliary register immediate	1	1	0	1	1	1	0	0	0	R	← K →							
LARP	Load auxiliary register pointer immediate	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	K
LDP	Load data memory page pointer	1	1	0	1	1	0	1	1	1	1	I	← D →						
LDPK	Load data memory page pointer immediate	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	K
MAR	Modify auxiliary register and pointer	1	1	0	1	1	0	1	0	0	0	I	← D →						
SAR	Store auxiliary register	1	1	0	0	1	1	0	0	0	R	I	← D →						

TABLE 3. TMS320 FIRST-GENERATION INSTRUCTION SET SUMMARY (CONTINUED)

BRANCH INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B	Branch unconditionally	2	2	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BANZ	Branch on auxiliary register not zero	2	2	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BGEZ	Branch if accumulator ≥ 0	2	2	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BGZ	Branch if accumulator > 0	2	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BIOZ	Branch on $\overline{\text{BIO}}$ = 0	2	2	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BLEZ	Branch if accumulator ≤ 0	2	2	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BLZ	Branch if accumulator < 0	2	2	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BNZ	Branch if accumulator ≠ 0	2	2	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BV	Branch on overflow	2	2	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BZ	Branch if accumulator = 0	2	2	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
CALA	Call subroutine from accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0
CALL	Call subroutine immediately	2	2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
RET	Return from subroutine or interrupt routine	2	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	1

T REGISTER, P REGISTER, AND MULTIPLY INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APAC	Add P register to accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1
LT	Load T register	1	1	0	1	1	0	1	0	1	0	I	◄— D —►						
LTA	LTA combines LT and APAC into one instruction	1	1	0	1	1	0	1	1	0	0	I	◄— D —►						
LTD	LTD combines LT, APAC, and DMOV into one instruction	1	1	0	1	1	0	1	0	1	1	I	◄— D —►						
MPY	Multiply with T register, store product in P register	1	1	0	1	1	0	1	1	0	1	I	◄— D —►						
MPYK	Multiply T register with immediate operand; store product in P register	1	1	1	0	0	◄— K —►												
PAC	Load accumulator from P register	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	0
SPAC	Subtract P register from accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0

TABLE 3. TMS320 FIRST-GENERATION INSTRUCTION SET SUMMARY (CONCLUDED)

CONTROL INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DINT	Disable interrupt	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1
EINT	Enable interrupt	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0
LST	Load status register	1	1	0	1	1	1	1	0	1	1	I	◀——D——▶						
NOP	No operation	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
POP	POP stack to accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1
PUSH	PUSH stack from accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0
ROVM	Reset overflow mode	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	0
SOVM	Set overflow mode	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1
SST	Store status register	1	1	0	1	1	1	1	1	0	0	I	◀——D——▶						

I/O AND DATA MEMORY OPERATIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMOV	Copy contents of data memory location into next higher location	1	1	0	1	1	0	1	0	0	1	I	◀——D——▶						
IN	Input data from port	2	1	0	1	0	0	0	◀PA▶			I	◀——D——▶						
OUT	Output data to port	2	1	0	1	0	0	1	◀PA▶			I	◀——D——▶						
TBLR	Table read from program memory to data RAM	3	1	0	1	1	0	0	1	1	1	I	◀——D——▶						
TBLW	Table write from data RAM to program memory	3	1	0	1	1	1	1	1	0	1	I	◀——D——▶						

aevelopment support products

Together, Texas Instruments and its authorized third-party suppliers offer an extensive line of development support products to assist the user in all aspects of TMS320 first-generation-based design and development. These products range from development and application software to complete hardware development and evaluation systems. Table 4 lists the development support products for the first-generation TMS320 devices.

System development may begin with the use of the simulator, evaluation module (EVM), or emulator (XDS), along with an assembler/linker. These tools give the TMS320 user various means of evaluation, from software simulation of the first-generation TMS320s (simulator) to full-speed in-circuit emulation with hardware and software break point tracing and timing capabilities (XDS).

Software and hardware can be developed simultaneously by using the macro assembler/linker or simulator for software development, the XDS for hardware development, and the evaluation module for both software development and limited hardware development.

Many third-party vendors offer additional development support for the first-generation TMS320s, including assembler/linkers, simulators, high-level languages, applications software, algorithm development tools, application boards, software development boards, and in-circuit emulators. Refer to the *TMS320 Family Development Support Reference Guide* (SPRU011A) for further information about TMS320 development support products offered by both Texas Instruments and its third-party suppliers.

Additional support for the TMS320 products consists of an extensive library of product and applications documentation. Three-day DSP design workshops are offered by the TI Regional Technology Centers (RTCs). These workshops provide insight into the architecture and the instruction set of the first-generation TMS320s as well as hands-on training with the TMS320 development tools. When technical questions arise in regard to a TMS320 member, contact Texas Instruments TMS320 Hotline at (713) 274-2320. Or, keep informed on the latest TI and third-party development support tools by accessing the libraries of application source code via the DSP Bulletin Board Service (BBS) at (713) 274-2323. The BBS provides access for the 2400-/1200-/300-bps modems.

TABLE 4. TMS320 FIRST-GENERATION SOFTWARE AND HARDWARE SUPPORT

SOFTWARE TOOLS	PART NUMBER
Macro Assembler/Linker	
PC/MS-DOS	TMDS3242850-02
VAX/VMS	TMDS3242250-08
VAX ULTRIX	TMDS3242260-08
SUN-3 UNIX	TMDS3242550-08
Simulator	
PC/MS-DOS	TMDS3240811-02
VAX/VMS	TMDS3240211-08
Digital Filter Design Package (DFDP)	
IBM PC PC-DOS	DFDP/IBM002
DSP Software Library	
PC/MS-DOS	TMDC3240812-12
VAX/VMS	TMDC3240212-18
TMS320 Bell 212A Modem Software	
PC/MS-DOS	TMDX3240813-12
Data Encryption Standard Software	
PC/MS-DOS	TMDX3240814-12
HARDWARE TOOLS	**PART NUMBER**
Evaluation Tools	
Evaluation Module (EVM)	RTC/EVM320A-03
Analog Interface Board 1 (AIB1)	RTC/EVM320C-06
Analog Interface Board 2 (AIB2)	RTC/AIB320A-06
EPROM DSP Starter Kit (TMS320E15)	RTC/EVM320E-15
XDS/22 Emulators	
TMS320C10/C15	TMDS3262211
TMS320C14	TMDX3262214
TMS320C17	TMDX3262217
XDS/22 Upgrade Kits	
TMS32010 → TMS320C10/C15	TMDS3282215
TMS320C10/C15 → TMS320C14	TMDX3285010 and TMDX3285018
TMS320C10/C15 → TMS320C17	TMDX3285014 and TMDX3285018
EPROM Programming Adaptor Sockets	
40- to 28-pin (TMS320E15/E17)	RTC/PGM320A-06
44- to 28-pin (TMS320E15/E17)	RTC/PGM320C-06
68- to 28-pin (TMS320E14)	TMDX3270110
Additional Target Connector	
44-lead PLCC (TMS320C10)	TMDX3288810

documentation support

Extensive documentation supports the first-generation TMS320 devices from product announcement through applications development. The types of documentation include data sheets with design specifications, complete user's guides, and 750 pages of application reports published in *Digital Signal Processing Applications with the TMS320 Family* (SPRA012A).

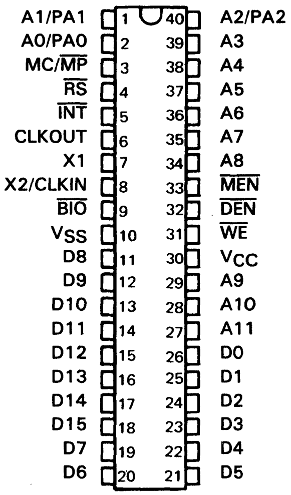
A series of DSP textbooks is being published to support digital signal processing research and education. The first book, *DFT/FFT and Convolution Algorithms,* is now available. The TMS320 newsletter, *Details on Signal Processing,* is published quarterly and distributed to update TMS320 customers on product information. The TMS320 DSP bulletin board service provides access to large amounts of information pertaining to the TMS320 family.

Refer to the *TMS320 Family Development Support Reference Guide* (SPRU011A) for further information about TMS320 documentation. To receive copies of first-generation TMS320 literature, call the Customer Response Center at 1-800-232-3200.

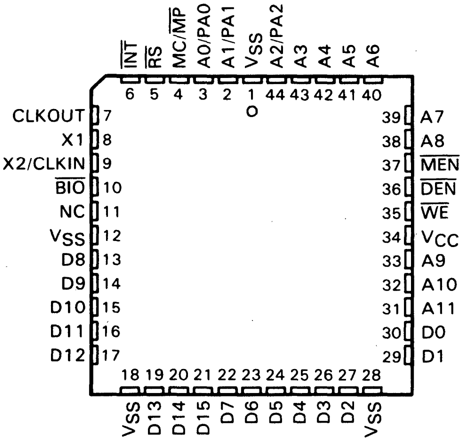
description

Since the TMS32010 was the first digital signal processor in the TMS320 family, its architecture has served as the basis from which first-generation spinoff devices have evolved. The TMS320C10 is a low-power CMOS version of the TMS32010 and identical to it. The TMS320C15/E15 is object-code and pin-for-pin compatible with the TMS32010 and offers expanded on-chip RAM and ROM or EPROM.

TMS32010, TMS320C10
TMS320C15, TMS320E15
N/JD PACKAGE
(TOP VIEW)



TMS320C10, TMS320C15, TMS320E15
FN AND FZ PACKAGES
(TOP VIEW)

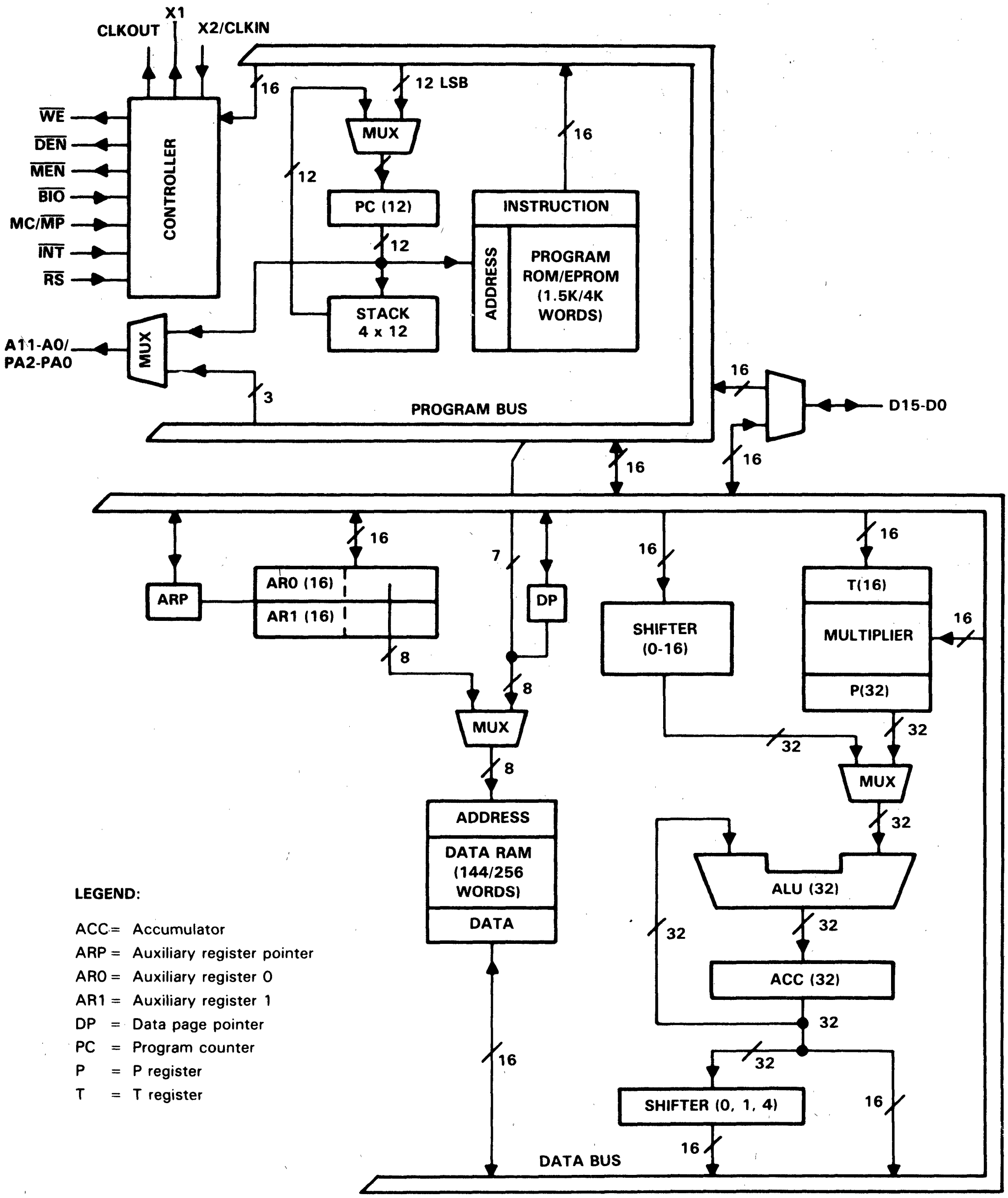


PIN NOMENCLATURE (TMS32010, TMS320C10, TMS320C15, TMS320E15†)

NAME	I/O‡	DEFINITION
A11-A0/PA2-PA0	O	External address bus. I/O port address multiplexed over PA2-PA0.
$\overline{\text{BIO}}$	I	External polling input
CLKOUT	O	System clock output, ¼ crystal/CLKIN frequency
D15-D0	I/O	16-bit parallel data bus
$\overline{\text{DEN}}$	O	Data enable for device input data on D15-D0
$\overline{\text{INT}}$	I	External interrupt input
MC/$\overline{\text{MP}}$	I	Memory mode select pin. High selects microcomputer mode. Low selects microprocessor mode.
$\overline{\text{MEN}}$	O	Memory enable indicates that D15-D0 will accept external memory instruction.
NC	O	No connection
$\overline{\text{RS}}$	I	Reset for initializing the device
V_{CC}	I	+5 V supply
V_{SS}	I	Ground
$\overline{\text{WE}}$	O	Write enable for device output data on D15-D0
X1	O	Crystal output for internal oscillator
X2/CLKIN	I	Crystal input for internal oscillator or external system clock input

†See EPROM programming section.
‡Input/Output/High-impedance state.

functional block diagram (TMS32010, TMS320C10, TMS320C15, TMS320E15)



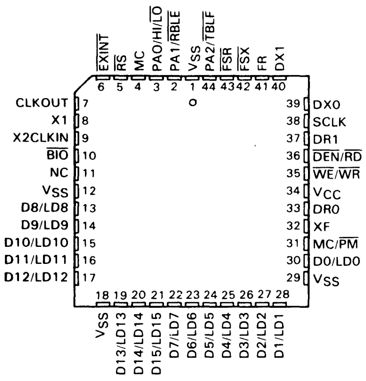
LEGEND:

ACC	=	Accumulator
ARP	=	Auxiliary register pointer
AR0	=	Auxiliary register 0
AR1	=	Auxiliary register 1
DP	=	Data page pointer
PC	=	Program counter
P	=	P register
T	=	T register

description

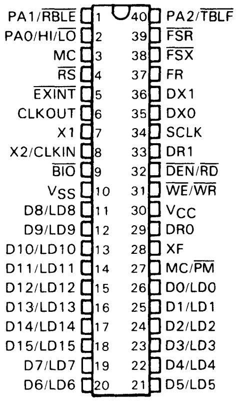
The TMS320C17, like the TMS320C15, has 256 words of on-chip data RAM and 4K words of on-chip program ROM. The TMS320C17 is object-code compatible with the TMS32010. The TMS320C17 provides a dual-channel serial port and is designed specifically to interface to two combo-codecs. A 16-bit coprocessor interface is also provided for interfacing to common 4/8/16/32-bit microcomputers/microprocessors.

TMS320C17, TMS320E17
FN AND FZ PACKAGES
(TOP VIEW)



$\overline{\text{EXINT}}$ 6
$\overline{\text{RS}}$ 5
MC 4
PA0/HI/$\overline{\text{LO}}$ 3
PA1/$\overline{\text{RBLE}}$ 2
V_{SS} 1
PA2/$\overline{\text{TBLF}}$ 44
$\overline{\text{FSR}}$ 43
$\overline{\text{FSX}}$ 42
FR 41
DX1 40
CLKOUT 7
X1 8
X2CLKIN 9
$\overline{\text{BIO}}$ 10
NC 11
V_{SS} 12
D8/LD8 13
D9/LD9 14
D10/LD10 15
D11/LD11 16
D12/LD12 17
39 DX0
38 SCLK
37 DR1
36 $\overline{\text{DEN}}$/$\overline{\text{RD}}$
35 $\overline{\text{WE}}$/$\overline{\text{WR}}$
34 V_{CC}
33 DR0
32 XF
31 MC/$\overline{\text{PM}}$
30 D0/LD0
29 V_{SS}
18 V_{SS}
19 D13/LD13
20 D14/LD14
21 D15/LD15
22 D7/LD7
23 D6/LD6
24 D5/LD5
25 D4/LD4
26 D3/LD3
27 D2/LD2
28 D1/LD1

TMS320C17, TMS320E17
N/JD PACKAGE
(TOP VIEW)



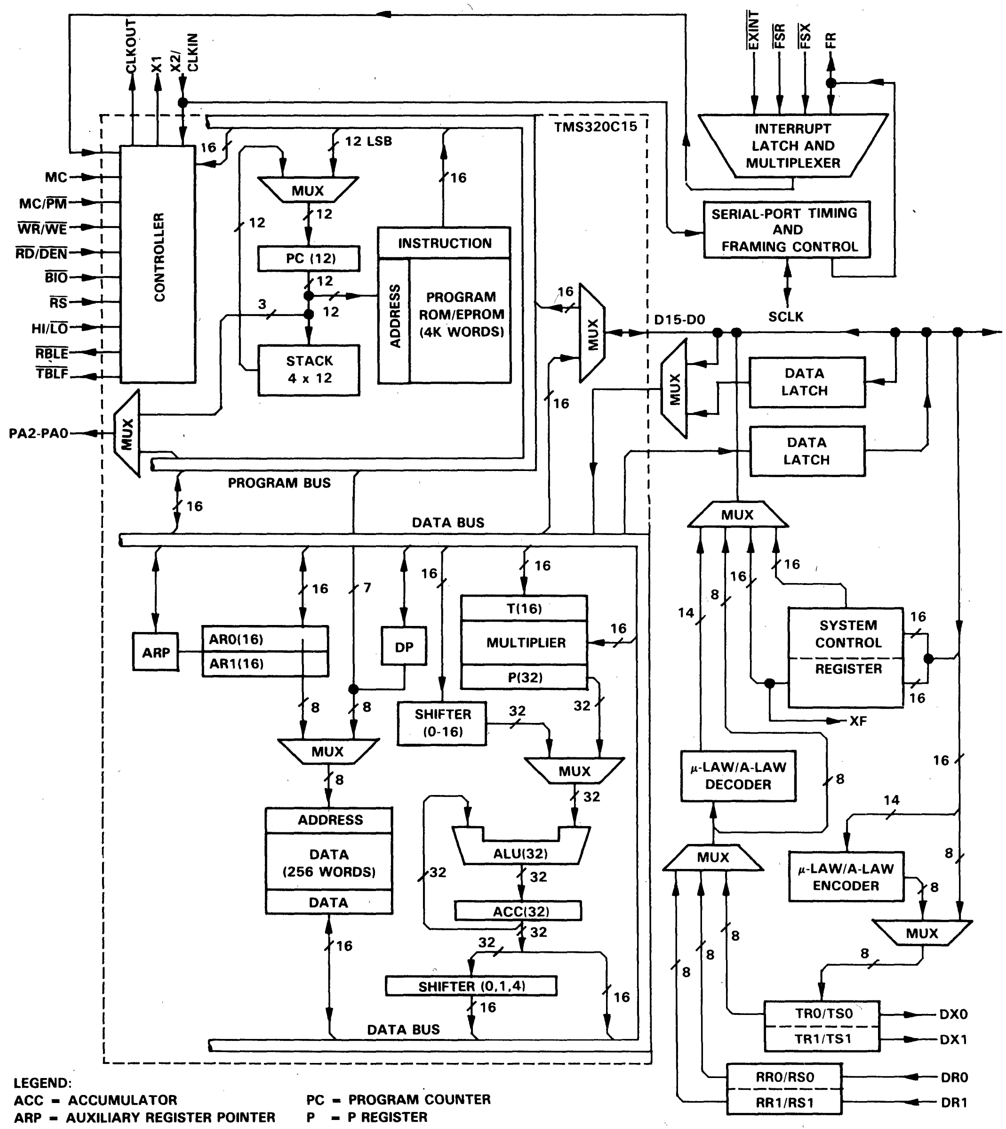
PA1/$\overline{\text{RBLE}}$ 1
PA0/HI/$\overline{\text{LO}}$ 2
MC 3
$\overline{\text{RS}}$ 4
$\overline{\text{EXINT}}$ 5
CLKOUT 6
X1 7
X2/CLKIN 8
$\overline{\text{BIO}}$ 9
V_{SS} 10
D8/LD8 11
D9/LD9 12
D10/LD10 13
D11/LD11 14
D12/LD12 15
D13/LD13 16
D14/LD14 17
D15/LD15 18
D7/LD7 19
D6/LD6 20
40 PA2/$\overline{\text{TBLF}}$
39 $\overline{\text{FSR}}$
38 $\overline{\text{FSX}}$
37 FR
36 DX1
35 DX0
34 SCLK
33 DR1
32 $\overline{\text{DEN}}$/$\overline{\text{RD}}$
31 $\overline{\text{WE}}$/$\overline{\text{WR}}$
30 V_{CC}
29 DR0
28 XF
27 MC/$\overline{\text{PM}}$
26 D0/LD0
25 D1/LD1
24 D2/LD2
23 D3/LD3
22 D4/LD4
21 D5/LD5

PIN NOMENCLATURE (TMS320C17, TMS320E17[†])

NAME	I/O[‡]	DEFINITION
$\overline{\text{BIO}}$	I	External polling input
CLKOUT	O	System clock output, ¼ crystal/CLKIN frequency
D15/LD15-D0/LD0	I/O	16-bit parallel data bus/data lines for coprocessor latch
$\overline{\text{DEN}}$/$\overline{\text{RD}}$	I/O	Data enable for device input data/external read for output latch
DR1, DR0	I	Serial-port receive-channel inputs
DX1, DX0	O	Serial-port transmit-channel outputs
$\overline{\text{EXINT}}$	I	External interrupt input
FR	O	Internal serial-port framing output
$\overline{\text{FSR}}$	I	External serial-port receive framing input
$\overline{\text{FSX}}$	I	External serial-port transmit framing input
MC	I	Microcomputer select (must be same state as MC/$\overline{\text{PM}}$)
MC/$\overline{\text{PM}}$	I	Microcomputer/peripheral coprocessor select (must be same state as MC)
PA0/HI/$\overline{\text{LO}}$	I/O	I/O port address output/latch byte select pin
PA1/$\overline{\text{RBLE}}$	O	I/O port address output/receive buffer latch empty flag
PA2/$\overline{\text{TBLF}}$	O	I/O port address output/transmit buffer latch full flag
$\overline{\text{RS}}$	I	Reset for initializing the device
SCLK	I/O	Serial-port clock
V_{CC}	I	+5 V Supply
V_{SS}	I	Ground
$\overline{\text{WE}}$/$\overline{\text{WR}}$	O	Write enable for device output data/external write for input latch
X1	O	Crystal output for internal oscillator
X2/CLKIN	I	Crystal input for internal oscillator or external oscillator system clock input
XF	O	External-flag output pin

[†] See EPROM programming section.
[‡] Input/Output/High-impedance state.

functional block diagram (TMS320C17, TMS320E17)



LEGEND:

ACC	ACCUMULATOR	PC	PROGRAM COUNTER
ARP	AUXILIARY REGISTER POINTER	P	P REGISTER
AR0	AUXILIARY REGISTER 0	T	T REGISTER
AR1	AUXILIARY REGISTER 1	TR	TRANSMIT REGISTER
DP	DATA PAGE POINTER	RR	RECEIVE REGISTER

architecture

The TMS320C17 consists of five major functional units: the TMS320C15 microcomputer, a system control register, a full-duplex dual-channel serial port, companding hardware, and a coprocessor port.

Three of the I/O ports are used by the serial port, companding hardware, and the coprocessor port. Their operation is determined by the 32 bits of the system control register (see Table 5 for the TMS320C17/E17 control register bit definitions). Control register 0, accessed through port 0, consists of the lower 16 register bits (CR15-CR0 bit), and is used to control the interrupts, serial port connections, and companding hardware operation. Port 1 accesses control register 1, consisting of the upper 16 control bits (CR31-CR16), as well as both serial port channels, the companding hardware, and the coprocessor port channels. Communication with the control register is via IN and OUT instructions to ports 0 and 1.

Interrupts fully support the TMS320C17/E17 serial port interface. Four maskable interrupts ($\overline{\mathrm{EXINT}}$, FR, $\overline{\mathrm{FSX}}$, and $\overline{\mathrm{FSR}}$) are mapped into I/O port 0 via control register 0. When disabled, these interrupts may be used as single-bit logic inputs polled by software.

serial port

The dual-channel serial port is capable of full-duplex serial communication and offers direct interface to two combo-codecs. Two receive and two transmit registers are mapped into I/O port 1, and operate with 8-bit data samples. Internal and external framing signals for serial port transfers (MSB first) are selected via the system control register. The serial port clock, SCLK, provides the bit timing for transfers with the serial port, and may be either an input or output. As an input, an external clock provides the timing for data transfers and framing pulse synchronization. As an output, SCLK provides the timing for standalone serial communication and is derived from the TMS320C17/E17 system clock, X2/CLKIN and system control register bits CR27-CR24 (see Table 6 for the available divide ratios). The internal framing (FR) pulse frequency is derived from the serial port clock (SCLK) and system control register bits CR23-CR16. This framing pulse signal provides framing pulses for combo-codecs, for a sample clock for voice-band systems, or for a timer used in control applications.

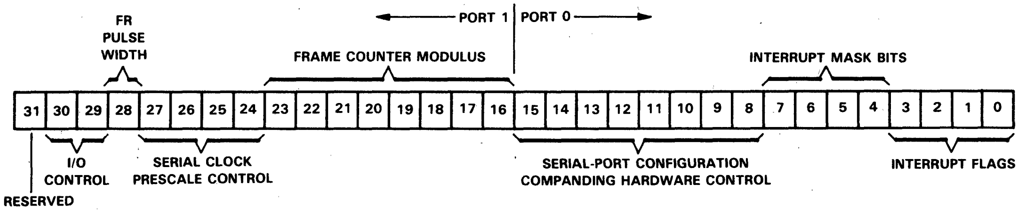
μ-law/A-law companding hardware

The TMS320C17/E17 features hardware companding logic that can operate in either μ-law or A-law format with either sign-magnitude or two's-complement numbers. Data may be companded in either a serial mode for operation on serial port data or a parallel mode for computation inside the device. The companding logic operation is selected through CR14. No bias is required when operating in two's-complement. A bias of 33 is required for sign-magnitude in μ-law companding. Upon reset, the device is programmed to operate in sign-magnitude mode. This mode can be changed by modifying control bit 29 (CR29) in control register 1. For further information on companding, see the *TCM29C13/TCM29C14/TCM29C16/TCM29C17 Combined Single-Chip PCM Codec and Filter Data Sheet*, and the application report, *"Companding Routines for the TMS32010/TMS32020,"* in the book, *Digital Signal Processing Applications with the TMS320 Family* (SPRA012A), both documents published by Texas Instruments.

In the serial mode, sign-magnitude linear PCM (13 magnitude bits plus 1 sign bit for μ-law format or 12 magnitude bits plus 1 sign bit for A-law format) is compressed to 8-bit sign-magnitude logarithmic PCM by the encoder and sent to the transmit register for transmission on an active framing pulse. The decoder converts 8-bit sign-magnitude log PCM from the serial port receive registers to sign-magnitude linear PCM.

In the parallel mode, the serial port registers are disabled to allow parallel data from internal memory to be encoded or decoded for computation inside the device. In the parallel encode mode, the encoder is enabled and a 14-bit sign-magnitude value written to port 1. The encoded value is returned with an IN instruction from port 1. In the parallel decode mode, the decoder is enabled and an 8-bit sign-magnitude log PCM value written to port 1. On the successive IN instruction from port 1, the decoded value is returned. At least one instruction should be inserted between an OUT and the successive IN when companding is performed with two's-complement values.

TABLE 5. CONTROL REGISTER CONFIGURATION



BIT	DESCRIPTION AND CONFIGURATION
0	$\overline{\text{EXINT}}$ interrupt flag†
1	$\overline{\text{FSR}}$ interrupt flag†
2	$\overline{\text{FSX}}$ interrupt flag†
3	FR interrupt flag†
4	$\overline{\text{EXINT}}$ interrupt enable mask. When set to logic 1, an interrupt on $\overline{\text{EXINT}}$ activates device interrupt circuitry.
5	$\overline{\text{FSR}}$ interrupt enable mask. Same as $\overline{\text{EXINT}}$ control.
6	$\overline{\text{FSX}}$ interrupt enable mask. Same as $\overline{\text{EXINT}}$ control.
7	FR interrupt enable mask. Same as $\overline{\text{EXINT}}$ control.
8	Port 1 configuration control: 0 = port 1 connects to either serial-port registers or companding hardware. 1 = port 1 accesses CR31-CR16.
9	External framing enable: 0 = serial-port data transfers controlled by active FR. 1 = serial-port data transfers controlled by active $\overline{\text{FSX}}/\overline{\text{FSR}}$.
10	XF external logic output flag latch
11	Serial-port enable: 0 = parallel companding mode; serial port disabled. 1 = serial companding mode; serial port registers enabled.
12	μ-law/A-law encoder enable: 0 = disabled. 1 = data written to port 1 is μ-law or A-law encoded.
13	μ-law/A-law decoder enable: 0 = disabled. 1 = data read from port 1 is μ-law or A-law decoded.
14	μ-law or A-law encode/decode select: 0 = companding hardware performs μ-law conversion. 1 = companding hardware performs A-law conversion.
15	Serial clock control: 0 = SCLK is an output, derived from the prescaler in timing logic. 1 = SCLK is an input that provides the clock for serial port and frame counter in timing logic.
23-16	Frame counter modulus. Controls FR frequency = SCLK/(CNT + 2) where CNT is binary value of CR23-CR16.‡
27-24	SCLK prescale control bits. (See Table 6 for divide ratios.)
28	FR pulse-width control: 0 = fixed-data rate; FR is 1 SCLK cycle wide. 1 = variable-data rate; FR is 8 SCLK cycles wide.
29	Two's-complement μ-law/A-law conversion enable: 0 = sign-magnitude companding 1 = two's-complement companding
30	8/16-bit length coprocessor mode select: 0 = 8-bit byte length 1 = 16-bit word length
31	Reserved for future expansion: Should be set zero.

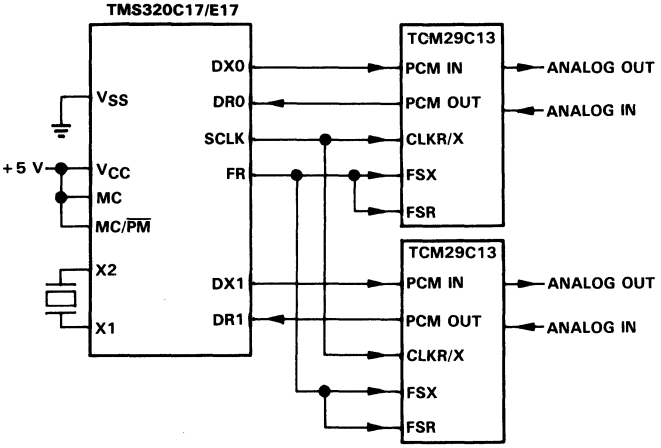
† Interrupt flag is cleared by writing a logic 1 to the bit with an OUT instruction to port 0.

‡ All ones in CR23-CR16 indicate a degenerative state and should be avoided. Bits are operational whether SCLK is an input or an output. CNT must be greater than 7.

TABLE 6. SERIAL CLOCK (SCLK) DIVIDE RATIOS (X2/CLKIN = 20.48 MHZ)

CR27	CR26	CR25	CR24	DIVIDE RATIO	SCLK FREQUENCY	UNIT
0	0	0	0	32	0.640	MHz
0	0	0	1	28	0.731	MHz
0	0	1	0	24	0.853	MHz
0	1	0	0	20	1.024	MHz
1	0	0	0	16	1.280	MHz
1	0	0	1	14	1.463	MHz
1	0	1	0	12	1.706	MHz
1	1	0	0	10	2.048	MHz

The specification for μ-law and A-law log PCM coding is part of the CCITT G.711 recommendation. The following diagram shows a TMS320C17/E17 interface to two codecs as used for μ-law or A-law companding format.



coprocessor port

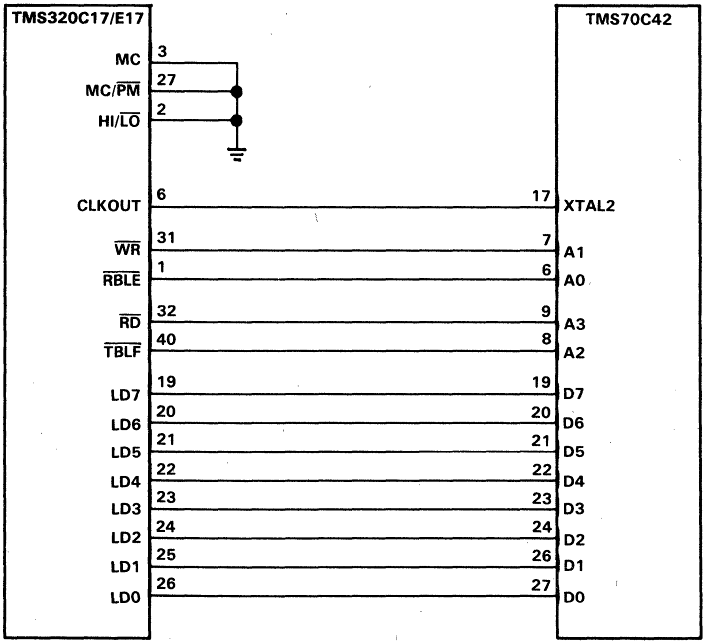
The coprocessor port, accessed through I/O port 5 using IN and OUT instructions, provides a direct connection to most 4/8-bit microcomputers and 16/32-bit microprocessors. The coprocessor interface allows the TMS320C17/E17 to act as a peripheral (slave) microcomputer to a microprocessor, or a master to a peripheral microcomputer such as TMS7042. The coprocessor port is enabled by setting MC/$\overline{\text{PM}}$ and MC low. The microcomputer mode is enabled by setting these two pins high. (Note that MC/$\overline{\text{PM}}$ ≠ MC is undefined.) In the microcomputer mode, the 16 data lines are used for the 6 parallel 16-bit I/O ports.

In the coprocessor mode, the 16-bit coprocessor port is reconfigured to operate as a 16-bit latched bus interface. Control bit 30 (CR30) in control register 1 is used to configure the coprocessor port to either an 8-bit or a 16-bit length. When CR30 is high, the coprocessor port is 16 bits wide, thereby making all 16 bits of the data port available for 16-bit transfers to 16 and 32-bit microprocessors. When CR30 is low, the port is 8 bits wide and mapped to the low byte of the data port for interfacing to 8-bit microcomputers. When operating in the 8-bit mode, both halves of the 16-bit latch can be addressed using the HI/$\overline{\text{LO}}$ pin, thus allowing 16-bit transfers over 8 data lines. When not in the coprocessor mode, port 5 can be used as a generic I/O port.

coprocessor port (continued)

The external processor recognizes the coprocessor interface, in which both processors run asynchronously, as a memory-mapped I/O operation. The external processor lowers the $\overline{WR}$ line and places data on the bus. It next raises the $\overline{WR}$ line to clock the data into the on-chip latch. The rising edge of $\overline{WR}$ automatically creates an interrupt to the TMS320C17/E17, and the falling edge of $\overline{WR}$ clears the $\overline{RBLE}$ (receive buffer latch empty) flag. When the TMS320C17/E17 reads the coprocessor port, it causes the $\overline{RBLE}$ signal to transition to a logic low state clears the data in the latch, and allows the interrupt condition to be cleared internally. Likewise, the external processor reads from the latch by driving the $\overline{RD}$ line active low, thus enabling the output latch to drive the latched data. When the data has been read, the external device will again bring the $\overline{RD}$ line high. This activates the $\overline{BIO}$ line to signal that the transfer is complete and the latch is available for the next transfer. The falling edge of $\overline{RD}$ resets the $\overline{TBLF}$ (transmit buffer latch full) flag. Note that the $\overline{EXINT}$ and $\overline{BIO}$ lines are reserved for coprocessor interface and cannot be driven externally when in the coprocessor mode.

An example of the use of a coprocessor interface is shown below, in which the TMS320C17/E17 is interfaced to the TMS70C42, an 8-bit microcontroller.



NMOS DEVICE ELECTRICAL SPECIFICATIONS

This section contains all the electrical specifications for the TMS320 NMOS first-generation devices. Refer to the top corner for the specific device.

absolute maximum ratings over specified temperature range (unless otherwise noted)†

Supply voltage range, V_{CC}‡	−0.3 V to 7 V
Input voltage range	−0.3 V to 15 V
Output voltage range	−0.3 V to 15 V
Continuous power dissipation	1.5 W
Air temperature range above operating device	0°C to 70°C
Storage temperature range	−55°C to +150°C

† Stresses beyond those listed under "Absolute Maximum Ratings" may cause permananet damage to the device. This is a stress rating only, and functional operation of the device or any other conditions beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

‡ All voltage values are with respect to V_{SS}.

recommended operating conditions

		MIN	NOM	MAX	UNIT
V_{CC} Supply voltage		4.75	5	5.25	V
V_{SS} Supply voltage			0		V
V_{IH} High-level input voltage	All inputs except CLKIN	2			V
	CLKIN	2.8			
V_{IL} Low-level input voltage (all inputs)				0.8	V
I_{OH} High-level output current (all outputs)				−300	µA
I_{OL} Low-level output current (all outputs)				2	mA
T_A Operating free-air temperature		0		70	°C

electrical characteristics over specified temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{OH} High-level output voltage		I_{OH} = MAX		2.4	3		V
V_{OL} Low-level output voltage		I_{OL} = MAX			0.3	0.5	V
I_{OZ} Off-state output current		V_{CC} = MAX	V_O = 2.4 V			20	µA
			V_O = 0.4 V			−20	
I_I Input current		V_I = V_{SS} to V_{CC}	All inputs except CLKIN			±20	µA
			CLKIN			±50	
I_{CC}‡ Supply current		V_{CC} = MAX	T_A = 0°C		180	275	mA
			T_A = 70°C			235§	
C_i Input capacitance	Data bus	f = 1 MHz, All other pins 0 V			25§		pF
	All others				15§		
C_o Output capacitance	Data bus				25§		pF
	All others				10§		

† All typical values except for I_{CC} are at V_{CC} = 5 V, T_A = 25°C.

‡ I_{CC} characteristics are inversely proportional to temperature; i.e., I_{CC} decreases approximately linearly with temperature.

§ Value derived from characterization data and not tested.

PARAMETER MEASUREMENT INFORMATION

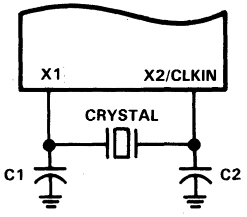


FIGURE 1. INTERNAL CLOCK OPTION

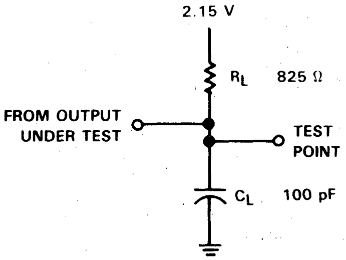


FIGURE 2. TEST LOAD CIRCUIT

input synchronization requirements

For systems using asynchronous inputs to the $\overline{\text{INT}}$ and $\overline{\text{BIO}}$ pins on the TMS32010, the external hardware shown in the diagrams below is recommended to ensure proper execution of interrupts and the BIOZ instruction. This hardware synchronizes the $\overline{\text{INT}}$ and $\overline{\text{BIO}}$ input signals with the rising edge of CLKOUT on the TMS32010. The pulse width required for these input signals is $t_{c(C)}$, which is one TMS32010 clock cycle, plus sufficient setup time for the flip-flop (dependent upon the flip-flop used). Note that these input synchronization requirements apply only to NMOS versions of the TMS32010 and not to other members of the TMS320 family.

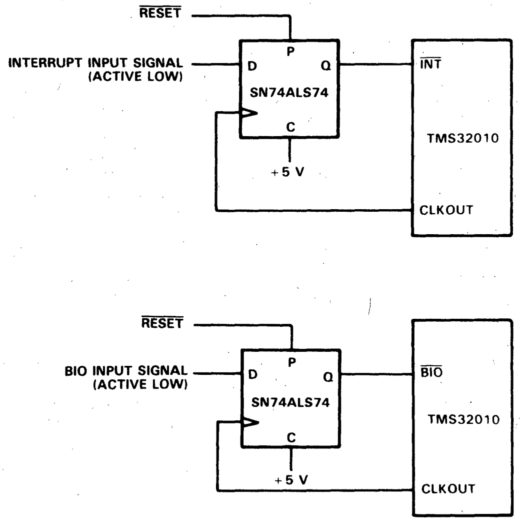


FIGURE 3. ASYNCHRONOUS INPUT SYNCHRONIZATION CIRCUITS

CLOCK CHARACTERISTICS AND TIMING

The TMS32010 can use either its internal oscillator or an external frequency source for a clock.

internal clock option

The internal oscillator is enabled by connecting a crystal across X1 and X2/CLKIN (see Figure 1). The frequency of CLKOUT is one-fourth the crystal fundamental frequency. The crystal should be fundamental mode, and parallel resonant, with an effective series resistance of 30 ohms, a power dissipation of 1 mW, and be specified at a load capacitance of 20 pF.

PARAMETER	TEST CONDITIONS	TMS32010			UNIT
		MIN	NOM	MAX	
Crystal frequency, f_x	0°C to 70°C	6.7		20.5	MHz
C1, C2	0°C to 70°C		10		pF

external clock option

An external frequency source can be used by injecting the frequency directly into X2/CLKIN with X1 left unconnected. The external frequency injected must conform to the specifications listed in the table below.

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	TMS32010			UNIT
			MIN	NOM	MAX	
$t_{c(C)}$	CLKOUT cycle time†	R_L = 825 Ω, C_L = 100 pF, See Figure 2	195.12	200		ns
$t_{r(C)}$	CLKOUT rise time			10		ns
$t_{f(C)}$	CLKOUT fall time			8		ns
$t_{w(CL)}$	Pulse duration, CLKOUT low			92		ns
$t_{w(CH)}$	Pulse duration, CLKOUT high			90		ns
$t_{d(MCC)}$	Delay time CLKIN↑ to CLKOUT↓		25‡		60‡	ns

† $t_{c(C)}$ is the cycle time of CLKOUT, i.e., 4*$t_{c(MC)}$ (4 times CLKIN cycle time if an external oscillator is used).

‡ Values derived from characterization data and not tested.

timing requirements over recommended operating conditions

		TMS32010			UNIT
		MIN	NOM	MAX	
$t_{c(MC)}$	Master clock cycle time	48.78	50	150	ns
$t_{r(MC)}$	Rise time master clock input		5†	10†	ns
$t_{f(MC)}$	Fall time master clock input		5†	10†	ns
$t_{w(MCP)}$	Pulse duration master clock	0.475$t_{c(MC)}$†		0.525$t_{c(MC)}$†	ns
$t_{w(MCL)}$	Pulse duration master clock high, $t_{c(MC)}$ = 50 ns		20†		ns
$t_{w(MCH)}$	Pulse duration master clock high, $t_{c(MC)}$ = 50 ns		20†		ns

† Values derived from characterization data and not tested.

MEMORY AND PERIPHERAL INTERFACE TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	TMS32010			UNIT
			MIN	TYP	MAX	
t_{d1}	Delay time CLKOUT↓ to address bus valid	R_L = 825 Ω, C_L = 100 pF, See Figure 2	10†		50	ns
t_{d2}	Delay time CLKOUT↓ to $\overline{\text{MEN}}$↓		$\frac{1}{4}t_{c(C)}-5$†		$\frac{1}{4}t_{c(C)}+15$	ns
t_{d3}	Delay time CLKOUT↓ to $\overline{\text{MEN}}$↑		−10†		15	ns
t_{d4}	Delay time CLKOUT↓ to $\overline{\text{DEN}}$↓		$\frac{1}{4}t_{c(C)}-5$†		$\frac{1}{4}t_{c(C)}+15$	ns
t_{d5}	Delay time CLKOUT↓ to $\overline{\text{DEN}}$↑		−10†		15	ns
t_{d6}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↓		$\frac{1}{2}t_{c(C)}-5$†		$\frac{1}{2}t_{c(C)}+15$	ns
t_{d7}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↑		−10†		15	ns
t_{d8}	Delay time CLKOUT↓ to data bus OUT valid				$\frac{1}{4}t_{c(C)}+65$	ns
t_{d9}	Time after CLKOUT↓ that data bus starts to be driven		$\frac{1}{4}t_{c(C)}-5$†			ns
t_{d10}	Time after CLKOUT↓ that data bus stops being driven				$\frac{1}{4}t_{c(C)}+30$†	ns
t_v	Data bus OUT valid after CLKOUT↓		$\frac{1}{4}t_{c(C)}-10$			ns
$t_{h(A\text{-}WMD)}$	Address hold time after $\overline{\text{WE}}$↑, $\overline{\text{MEN}}$↑ or $\overline{\text{DEN}}$↑ (see Note 1)		0			ns
$t_{su(A\text{-}MD)}$	Address bus setup time prior to $\overline{\text{MEN}}$↓ or $\overline{\text{DEN}}$↓		$\frac{1}{4}t_{c(C)}-45$			ns

†Values derived from characterization data and not tested.

NOTE 1: Address bus will be valid upon $\overline{\text{WE}}$↑, $\overline{\text{DEN}}$↑, or $\overline{\text{MEN}}$↑.

timing requirements over recommended operating conditions

TEST CONDITIONS			TMS32010			UNIT
			MIN	NOM	MAX	
$t_{su(D)}$	Setup time data bus valid prior to CLKOUT↓	R_L = 825 Ω, C_L = 100 pF, See Figure 2	50			ns
$t_{h(D)}$	Hold time data bus held valid after CLKOUT↓ (see Note 2)		0			ns

NOTE 2: Data may be removed from the data bus upon $\overline{\text{MEN}}$↑ or $\overline{\text{DEN}}$↑ preceding CLKOUT↓.

RESET ($\overline{RS}$) TIMING

switching characteristics over recommended operating conditions

	PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d11}	Delay time $\overline{DEN}$↑, $\overline{WE}$↑, and $\overline{MEN}$↑ from $\overline{RS}$	R_L = 825 Ω, C_L = 100 pF, See Figure 2			½$t_{c(C)}$+50†	ns
$t_{dis(R)}$	Data bus disable time after $\overline{RS}$				¼$t_{c(C)}$+50†	ns

†Values derived from characterization data and not tested.

timing requirements over recommended operating conditions

		TMS32010			UNIT
		MIN	NOM	MAX	
$t_{su(R)}$	Reset $\overline{RS}$ setup time prior to CLKOUT (see Note 3)	50			ns
$t_{w(R)}$	$\overline{RS}$ pulse duration	5$t_{c(C)}$			ns

NOTE 3: $\overline{RS}$ can occur anytime during a clock cycle. Time given is minimum to ensure synchronous operation.

INTERRUPT ($\overline{INT}$) TIMING

timing requirements over recommended operating conditions

		TMS32010			UNIT
		MIN	NOM	MAX	
$t_{f(INT)}$	Fall time ($\overline{INT}$)			15	ns
$t_{w(INT)}$	Pulse duration $\overline{INT}$	$t_{c(C)}$			ns
$t_{su(INT)}$	Setup time $\overline{INT}$↓ before $\overline{CLKOUT}$↓	50			ns

I/O ($\overline{BIO}$) TIMING

timing requirements over recommended operating conditions

		TMS32010			UNIT
		MIN	NOM	MAX	
$t_{f(IO)}$	Fall time $\overline{BIO}$			15	ns
$t_{w(IO)}$	Pulse duration $\overline{BIO}$	$t_{c(C)}$			ns
$t_{su(IO)}$	Setup time $\overline{BIO}$↓ before CLKOUT↓	50			ns

CMOS DEVICE ELECTRICAL SPECIFICATIONS

This section contains all the electrical specifications for the TMS320 CMOS first-generation devices. Refer to the top corner for the specific device.

absolute maximum ratings over specified temperature range (unless otherwise noted)[†]

Supply voltage range, V_{CC}[‡]		−0.3 V to 7 V
Input voltage range		−0.3 V to 15 V
Output voltage range		−0.3 V to 15 V
Continuous power dissipation:	20-MHz version	0.3 W
	25-MHz version	0.35 W
Air temperature range above operating device:	L version	0°C to 70°C
	A version	−40°C to 85°C
Storage temperature range		−55°C to +150°C

[†] Stresses beyond 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 beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

[‡] All voltage values are with respect to V_{SS}.

recommended operating conditions

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage	EPROM devices	4.75	5	5.25	V
		All other devices	4.5	5	5.5	
V_{SS}	Supply voltage			0		V
V_{IH}	High-level input voltage	All inputs except CLKIN	2			V
		CLKIN	3			
V_{IL}	Low-level input voltage	All inputs except MC/$\overline{\text{MP}}$			0.8	V
		MC/$\overline{\text{MP}}$			0.6	
I_{OH}	High-level output current (all outputs)				−300	μA
I_{OL}	Low-level output current (all outputs)				2	mA
T_A	Operating free-air temperature	L version	0		70	°C
		A version	−40		85	°C

electrical characteristics over specified temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS		MIN	TYP[†]	MAX	UNIT
V_{OH}	High-level output voltage		I_{OH} = MAX		2.4	3		V
			I_{OH} = 20 μA (see Note 4)		V_{CC}−0.4[‡]			V
V_{OL}	Low-level output voltage		I_{OL} = MAX			0.3	0.5	V
I_{OZ}	Off-state output current		V_{CC} = MAX	V_O = 2.4 V			20	μA
				V_O = 0.4 V			−20	
I_I	Input current		V_I = V_{SS} to V_{CC}	All inputs except CLKIN			±20	μA
				CLKIN			±50	
C_i	Input capacitance	Data bus	f = 1 MHz, All other pins 0 V			25[‡]		pF
		All others				15[‡]		
C_o	Output capacitance	Data bus				25[‡]		pF
		All others				10[‡]		

[†] All typical values are at V_{CC} = 5 V, T_A = 25°C.

[‡] Values derived from characterization data and not tested.

NOTE 4: This voltage specification is included for interface to HC logic. However, note that all of the other timing parameters defined in this data sheet are specified for TTL logic levels and will differ for HC logic levels.

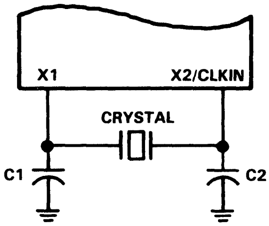


FIGURE 4. INTERNAL CLOCK OPTION

PARAMETER MEASUREMENT INFORMATION

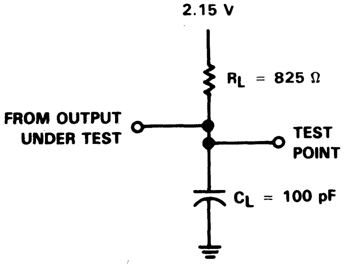


FIGURE 5. TEST LOAD CIRCUIT

electrical characteristics over specified temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
I_{CC}‡ Supply current	TMS320C10	f = 20.5 MHz, V_{CC} = 5.5 V, T_A = −40°C to 85°C		33	55	mA
	TMS320C10-25	f = 25.6 MHz, V_{CC} = 5.5 V, T_A = 0°C to 70°C		40	65	mA

†All typical values are at T_A = 70°C and are used for thermal resistance calculations.
‡I_{CC} characteristics are inversely proportional to temperature. For I_{CC} dependance on temperature, frequency, and loading, see Figure 9.

CLOCK CHARACTERISTICS AND TIMING

The TMS320C10/C10-25 can use either its internal oscillator or an external frequency source for a clock.

internal clock option

The internal oscillator is enabled by connecting a crystal across X1 and X2/CLKIN (see Figure 4). The frequency of CLKOUT is one-fourth the crystal fundamental frequency. The crystal should be fundamental mode, and parallel resonant, with an effective series resistance of 30 ohms, a power dissipation of 1 mW, and be specified at a load capacitance of 20 pF.

PARAMETER		TEST CONDITIONS	MIN	NOM	MAX	UNIT
Crystal frequency, f_x	TMS320C10	T_A = −40°C to 85°C	6.7		20.5	MHz
	TMS320C10-25	T_A = 0°C to 70°C	6.7		25.6	MHz
C1, C2		T_A = −40°C to 85°C		10		pF

external clock option

An external frequency source can be used by injecting the frequency directly into X2/CLKIN with X1 left unconnected. The external frequency injected must conform to the specifications listed in the table below.

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	TMS320C10			TMS320C10-25			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	
$t_{c(C)}$	CLKOUT cycle time†	R_L = 825 Ω, C_L = 100 pF, See Figure 5	195.12	200		156.25	160		ns
$t_{r(C)}$	CLKOUT rise time			10‡			10‡		ns
$t_{f(C)}$	CLKOUT fall time			8‡			8‡		ns
$t_{w(CL)}$	Pulse duration, CLKOUT low			92‡			72‡		ns
$t_{w(CH)}$	Pulse duration, CLKOUT high			90‡			70‡		ns
$t_{d(MCC)}$	Delay time CLKIN↑ to CLKOUT↓		25‡		60‡	25‡		50‡	ns

†$t_{c(C)}$ is the cycle time of CLKOUT, i.e., 4*$t_{c(MC)}$ (4 times CLKIN cycle time if an external oscillator is used).
‡Values derived from characterization data and not tested.

timing requirements over recommended operating conditions

		TMS320C10			TMS320C10-25			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
$t_{c(MC)}$	Master clock cycle time	48.78	50	150	39.06	40	150	ns
$t_{r(MC)}$	Rise time master clock input		5†	10†		5†	10†	ns
$t_{f(MC)}$	Fall time master clock input		5†	10†		5†	10†	ns
$t_{w(MCP)}$	Pulse duration master clock	0.4$t_{c(MC)}$†		0.6$t_{c(MC)}$†	0.45$t_{c(MC)}$†		0.55$t_{c(MC)}$†	ns
$t_{w(MCL)}$	Pulse duration master clock low		20†			15†		ns
$t_{w(MCH)}$	Pulse duration master clock high		20†			15†		ns

†Values derived from characterization data and not tested.

MEMORY AND PERIPHERAL INTERFACE TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	TMS320C10			TMS320C10-25			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{d1}	Delay time CLKOUT↓ to address bus valid	R_L = 825 Ω, C_L = 100 pF, See Figure 5	$10^{\dagger}$		50	$10^{\dagger}$		40	ns
t_{d2}	Delay time CLKOUT↓ to $\overline{\text{MEN}}$↓		$\frac{1}{4}t_{c(C)}-5^{\dagger}$		$\frac{1}{4}t_{c(C)}+15$	$\frac{1}{4}t_{c(C)}-5^{\dagger}$		$\frac{1}{4}t_{c(C)}+12$	ns
t_{d3}	Delay time CLKOUT↓ to $\overline{\text{MEN}}$↑		$-10^{\dagger}$		15	$-10^{\dagger}$		12	ns
t_{d4}	Delay time CLKOUT↓ to $\overline{\text{DEN}}$↓		$\frac{1}{4}t_{c(C)}-5^{\dagger}$		$\frac{1}{4}t_{c(C)}+15$	$\frac{1}{4}t_{c(C)}-5^{\dagger}$		$\frac{1}{4}t_{c(C)}+12$	ns
t_{d5}	Delay time CLKOUT↓ to $\overline{\text{DEN}}$↑		$-10^{\dagger}$		15	$-10^{\dagger}$		12	ns
t_{d6}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↓		$\frac{1}{4}t_{c(C)}-5^{\dagger}$		$\frac{1}{4}t_{c(C)}+15$	$\frac{1}{2}t_{c(C)}-5^{\dagger}$		$\frac{1}{2}t_{c(C)}+12$	ns
t_{d7}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↑		$-10^{\dagger}$		15	$-10^{\dagger}$		12	ns
t_{d8}	Delay time CLKOUT↓ to data bus OUT valid				$\frac{1}{4}t_{c(C)}+65$			$\frac{1}{4}t_{c(C)}+52$	ns
t_{d9}	Time after CLKOUT↓ that data bus starts to be driven		$\frac{1}{4}t_{c(C)}-5^{\dagger}$			$\frac{1}{4}t_{c(C)}-5^{\dagger}$			ns
t_{d10}	Time after CLKOUT↓ that data bus stops being driven				$\frac{1}{4}t_{c(C)}+40^{\dagger}$			$\frac{1}{4}t_{c(C)}+40^{\dagger}$	ns
t_v	Data bus OUT valid after CLKOUT↓		$\frac{1}{4}t_{c(C)}-10$			$\frac{1}{4}t_{c(C)}-10$			ns
$t_{h(A\text{-}WMD)}$	Address hold time after $\overline{\text{WE}}$↑, $\overline{\text{MEN}}$↑, or $\overline{\text{DEN}}$↑ (see Note 5)		$-10^{\dagger}$			$-10^{\dagger}$			ns
$t_{su(A\text{-}MD)}$	Address bus setup time prior to $\overline{\text{MEN}}$↓ or $\overline{\text{DEN}}$↓		$\frac{1}{4}t_{c(C)}-45$			$\frac{1}{4}t_{c(C)}-35$			ns

† Values derived from characterization data and not tested.

NOTE 5: For interfacing I/O devices, see Figure 6.

timing requirements over recommended operating conditions

		TEST CONDITIONS	TMS320C10			TMS320C10-25			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	
$t_{su(D)}$	Setup time data bus valid prior to CLKOUT↓	R_L = 825 Ω, C_L = 100 pF, See Figure 5	50			40			ns
$t_{h(D)}$	Hold time data bus held valid after CLKOUT↓ (see Note 2)		0			0			ns

NOTE 2: Data may be removed from the data bus upon $\overline{\text{MEN}}$↑ or $\overline{\text{DEN}}$↑ preceding CLKOUT↓.

SUGGESTED I/O DECODE CIRCUIT

The circuit shown in Figure 6 is a design example for interfacing I/O devices to the TMS320C10/C10-25. This circuit decodes the address for output operations using the OUT instruction. The same circuit can be used to decode input and output operations if the inverter ('ALS04) is replaced with a NAND gate and both $\overline{\text{DEN}}$ and $\overline{\text{WE}}$ are connected. Inputs and outputs can be decoded at the same port provided the output of the decoder ('AS137) is gated with the appropriate signal ($\overline{\text{DEN}}$ or $\overline{\text{WE}}$) to select read or write (using an 'ALS32). Access times can be increased when the circuit shown in Figure 6 is repeated to support IN instructions with $\overline{\text{DEN}}$ connected rather than $\overline{\text{WE}}$.

The table write (TBLW) function requires a different circuit. A detailed discussion of an example circuit for this function is described on page 315 of the application report, "Interfacing External Memory to the TMS32010," published in the book, *Digital Signal Processing Applications with the TMS320 Family* (SPRA012A). A schematic of this circuit is shown on page 318 of that book.

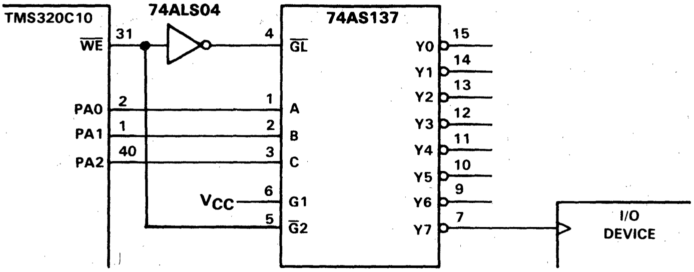


FIGURE 6. I/O DECODE CIRCUIT

RESET ($\overline{\text{RS}}$) TIMING

switching characteristics over recommended operating conditions

PARAMETER	TEST CONDITIONS	MIN TYP MAX	UNIT
t_{d11} Delay time $\overline{\text{DEN}}$↑, $\overline{\text{WE}}$↑, and $\overline{\text{MEN}}$↑ from $\overline{\text{RS}}$	R_L = 825 Ω, C_L = 100 pF, See Figure 5	$\frac{1}{2}t_{c(C)} + 50$†	ns
$t_{dis(R)}$ Data bus disable time after $\overline{\text{RS}}$		$\frac{1}{4}t_{c(C)} + 50$†	ns

†These values were derived from characterization data and not tested.

timing requirements over recommended operating conditions

	TMS320C10			TMS320C10-25			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
$t_{su(R)}$ Reset ($\overline{\text{RS}}$) setup time prior to CLKOUT (see Note 3)	50			40			ns
$t_{w(R)}$ $\overline{\text{RS}}$ pulse duration	$5t_{c(C)}$			$5t_{c(C)}$			ns

NOTE 3: $\overline{\text{RS}}$ can occur anytime during a clock cycle. Time given is minimum to ensure synchronous operation.

INTERRUPT ($\overline{\text{INT}}$) TIMING

timing requirements over recommended operating conditions

	TMS320C10			TMS320C10-25			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
$t_{f(INT)}$ Fall time $\overline{\text{INT}}$			15			15	ns
$t_{w(INT)}$ Pulse duration $\overline{\text{INT}}$	$t_{c(C)}$			$t_{c(C)}$			ns
$t_{su(INT)}$ Setup time $\overline{\text{INT}}$↓ before CLKOUT↓	50			40			ns

I/O ($\overline{\text{BIO}}$) TIMING

timing requirements over recommended operating conditions

	TMS320C10			TMS320C10-25			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
$t_{f(IO)}$ Fall time $\overline{\text{BIO}}$			15			15	ns
$t_{w(IO)}$ Pulse duration $\overline{\text{BIO}}$	$t_{c(C)}$			$t_{c(C)}$			ns
$t_{su(IO)}$ Setup time $\overline{\text{BIO}}$↓ before CLKOUT↓	50			40			ns

electrical characteristics over specified temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
I_{CC}‡ Supply current	f = 14.4 MHz, V_{CC} = 5.5 V, T_A = 0°C to 70°C		28	55	mA

†All typical values are at T_A = 70°C and are used for thermal resistance calculations.
‡I_{CC} characteristics are inversely proportional to temperature. For I_{CC} dependence on temperature, frequency, and loading, see Figure 9.

CLOCK CHARACTERISTICS AND TIMING

The TMS320C10-14 can use either its internal oscillator or an external frequency source for a clock.

internal clock option

The internal oscillator is enabled by connecting a crystal across X1 and X2/CLKIN (see Figure 4). The frequency of CLKOUT is one-fourth the crystal fundamental frequency. The crystal should be fundamental mode, and parallel resonant, with an effective series resistance of 30 ohms, a power dissipation of 1 mW, and be specified at a load capacitance of 20 pF.

PARAMETER	TEST CONDITIONS	MIN	NOM	MAX	UNIT
Crystal frequency, f_x	T_A = 0°C to 70°C	6.7		14.4	MHz
C1, C2	T_A = 0°C to 70°C		10		pF

external clock option

An external frequency source can be used by injecting the frequency directly into X2/CLKIN with X1 left unconnected. The external frequency injected must conform to the specifications listed in the table below.

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	NOM	MAX	UNIT
$t_{c(C)}$	CLKOUT cycle time†	R_L = 825 Ω, C_L = 100 pF, See Figure 5	277.78			ns
$t_{r(C)}$	CLKOUT rise time			10		ns
$t_{f(C)}$	CLKOUT fall time			8		ns
$t_{w(CL)}$	Pulse duration, CLKOUT low			131		ns
$t_{w(CH)}$	Pulse duration, CLKOUT high			129		ns
$t_{d(MCC)}$	Delay time CLKIN↑ to CLKOUT↓		25‡		60‡	ns

†$t_{c(C)}$ is the cycle time of CLKOUT, i.e., 4*$t_{c(MC)}$ (4 times CLKIN cycle time if an external oscillator is used).
‡Values derived from characterization data and not tested.

timing requirements over recommended operating conditions

		MIN	NOM	MAX	UNIT
$t_{c(MC)}$	Master clock cycle time	69.5		150	ns
$t_{r(MC)}$	Rise time master clock input		5†	10†	ns
$t_{f(MC)}$	Fall time master clock input		5†	10†	ns
$t_{w(MCP)}$	Pulse duration master clock	0.4$t_{c(MC)}$†		0.6$t_{c(MC)}$†	ns
$t_{w(MCL)}$	Pulse duration master clock low, $t_{c(MC)}$ = 50 ns		20†		ns
$t_{w(MCH)}$	Pulse duration master clock high, $t_{c(MC)}$ = 50 ns		20†		ns

†Values derived from characterization data and not tested.

MEMORY AND PERIPHERAL INTERFACE TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d1}	Delay time CLKOUT↓ to address bus valid	R_L = 825 Ω, C_L = 100 pF, See Figure 5	10†		50	ns
t_{d2}	Delay time CLKOUT↓ to $\overline{\text{MEN}}$↓		$\frac{1}{4}t_{c(C)}-5$†		$\frac{1}{4}t_{c(C)}+15$	ns
t_{d3}	Delay time CLKOUT↓ to $\overline{\text{MEN}}$↑		-10†		15	ns
t_{d4}	Delay time CLKOUT↓ to $\overline{\text{DEN}}$↓		$\frac{1}{4}t_{c(C)}-5$†		$\frac{1}{4}t_{c(C)}+15$	ns
t_{d5}	Delay time CLKOUT↓ to $\overline{\text{DEN}}$↑		-10†		15	ns
t_{d6}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↓		$\frac{1}{2}t_{c(C)}-5$†		$\frac{1}{2}t_{c(C)}+15$	ns
t_{d7}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↑		-10†		15	ns
t_{d8}	Delay time CLKOUT↓ to data bus OUT valid				$\frac{1}{4}t_{c(C)}+65$	ns
t_{d9}	Time after CLKOUT↓ that data bus starts to be driven		$\frac{1}{4}t_{c(C)}-5$†			ns
t_{d10}	Time after CLKOUT↓ that data bus stops being driven				$\frac{1}{4}t_{c(C)}+40$†	ns
t_v	Data bus OUT valid after CLKOUT↓		$\frac{1}{4}t_{c(C)}-10$			ns
$t_{h(A\text{-}WMD)}$	Address hold time after $\overline{\text{WE}}$↑, $\overline{\text{MEN}}$↑, or $\overline{\text{DEN}}$↑ (see Note 5)		-10†			ns
$t_{su(A\text{-}MD)}$	Address bus setup time prior to $\overline{\text{MEN}}$↓ or $\overline{\text{DEN}}$↓		$\frac{1}{4}t_{c(C)}-45$			ns

†Values derived from characterization data and not tested.

NOTE 5: For interfa^-'ng I/O devices, see Figure 6.

timing requirements over recommended operating conditions

		TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{su(D)}$	Setup time data bus valid prior to CLKOUT↓	R_L = 825 Ω, C_L = 100 pF, See Figure 5	50			ns
$t_{h(D)}$	Hold time data bus held valid after CLKOUT↓ (see Note 2)		0			ns

NOTE 2: Data may be removed from the data bus upon $\overline{\text{MEN}}$↑ or $\overline{\text{DEN}}$↑ preceding CLKOUT↓.

RESET ($\overline{\text{RS}}$) TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d11}	Delay time $\overline{\text{DEN}}\uparrow$, $\overline{\text{WE}}\uparrow$, and $\overline{\text{MEN}}\uparrow$ from $\overline{\text{RS}}$	R_L = 825 Ω, C_L = 100 pF, See Figure 5			$\frac{1}{2}t_{c(C)} + 50$†	ns
$t_{dis(R)}$	Data bus disable time after $\overline{\text{RS}}$				$\frac{1}{4}t_{c(C)} + 50$†	ns

†These values were derived from characterization data and not tested.

timing requirements over recommended operating conditions

		MIN	NOM	MAX	UNIT
$t_{su(R)}$	Reset ($\overline{\text{RS}}$) setup time prior to CLKOUT (see Note 3)	50			ns
$t_{w(R)}$	$\overline{\text{RS}}$ pulse duration	$5t_{c(C)}$			ns

NOTE 3: $\overline{\text{RS}}$ can occur anytime during a clock cycle. Time given is minimum to ensure synchronous operation.

INTERRUPT ($\overline{\text{INT}}$) TIMING

timing requirements over recommended operating conditions

		MIN	NOM	MAX	UNIT
$t_{f(INT)}$	Fall time $\overline{\text{INT}}$			15	ns
$t_{w(INT)}$	Pulse duration $\overline{\text{INT}}$	$t_{c(C)}$			ns
$t_{su(INT)}$	Setup time $\overline{\text{INT}}\downarrow$ before CLKOUT$\downarrow$	50			ns

I/O ($\overline{\text{BIO}}$) TIMING

timing requirements over recommended operating conditions

		MIN	NOM	MAX	UNIT
$t_{f(IO)}$	Fall time $\overline{\text{BIO}}$			15	ns
$t_{w(IO)}$	Pulse duration $\overline{\text{BIO}}$	$t_{c(C)}$			ns
$t_{su(IO)}$	Setup time $\overline{\text{BIO}}\downarrow$ before CLKOUT$\downarrow$	50			ns

electrical characteristics over specified temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP[†]	MAX	UNIT
I_{CC}[‡] Supply current	TMS320C15	f = 20.5 MHz, V_{CC} = 5.5 V, T_A = 0°C to 70°C		45	55	mA
	TMS320C15-25	f = 25.6 MHz, V_{CC} = 5.5 V, T_A = 0°C to 70°C		50	65	
	TMS320E15	f = 20.5 MHz, V_{CC} = 5.25 V, T_A = −40°C to 85°C		55	75	
	TMS320E15-25	f = 25.6 MHz, V_{CC} = 5.25 V, T_A = 0°C to 70°C		65	85	

[†] All typical values are at T_A = 70°C and are used for thermal resistance calculations.
[‡] I_{CC} characteristics are inversely proportional to temperature. For I_{CC} dependence on temperature, frequency, and loading, see Figure 9.

CLOCK CHARACTERISTICS AND TIMING

The TMS320C15/E15 can use either its internal oscillator or an external frequency source for a clock.

internal clock option

The internal oscillator is enabled by connecting a crystal across X1 and X2/CLKIN (see Figure 4). The frequency of CLKOUT is one-fourth the crystal fundamental frequency. The crystal should be fundamental mode, and parallel resonant, with an effective series resistance of 30 ohms, a power dissipation of 1 mW, and be specified at a load capacitance of 20 pF.

PARAMETER		TEST CONDITIONS	MIN	NOM	MAX	UNIT
Crystal frequency, f_x	TMS320C15	T_A = 0°C to 70°C	6.7		20.5	MHz
	TMS320E15	T_A = −40°C to 85°C	6.7		20.5	MHz
	TMS320C15-25/E15-25	T_A = 0°C to 70°C	6.7		25.6	MHz
C1, C2		T_A = 0°C to 70°C		10		pF

external clock option

An external frequency source can be used by injecting the frequency directly into X2/CLKIN with X1 left unconnected. The external frequency injected must conform to the specifications listed in the table below.

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	TMS320C15/E15			TMS320C15-25/E15-25			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	
$t_{c(C)}$	CLKOUT cycle time[†]	R_L = 825 Ω, C_L = 100 pF, See Figure 5	195.12	200		156.25	160		ns
$t_{r(C)}$	CLKOUT rise time			10[‡]			10[‡]		ns
$t_{f(C)}$	CLKOUT fall time			8[‡]			8[‡]		ns
$t_{w(CL)}$	Pulse duration, CLKOUT low			92[‡]			72[‡]		ns
$t_{w(CH)}$	Pulse duration, CLKOUT high			90[‡]			70[‡]		ns
$t_{d(MCC)}$	Delay time CLKIN↑ to CLKOUT↓		25[‡]		60[‡]	25[‡]		50[‡]	ns

[†] $t_{c(C)}$ is the cycle time of CLKOUT, i.e., 4*$t_{c(MC)}$ (4 times CLKIN cycle time if an external oscillator is used).
[‡] Values derived from characterization data and not tested.

timing requirements over recommended operating conditions

		TMS320C15/E15			TMS320C15-25/E15-25			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
$t_{c(MC)}$	Master clock cycle time	48.78	50	150	39.06	40	150	ns
$t_{r(MC)}$	Rise time master clock input		5†	10†		5†	10†	ns
$t_{f(MC)}$	Fall time master clock input		5†	10†		5†	10†	ns
$t_{w(MCP)}$	Pulse duration master clock	$0.4t_{c(MC)}$†		$0.6t_{c(MC)}$†	$0.45t_{c(MC)}$†		$0.55t_{c(MC)}$†	ns
$t_{w(MCL)}$	Pulse duration master clock low		20†			15†		ns
$t_{w(MCH)}$	Pulse duration master clock high		20†			15†		ns

†Values derived from characterization data and not tested.

MEMORY AND PERIPHERAL INTERFACE TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	TMS320C15/E15			TMS320C15-25/E15-25			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{d1}	Delay time CLKOUT↓ to address bus valid	R_L = 825 Ω, C_L = 100 pF, See Figure 5	10†		50	10†		40	ns
t_{d2}	Delay time CLKOUT↓ to $\overline{\text{MEN}}$↓		$\frac{1}{4}t_{c(C)}-5$†		$\frac{1}{4}t_{c(C)}+15$	$\frac{1}{4}t_{c(C)}-5$†		$\frac{1}{4}t_{c(C)}+12$	ns
t_{d3}	Delay time CLKOUT↓ to $\overline{\text{MEN}}$↑		−10†		15	−10†		12	ns
t_{d4}	Delay time CLKOUT↓ to $\overline{\text{DEN}}$↓		$\frac{1}{4}t_{c(C)}-5$†		$\frac{1}{4}t_{c(C)}+15$	$\frac{1}{4}t_{c(C)}-5$†		$\frac{1}{4}t_{c(C)}+12$	ns
t_{d5}	Delay time CLKOUT↓ to $\overline{\text{DEN}}$↑		−10†		15	−10†		12	ns
t_{d6}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↓		$\frac{1}{2}t_{c(C)}-5$†		$\frac{1}{2}t_{c(C)}+15$	$\frac{1}{2}t_{c(C)}-5$†		$\frac{1}{2}t_{c(C)}+12$	ns
t_{d7}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↑		−10†		15	−10†		12	ns
t_{d8}	Delay time CLKOUT↓ to data bus OUT valid				$\frac{1}{4}t_{c(C)}+65$			$\frac{1}{4}t_{c(C)}+52$	ns
t_{d9}	Time after CLKOUT↓ that data bus starts to be driven		$\frac{1}{4}t_{c(C)}-5$†			$\frac{1}{4}t_{c(C)}-5$†			ns
t_{d10}	Time after CLKOUT↓ that data bus stops being driven (TMS320C15/C15-25 only)				$\frac{1}{4}t_{c(C)}+40$†			$\frac{1}{4}t_{c(C)}+40$†	ns
t_{d10}	Time after CLKOUT↓ that data bus stops being driven (TMS320E15/E15-25 only)				$\frac{1}{4}t_{c(C)}+70$†			$\frac{1}{4}t_{c(C)}+70$†	ns
t_v	Data bus OUT valid after CLKOUT↓		$\frac{1}{4}t_{c(C)}-10$			$\frac{1}{4}t_{c(C)}-10$			ns
$t_{h(A-WMD)}$	Address hold time after $\overline{\text{WE}}$↑, $\overline{\text{MEN}}$↑, or $\overline{\text{DEN}}$↑ (see Note 1)		0†	2†		0†	2†		ns
$t_{su(A-MD)}$	Address bus setup time prior to $\overline{\text{DEN}}$↓		$\frac{1}{4}t_{c(C)}-45$			$\frac{1}{4}t_{c(C)}-35$			ns

†Values derived from characterization data and not tested.

NOTE 1: Address bus will be valid upon $\overline{\text{WE}}$↑, $\overline{\text{DEN}}$↑, or $\overline{\text{MEN}}$↑.

timing requirements over recommended operating conditions

		TEST CONDITIONS	TMS320C15/E15			TMS320C15-25/E15-25			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	
$t_{su(D)}$	Setup time data bus valid prior to CLKOUT↓	R_L = 825 Ω, C_L = 100 pF, See Figure 5	50			40			ns
$t_{h(D)}$	Hold time data bus held valid after CLKOUT↓ (see Note 2)		0			0			ns

NOTE 2: Data may be removed from the data bus upon $\overline{\text{MEM}}$↑ or $\overline{\text{DEN}}$↑ preceding CLKOUT↓.

RESET ($\overline{RS}$) TIMING

switching characteristics over recommended operating conditions

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d11} Delay time $\overline{DEN}$↑, $\overline{WE}$↑, and $\overline{MEN}$↑ from $\overline{RS}$	R_L = 825 Ω, C_L = 100 pF, See Figure 5			$\frac{1}{2} t_{c(C)} + 50$†	ns
$t_{dis(R)}$ Data bus disable time after $\overline{RS}$				$\frac{1}{4} t_{c(C)} + 50$†	ns

†Values derived from characterization data and not tested.

timing requirements over recommended operating conditions

	TMS320C15/E15			TMS320C15-25/E15-25			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
$t_{su(R)}$ Reset ($\overline{RS}$) setup time prior to CLKOUT (see Note 3)	50			40			ns
$t_{w(R)}$ $\overline{RS}$ pulse duration	$5t_{c(C)}$			$5t_{c(C)}$			ns

NOTE 3: $\overline{RS}$ can occur anytime during a clock cycle. Time given is minimum to ensure synchronous operation.

INTERRUPT ($\overline{INT}$) TIMING

timing requirements over recommended operating conditions

	TMS320C15/E15			TMS320C15-25/E15-25			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
$t_{f(INT)}$ Fall time $\overline{INT}$			15			15	ns
$t_{w(INT)}$ Pulse duration $\overline{INT}$	$t_{c(C)}$			$t_{c(C)}$			ns
$t_{su(INT)}$ Setup time $\overline{INT}$↓ before CLKOUT↓	50			40			ns

I/O ($\overline{BIO}$) TIMING

timing requirements over recommended operating conditions

	TMS320C15/E15			TMS320C15-25/E15-25			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
$t_{f(IO)}$ Fall time $\overline{BIO}$			15			15	ns
$t_{w(IO)}$ Pulse duration $\overline{BIO}$	$t_{c(C)}$			$t_{c(C)}$			ns
$t_{su(IO)}$ Setup time $\overline{BIO}$↓ before CLKOUT↓	50			40			ns

electrical characteristics over specified temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP[†]	MAX	UNIT
I_{CC}[‡] Supply current	TMS320C17	f = 20.5 MHz, V_{CC} = 5.5 V, T_A = 0°C to 70°C		50	65	mA
	TMS320E17	f = 20.5 MHz, V_{CC} = 5.25 V, T_A = −40°C to 85°C		55	75	

[†] All typical values are at T_A = 70°C and are used for thermal resistance calculations.
[‡] I_{CC} characteristics are inversely proportional to temperature. For I_{CC} dependance on temperature, frequency, and loading, see Figure 9.

CLOCK CHARACTERISTICS AND TIMING

The TMS320C17/E17 can use either its internal oscillator or an external frequency source for a clock.

internal clock option

The internal oscillator is enabled by connecting a crystal across X1 and X2/CLKIN (see Figure 4). The frequency of CLKOUT is one-fourth the crystal fundamental frequency. The crystal should be fundamental mode, and parallel resonant, with an effective series resistance of 30 ohms, a power dissipation of 1 mW, and be specified at a load capacitance of 20 pF.

PARAMETER		TEST CONDITIONS	MIN	NOM	MAX	UNIT
Crystal frequency, f_x	TMS320C17	T_A = 0°C to 70°C	6.7		20.5	MHz
	TMS320E17	T_A = −40°C to 85°C	6.7		20.5	MHz
C1, C2		T_A = 0°C to 70°C		10		pF

external clock option

An external frequency source can be used by injecting the frequency directly into X2/CLKIN with X1 left unconnected. The external frequency injected must conform to the specifications listed in the table below.

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	TMS320C17/E17			UNIT
			MIN	NOM	MAX	
$t_{c(C)}$	CLKOUT cycle time[†]	R_L = 825 Ω, C_L = 100 pF, See Figure 5	195.12	200		ns
$t_{r(C)}$	CLKOUT rise time			10[‡]		ns
$t_{f(C)}$	CLKOUT fall time			8[‡]		ns
$t_{w(CL)}$	Pulse duration, CLKOUT low			92[‡]		ns
$t_{w(CH)}$	Pulse duration, CLKOUT high			90[‡]		ns
$t_{d(MCC)}$	Delay time CLKIN↑ to CLKOUT↓		25[‡]		60[‡]	ns

[†] $t_{c(C)}$ is the cycle time of CLKOUT, i.e., 4*$t_{c(MC)}$ (4 times CLKIN cycle time if an external oscillator is used).
[‡] Values derived from characterization data and not tested.

timing requirements over recommended operating conditions

		TMS320C17/E17			UNIT
		MIN	NOM	MAX	
$t_{c(MC)}$	Master clock cycle time	48.78	50	150	ns
$t_{r(MC)}$	Rise time master clock input		5[†]	10[†]	ns
$t_{f(MC)}$	Fall time master clock input		5[†]	10[†]	ns
$t_{w(MCP)}$	Pulse duration master clock	$0.45t_{c(MC)}$[†]		$0.6t_{c(MC)}$[†]	ns
$t_{w(MCL)}$	Pulse duration master clock low		20[†]		ns
$t_{w(MCH)}$	Pulse duration master clock high		20[†]		ns

[†] Values derived from characterization data and not tested.

MEMORY AND PERIPHERAL INTERFACE TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	TMS320C17/E17			UNIT
			MIN	NOM	MAX	
t_{d1}	Delay time CLKOUT↓ to address bus valid	R_L = 825 Ω, C_L = 100 pF, See Figure 5	10[†]		50	ns
t_{d4}	Delay time CLKOUT↓ to $\overline{\text{DEN}}$↓		$\frac{1}{4}t_{c(C)} - 5$[†]		$\frac{1}{4}t_{c(C)} + 15$	ns
t_{d5}	Delay time CLKOUT↓ to $\overline{\text{DEN}}$↑		-10[†]		15	ns
t_{d6}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↓		$\frac{1}{2}t_{c(C)} - 5$[†]		$\frac{1}{2}t_{c(C)} + 15$	ns
t_{d7}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↑		-10[†]		15	ns
t_{d8}	Delay time CLKOUT↓ to data bus OUT valid				$\frac{1}{4}t_{c(C)} + 65$	ns
t_{d9}	Time after CLKOUT↓ that data bus starts to be driven		$\frac{1}{4}t_{c(C)} - 5$[†]			ns
t_{d10}	Time after CLKOUT↓ that data bus stops being driven				$\frac{1}{4}t_{c(C)} + 70$[†]	ns
t_v	Data bus OUT valid after CLKOUT↓		$\frac{1}{4}t_{c(C)} - 10$			ns
$t_{h(A\text{-}WMD)}$	Address hold time after $\overline{\text{WE}}$↑ or $\overline{\text{DEN}}$↑ (see Note 1)		0[†]	2[†]		ns
$t_{su(A\text{-}MD)}$	Address bus setup time prior to $\overline{\text{DEN}}$↓		$\frac{1}{4}t_{c(C)} - 45$			ns

[†] Values derived from characterization data and not tested.

NOTE 1: Address bus will be valid upon $\overline{\text{WE}}$↑, $\overline{\text{DEN}}$↑, or $\overline{\text{MEN}}$↑.

timing requirements over recommended operating conditions

		TEST CONDITIONS	TMS320C17/E17			UNIT
			MIN	NOM	MAX	
$t_{su(D)}$	Setup time data bus valid prior to CLKOUT↓	R_L = 825 Ω, C_L = 100 pF, See Figure 5	50			ns
$t_{h(D)}$	Hold time data bus held valid after CLKOUT↓ (see Note 2)		0			ns

NOTE 2: Data may be removed from the data bus upon $\overline{\text{DEN}}$↑ preceding CLKOUT↓.

RESET ($\overline{RS}$) TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN TYP MAX	UNIT
t_{d11}	Delay time $\overline{DEN}$↑ and $\overline{WE}$↑ from $\overline{RS}$	R_L = 825 Ω, C_L = 100 pF, See Figure 5	½$t_{c(C)}$+50†	ns
$t_{dis(R)}$	Data bus disable time after $\overline{RS}$		¼$t_{c(C)}$+50†	ns
t_{d12}	Delay time from $\overline{RS}$↓ to high-impedance SCLK		200†	ns
t_{d13}	Delay time from $\overline{RS}$↓ to high-impedance DX1, DX0		200†	ns

†Values derived from characterization data and not tested.

timing requirements over recommended operating conditions

		TMS320C17/E17			UNIT
		MIN	NOM	MAX	
$t_{su(R)}$	Reset ($\overline{RS}$) setup time prior to CLKOUT (see Note 3)	50			ns
$t_{w(R)}$	$\overline{RS}$ pulse duration	5$t_{c(C)}$			ns

NOTE 3: RS can occur anytime during a clock cycle. Time given is minimum to ensure synchronous operation.

INTERRUPT ($\overline{EXINT}$) TIMING

timing requirements over recommended operating conditions

		TMS320C17/E17			UNIT
		MIN	NOM	MAX	
$t_{f(INT)}$	Fall time $\overline{EXINT}$			15	ns
$t_{w(INT)}$	Pulse duration $\overline{EXINT}$	$t_{c(C)}$			ns
$t_{su(INT)}$	Setup time $\overline{EXINT}$↓ before CLKOUT↓	50			ns

I/O ($\overline{BIO}$) TIMING

timing requirements over recommended operating conditions

		TMS320C17/E17			UNIT
		MIN	NOM	MAX	
$t_{f(IO)}$	Fall time $\overline{BIO}$			15	ns
$t_{w(IO)}$	Pulse duration $\overline{BIO}$	$t_{c(C)}$			ns
$t_{su(IO)}$	Setup time $\overline{BIO}$↓ before CLKOUT↓	50			ns

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{d(XF)}$	Delay time CLKOUT↓ to valid XF	R_L = 825 Ω, C_L = 100 pF, See Figure 5	5†		115	ns

†Values derived from characterization data and not tested.

SERIAL PORT TIMING

switching characteristics over recommended operating conditions

PARAMETER		MIN	TYP	MAX	UNIT
$t_{d(CH\text{-}FR)}$	Internal framing (FR) delay from SCLK rising edge			70	ns
$t_{d(DX1\text{-}CL)}$	DX bit 1 valid before SCLK falling edge	20			ns
$t_{d(DX2\text{-}CL)}$	DX bit 2 valid before SCLK falling edge	20			ns
$t_{h(DX)}$	DX hold time after SCLK falling edge	$t_{c(SCLK)}/2$			ns

timing requirements over recommended operating conditions, f = 25 MHz

		MIN	NOM	MAX	UNIT
$t_{c(SCLK)}$	Serial port clock (SCLK) cycle time (see Note 6)	390		4770	ns
$t_{f(SCLK)}$	Serial port clock (SCLK) fall time			30†	ns
$t_{r(SCLK)}$	Serial port clock (SCLK) rise time			30†	ns
$t_{w(SCLKL)}$	Serial port clock (SCLK) low-pulse duration (see Note 7)	185		2500	ns
$t_{w(SCLKH)}$	Serial port clock (SCLK) high-pulse duration (see Note 7)	185		2500	
$t_{su(FS)}$	$\overline{FSX}/\overline{FSR}$ setup time before SCLK falling edge	100			ns
$t_{su(DR)}$	DR setup time before SCLK falling edge	20			ns
$t_{h(DR)}$	DR hold time after SCLK falling edge	20			ns

†Values derived from characterization data and not tested.

NOTES: 6. Minimum cycle time is $2t_{c(C)}$ where $t_{c(C)}$ is CLKOUT cycle time.
7. The duty cycle of the serial port clock must be within 45 to 55 percent.

COPROCESSOR INTERFACE TIMING

switching characteristics over recommended operating conditions

PARAMETER		MIN	NOM	MAX	UNIT
$t_{d(R\text{-}A)}$	$\overline{RD}$ low to $\overline{TBLF}$ high			75	ns
$t_{d(W\text{-}A)}$	$\overline{WR}$ low to $\overline{RBLE}$ high			75	ns
$t_{a(RD)}$	$\overline{RD}$ low to data valid			80	ns
$t_{h(RD)}$	Data hold time after $\overline{RD}$ high	25			ns

timing requirements over recommended operating conditions

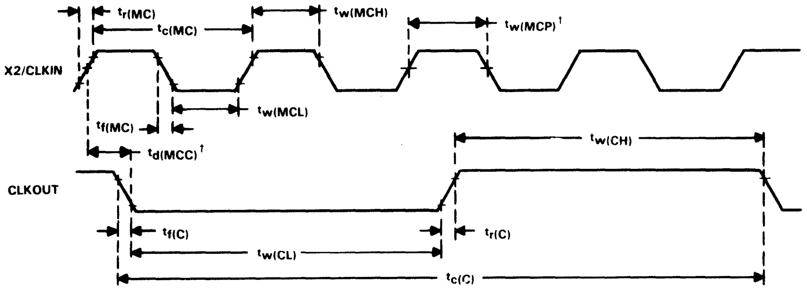
		MIN	NOM	MAX	UNIT
$t_{h(HL)}$	HI/$\overline{LO}$ hold time after $\overline{WR}$ or $\overline{RD}$ high	25			ns
$t_{su(HL)}$	HI/$\overline{LO}$ setup time prior to $\overline{WR}$ or $\overline{RD}$ low	40			ns
$t_{su(WR)}$	Data setup time prior to $\overline{WR}$ high	30			ns
$t_{h(WR)}$	Data hold time after $\overline{WR}$ high	25			ns
$t_{w(RDL)}$	$\overline{RD}$ low-pulse duration	80			ns
$t_{w(WRL)}$	$\overline{WR}$ low-pulse duration	60			ns

TIMING DIAGRAMS

This section contains all the timing diagrams for the TMS320 first-generation devices. For a specific device, refer to the top corner of pages 46-51.

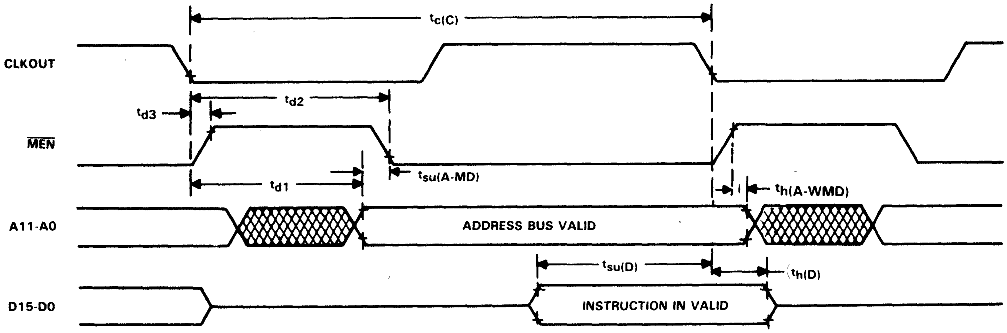
Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

clock timing

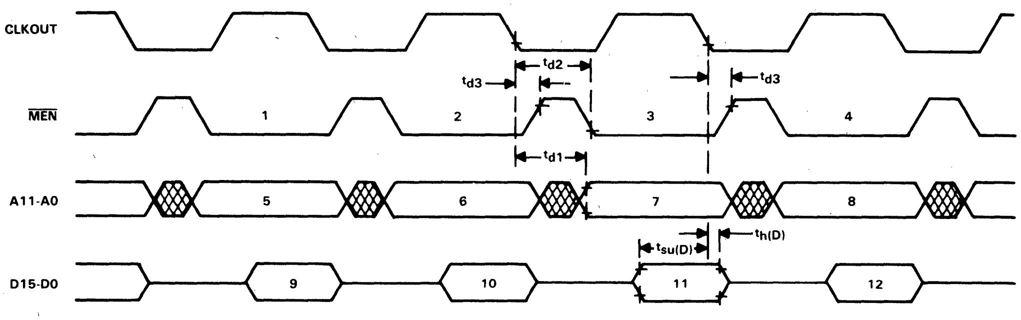


†$t_{d(MCC)}$ and $t_{w(MCP)}$ are referenced to an intermediate level of 1.5 volts on the CLKIN waveform.

memory read timing



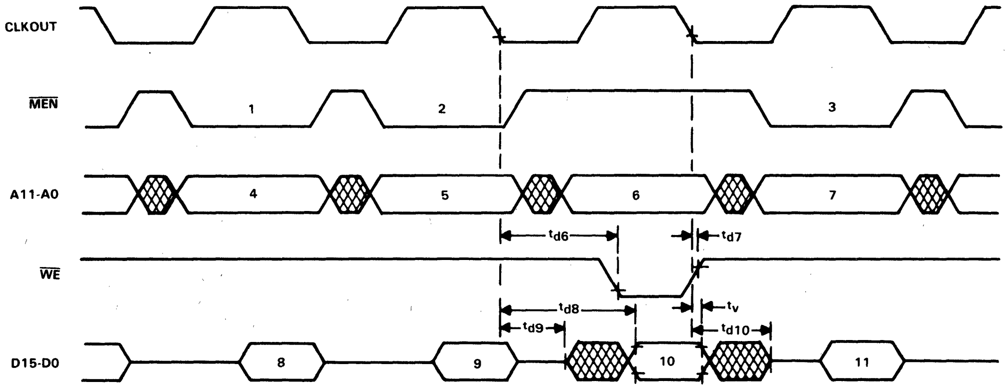
TBLR instruction timing



LEGEND:

1. TBLR INSTRUCTION PREFETCH
2. DUMMY PREFETCH
3. DATA FETCH
4. NEXT INSTRUCTION PREFETCH
5. ADDRESS BUS VALID
6. ADDRESS BUS VALID
7. ADDRESS BUS VALID
8. ADDRESS BUS VALID
9. INSTRUCTION VALID
10. INSTRUCTION VALID
11. DATA INPUT VALID
12. INSTRUCTION VALID

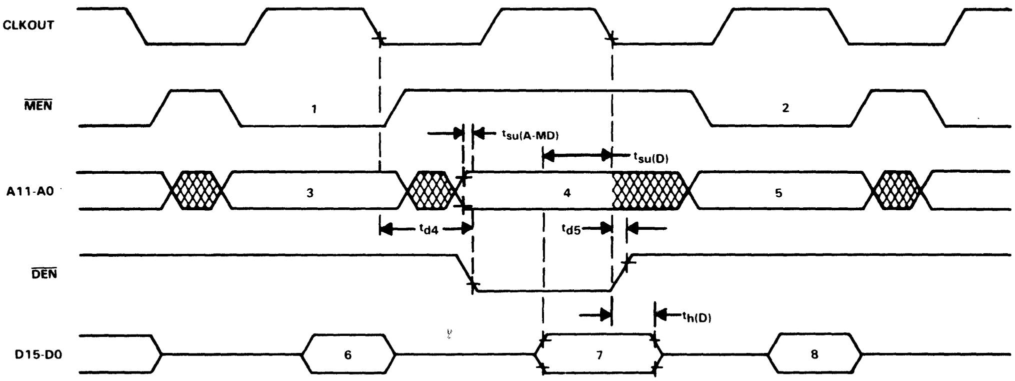
TBLW instruction timing



LEGEND:

1. TBLW INSTRUCTION PREFETCH
2. DUMMY PREFETCH
3. NEXT INSTRUCTION PREFETCH
4. ADDRESS BUS VALID
5. ADDRESS BUS VALID
6. ADDRESS BUS VALID
7. ADDRESS BUS VALID
8. INSTRUCTION VALID
9. INSTRUCTION VALID
10. DATA OUTPUT VALID
11. INSTRUCTION VALID

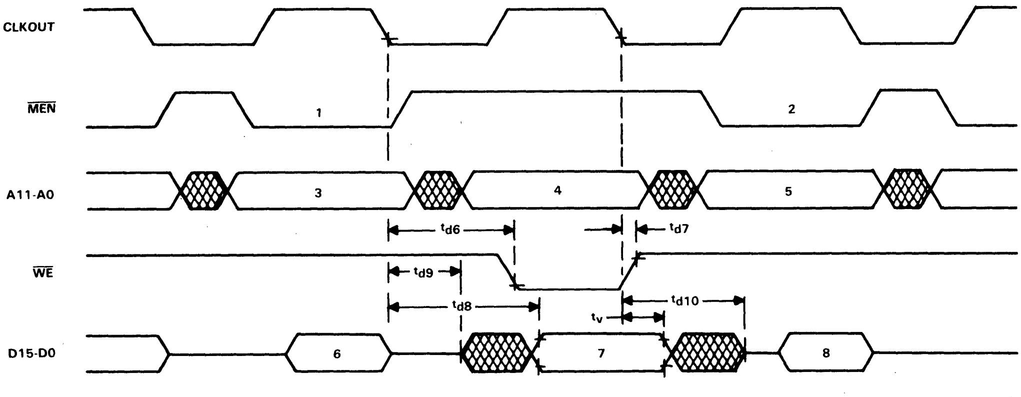
IN instruction timing



LEGEND:

1. IN INSTRUCTION PREFETCH
2. NEXT INSTRUCTION PREFETCH
3. ADDRESS BUS VALID
4. PERIPHERAL ADDRESS VALID
5. ADDRESS BUS VALID
6. INSTRUCTION VALID
7. DATA INPUT VALID
8. INSTRUCTION VALID

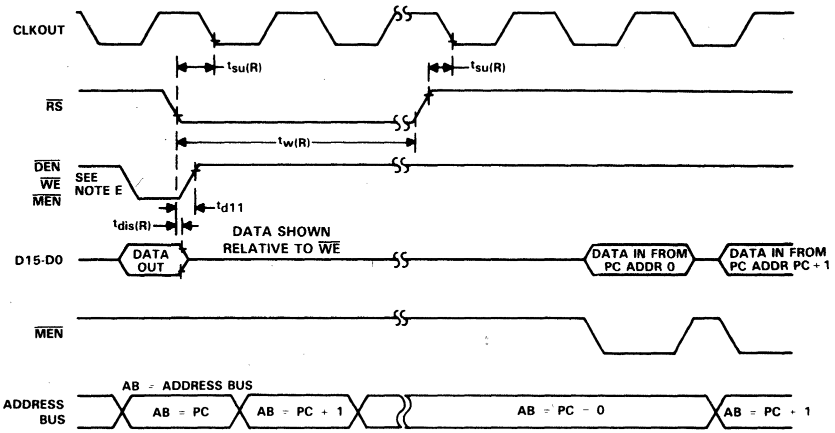
OUT instruction timing



LEGEND:

1. OUT INSTRUCTION PREFETCH
2. NEXT INSTRUCTION PREFETCH
3. ADDRESS BUS VALID
4. PERIPHERAL ADDRESS VALID
5. ADDRESS BUS VALID
6. INSTRUCTION VALID
7. DATA OUTPUT VALID
8. INSTRUCTION VALID

reset timing



NOTES: A. $\overline{RS}$ forces $\overline{DEN}$, $\overline{WE}$, and $\overline{MEN}$ high and places data bus D0 through D15 in a high-impedance state. AB outputs (and program counter) are synchronously cleared to zero after the next complete CLK cycle from $\overline{RS}$↓.

B. $\overline{RS}$ must be maintained for a minimum of five clock cycles.

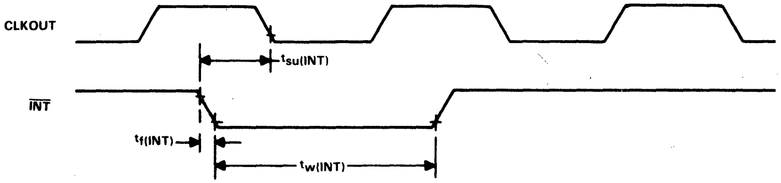
C. Resumption of normal program will commence after one complete CLK cycle from $\overline{RS}$↑.

D. Due to the synchronizing action on $\overline{RS}$, time to execute the function can vary dependent upon when $\overline{RS}$↑ or $\overline{RS}$↓ occur in the CLK cycle.

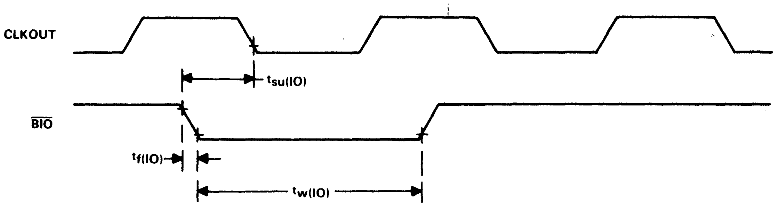
E. Diagram shown is for definition purpose only. $\overline{DEN}$, $\overline{WE}$, and $\overline{MEN}$ are mutually exclusive.

F. During a write cycle, $\overline{RS}$ may produce an invalid write address.

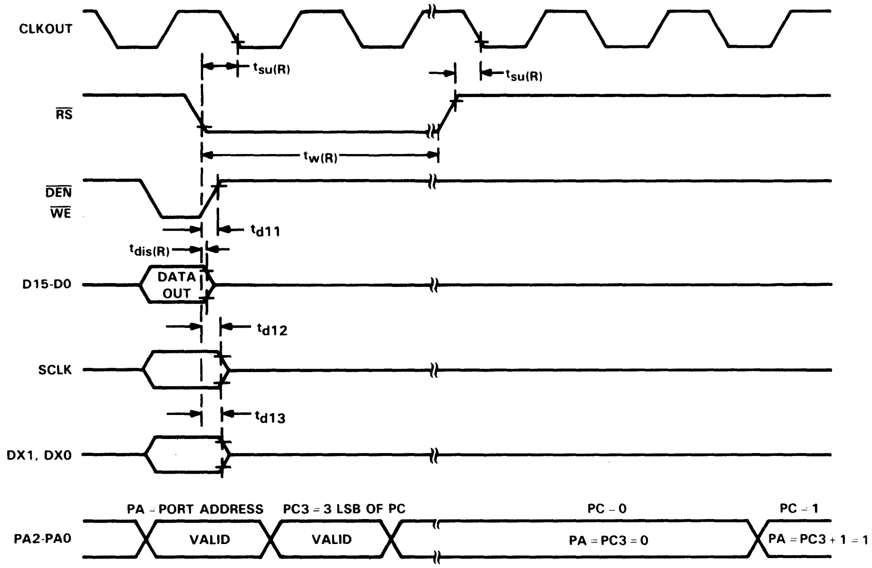
interrupt timing



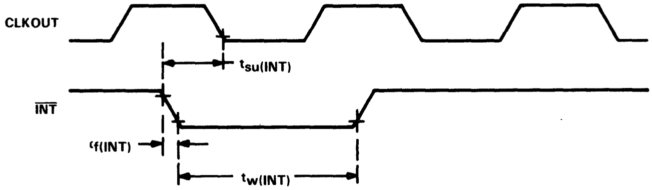
$\overline{BIO}$ timing



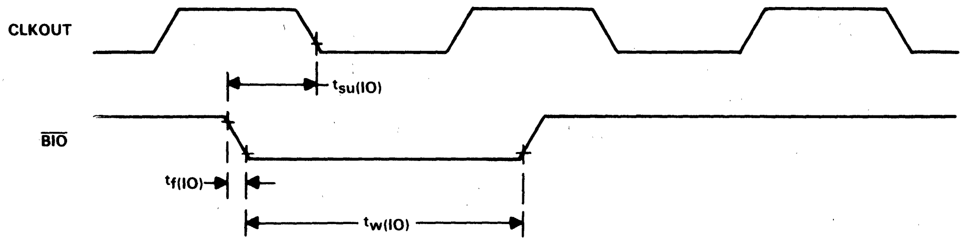
reset timing



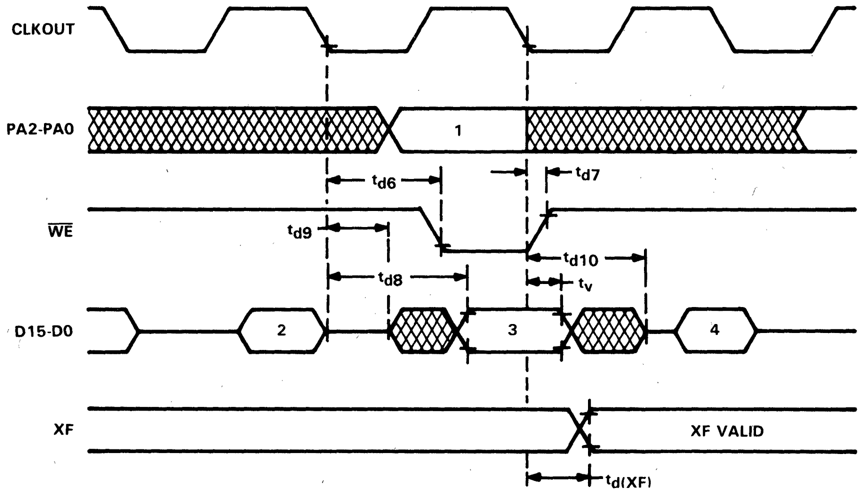
interrupt timing



$\overline{BIO}$ timing



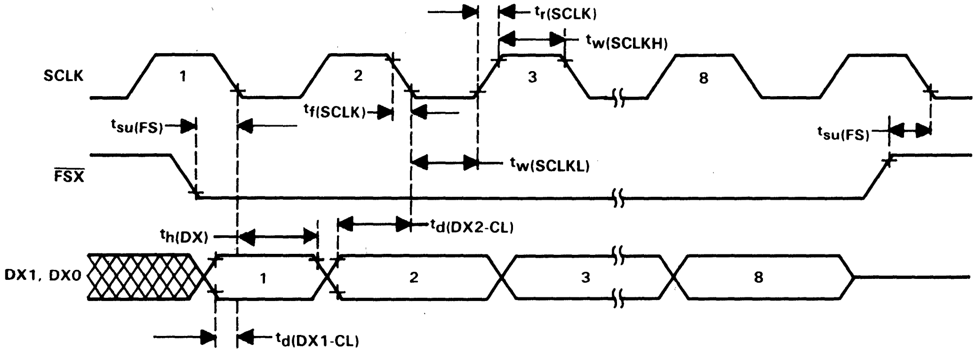
XF timing



LEGEND:

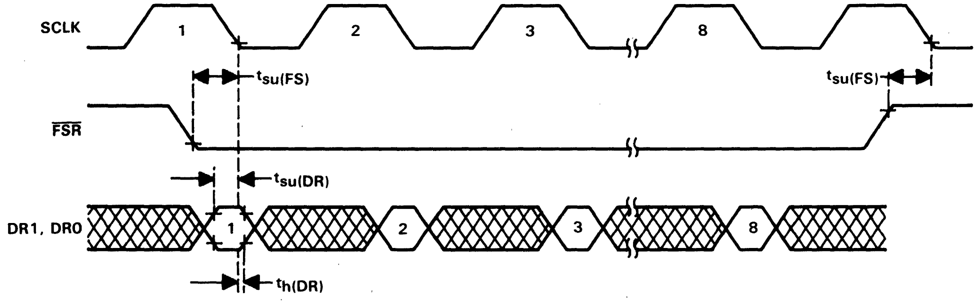
1. PORT ADDRESS VALID
2. OUT OPCODE VALID
3. PORT DATA VALID
4. NEXT INSTRUCTION OPCODE VALID

external framing: transmit timing



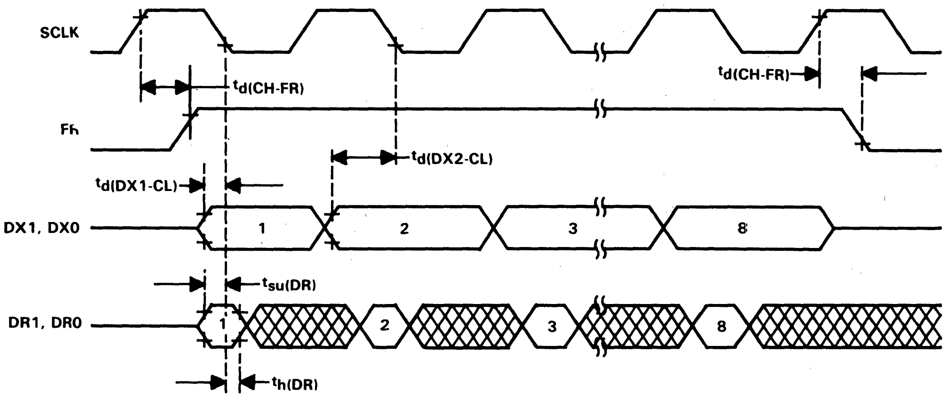
NOTES: G. Data valid on transmit outputs until SCLK rises.
H. The most significant bit is shifted first.

external framing: receive timing



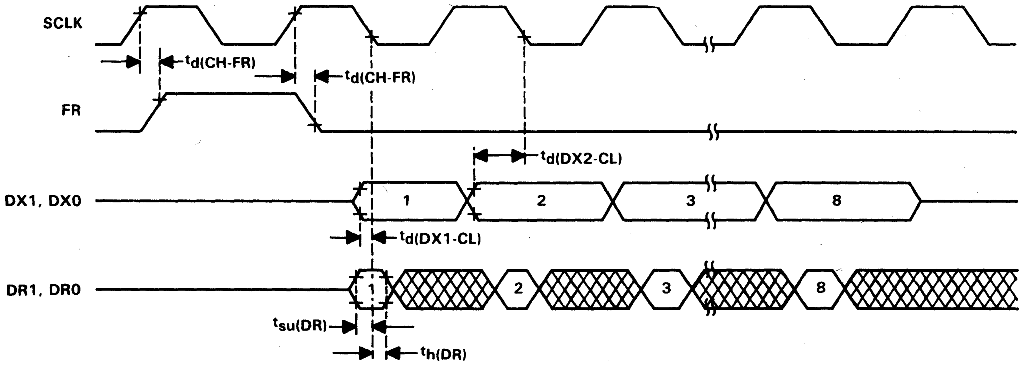
NOTE H: The most significant bit is shifted first.

internal framing: variable-data rate



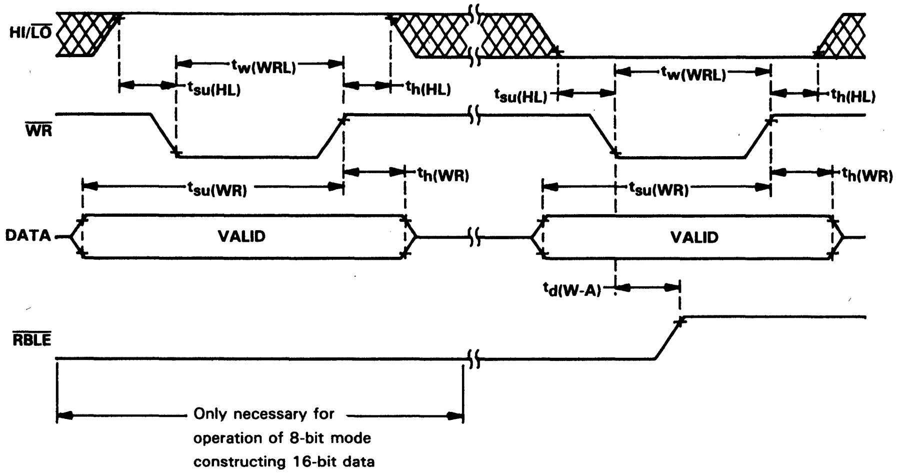
NOTE H: The most significant bit is shifted first.

internal framing: fixed-data rate

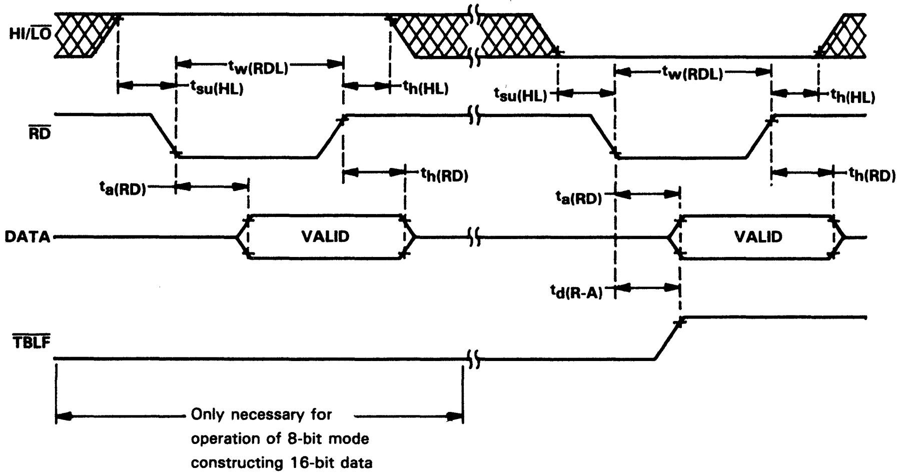


NOTE H: The most significant bit is shifted first.

coprocessor timing: external write to coprocessor port



coprocessor timing: external read from coprocessor port



EPROM PROGRAMMING

absolute maximum ratings over specified temperature range (unless otherwise noted)†

Supply voltage range, V_{PP} (see Note 1) . . . . −0.6 V to 14 V

†Stresses beyond 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 beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1. All voltage values are with respect to GND.

recommended operating conditions

		MIN	NOM	MAX	UNIT
V_{PP}	Supply voltage (see Note 2)		12.5	12.75	V

NOTE 2:. V_{PP} can be connected to V_{CC} directly (except in the program mode). V_{CC} supply current in this case would be I_{CC} + I_{PP}. During programming, V_{PP} must be maintained at 12.5 V (±0.25 V).

electrical characteristics over specified temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
I_{PP1}	V_{PP} supply current	$V_{PP} = V_{CC} = 5.5$ V			100	μA
I_{PP2}	V_{PP} supply current (during program pulse)	V_{PP} = :2.5 V		30	50	mA

†All typical values except for I_{CC} are at V_{CC} = 5 V, T_A = 25°C.

recommended timing requirements for programming, T_A = 25°C, V_{CC} = 6 V, V_{PP} = 12.5 V (see Note 3)

		MIN	NOM	MAX	UNIT
$t_{w(IPGM)}$	Initial program pulse duration	0.95	1	1.05	ms
$t_{w(FPGM)}$	Final pulse duration	3.8		63	ms
$t_{su(A)}$	Address setup time	2			μs
$t_{su(E)}$	$\overline{E}$ setup time	2			μs
$t_{su(G)}$	$\overline{G}$ setup time	2			μs
$t_{dis(G)}$	Output disable time from $\overline{G}$	0		130†	ns
$t_{en(G)}$	Output enable time from $\overline{G}$			150†	ns
$t_{su(D)}$	Data setup time	2			μs
$t_{su(VPP)}$	V_{PP} setup time	2			μs
$t_{su(VCC)}$	VCC setup time	2			μs
$t_{h(A)}$	Address hold time	0			μs
$t_{h(D)}$	Data hold time	2			μs

†Values derived from characterization data and not tested.

NOTES: 3. For all switching characteristics and timing measurements, input pulse levels are 0.40 V to 2.4 V and V_{PP} = 12.5 V ± 0.25 V during programming.

4. Common test conditions apply for $t_{dis(G)}$ except during programming.

PROGRAMMING THE TMS320E15/E17 EPROM CELL

Each TMS320E15/E17 device includes a 4K x 16-bit industry-standard EPROM cell for prototyping, early field testing, and low-volume production. In conjunction with this EPROM, the TMS320C15/C17 with a 4K-word masked ROM, then, provides more migration paths for cost-effective production.

EPROM adaptor sockets are available that provide pin-to-pin conversions for programming any TMS320E15/E17 device. One adaptor socket (part number RTC/PGM320A-06), shown in Figure 7, converts a 40-pin DIP device into an equivalent 28-pin device. Another socket (part number RTC/PGM320C-06), not shown, permits 44- to 28-pin conversion.

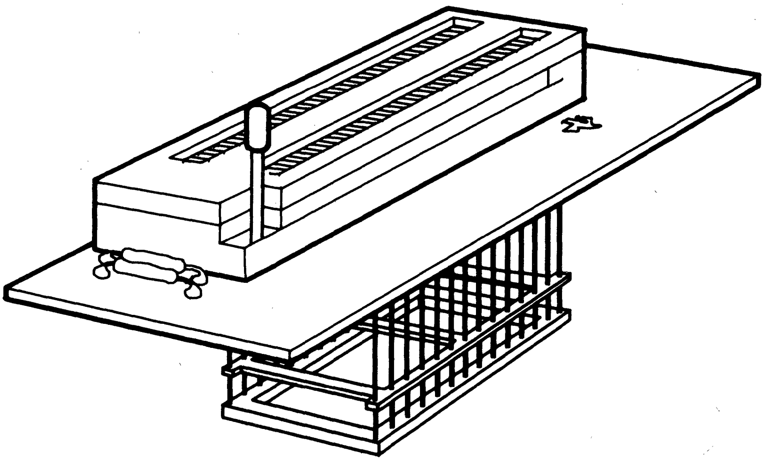
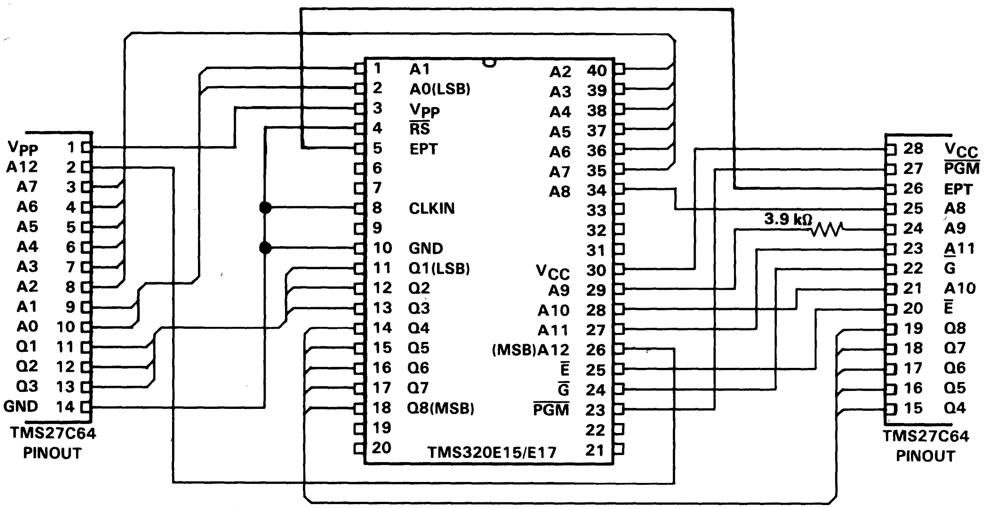


FIGURE 7. EPROM ADAPTOR SOCKET (40-pin to 28-pin DIP Conversion)

Key features of the EPROM cell include the normal programming operation as well as verification. The EPROM cell also includes a code protection feature that allows code to be protected against copyright violations.

The TMS320E15/E17 EPROM cell is programmed using the same family and device codes as the TMS27C64 8K x 8-bit EPROM. The TMS27C64 EPROM series are ultraviolet-light erasable, electrically programmable, read-only memories, fabricated using HVCMOS technology. They are pin-compatible with existing 28-pin ROMs and EPROMs. These EPROMs operate from a single 5-V supply in the read mode; however, a 12.5-V supply is needed for programming. All programming signals are TTL level. For programming outside the system, existing EPROM programmers can be used. Locations may be programmed singly, in blocks, or at random.

Figure 8 shows the wiring conversion to program the TMS320E15/E17 using the 28-pin pinout of the TMS27C64. The table of pin nomenclature provides a description of the TMS27C64 pins. The code to be programmed into the device should be in serial mode. The TMS320E15/E17 uses 13 address lines to address the 4K-word memory in byte format.



CAUTION

Although acceptable by some EPROM programmers, the signature mode cannot be used on any TMS320E1x device. The signagure mode will input a high-level voltage (12.5 Vdc) onto pin A9. Since this pin is not designed for high voltage, the cell will be damaged. To prevent an accidental application of voltage, Texas Instruments has inserted a 3.9 kΩ resistor between pin A9 of the TI programmer socket and the programmer itself.

PIN NOMENCLATURE (TMS320E15/TMS320E17)

NAME	I/O	DEFINITION
A0-A12	I	On-chip EPROM programming address lines
CLKIN	I	Clock oscillator input
E	I	EPROM chip select
EPT	I	EPROM test mode select
G	I	EPROM read/verify select
GND	I	Ground
PGM	I	EPROM write/program select
Q1-Q8	I/O	Data lines for byte-wide programming of on-chip 8K bytes of EPROM
RS	I	Reset for initializing the device
V_{CC}	I	5-V power supply
V_{PP}	I	12.5-V power supply

FIGURE 8. TMS320E15/E17 EPROM PROGRAMMING CONVERSION TO TMS27C64 EPROM PINOUT

Table 8 shows the programming levels required for programming, verifying, reading, and protecting the EPROM cell.

TABLE 8. TMS320E15/E17 PROGRAMMING MODE LEVELS

SIGNAL NAME	TMS320E15 PIN	TMS27C64 PIN	PROGRAM	VERIFY	READ	PROTECT VERIFY	EPROM PROTECT
$\overline{E}$	25	20	V_{IL}	V_{IL}	V_{IL}	V_{IL}	V_{IH}
$\overline{G}$	24	22	V_{IH}	$\overline{PULSE}$	$\overline{PULSE}$	V_{IL}	V_{IH}
$\overline{PGM}$	23	27	$\overline{PULSE}$	V_{IH}	V_{IH}	V_{IH}	V_{IH}
V_{PP}	3	1	V_{PP}	V_{PP}	V_{CC}	$V_{CC} + 1$	V_{PP}
V_{CC}	30	28	V_{CC}	V_{CC}	V_{CC}	$V_{CC} + 1$	$V_{CC} + 1$
V_{SS}	10	14	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
CLKIN	8	14	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
$\overline{RS}$	4	14	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
EPT	5	26	V_{SS}	V_{SS}	V_{SS}	V_{PP}	V_{PP}
Q1-Q8	11-18	11-13, 15-19	D_{IN}	Q_{OUT}	Q_{OUT}	Q8 = RBIT	Q8 = $\overline{PULSE}$
A0-A3	2,1,40,39	10-7	ADDR	ADDR	ADDR	X	X
A4	38	6	ADDR	ADDR	ADDR	X	V_{IH}
A5	37	5	ADDR	ADDR	ADDR	X	X
A6	36	4	ADDR	ADDR	ADDR	V_{IL}	X
A7-A9	35,34,29	3,25,24	ADDR	ADDR	ADDR	X	X
A10-A12	28-26	21,23,2	ADDR	ADDR	ADDR	X	X

LEGEND:

V_{IH} = TTL high level; V_{IL} = TTL low level; ADDR = byte address bit
V_{PP} = 12.5 V ± 0.25 V; V_{CC} = 5 V ± 0.25 V; X = don't care
$\overline{PULSE}$ = low-going TTL level pulse; D_{IN} = byte to be programmed at ADDR
Q_{OUT} = byte stored at ADDR; RBIT = ROM protect bit.

programming

Since every memory bit in the cell is a logic 1, the programming operation reprograms certain bits to 0. Once programmed, these bits can only be erased using ultraviolet light. The correct byte is placed on the data bus with Vpp set to the 12.5-V level. The $\overline{PGM}$ pin is then pulsed low to program in the zeroes.

erasure

Before programming, the device must be erased by exposing it to ultraviolet light. The recommended minimum exposure dose (UV-intensity X exposure-time) is 15 watt-seconds per square centimeter. A typical 12 milliwatt-seconds per square centimeter, filterless UV lamp will erase the device in 21 minutes. The lamp should be located about 2.5 centimeters above the chip during erasure. After exposure, all bits are in the high state.

verify/read

To verify correct programming, the EPROM cell can be read using either the verify or read line definitions shown in Table 8, assuming the inhibit bit has not been programmed.

program inhibit

Programming may be inhibited by maintaining a high level input on the $\overline{E}$ pin or $\overline{PGM}$ pin.

read

The EPROM contents may be read independent of the programming cycle, provided the RBIT (ROM protect bit) has not been programmed. The read is accomplished by setting $\overline{E}$ to zero and pulsing $\overline{G}$ low. The contents of the EPROM location selected by the value on the address inputs appear on Q8-Q1.

output disable

During the EPROM programming process, the EPROM data outputs may be disabled, if desired, by establishing the output disable state. This state is selected by setting $\overline{G}$ and $\overline{E}$ pins high. While output disable is selected, Q8-Q1 are placed in the high-impedance state.

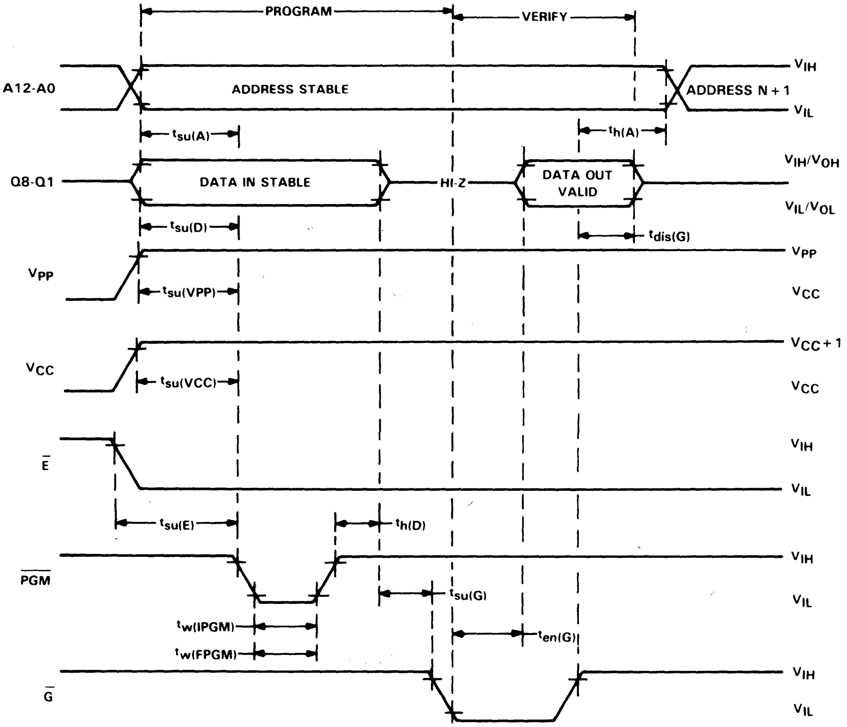
EPROM protection

To protect the proprietary algorithms existing in the code programmed on-chip, the ability to read or verify code from external accesses can be completely disabled. Programming the RBIT disables external access of the EPROM cell, making it impossible to access the code resident in the EPROM cell. The only way to remove this protection is to erase the entire EPROM cell, thus removing the proprietary information. The signal requirements for programming this bit are shown in Table 8. The cell can be determined as protected by verifying the programming of the RBIT shown in the table.

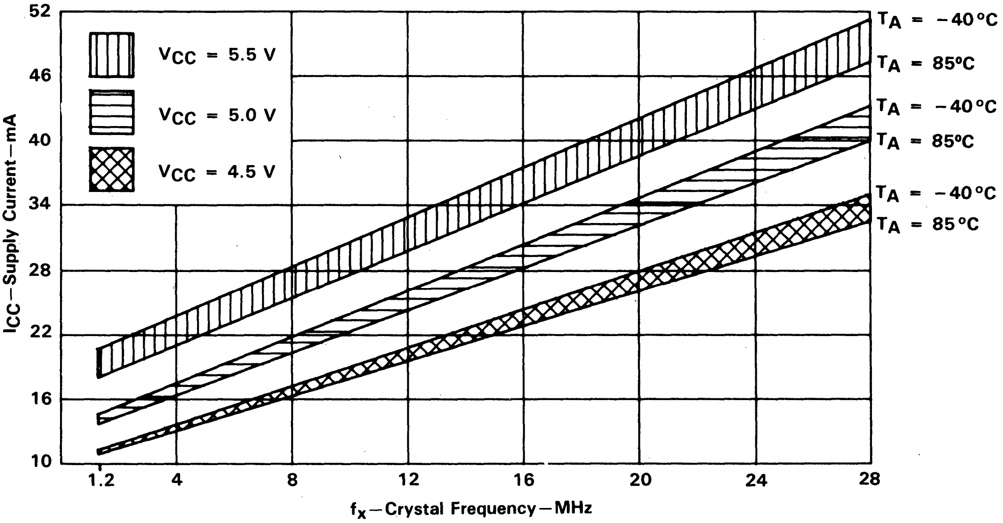
standard programming procedure

Before programming, the device must first be completely erased. Then the device can be programmed with the correct code. It is advisable to program unused sections with zeroes as a further security measure. After the programming is complete, the code programmed into the cell should be verified. If the cell passes verification, the next step is to program the ROM protect bit (RBIT). Once the RBIT programming is verified, an opaque label should be placed over the window to protect the EPROM cell from inadvertent erasure by ambient light. At this point, the programming is complete, and the device is ready to be placed into its destination circuit.

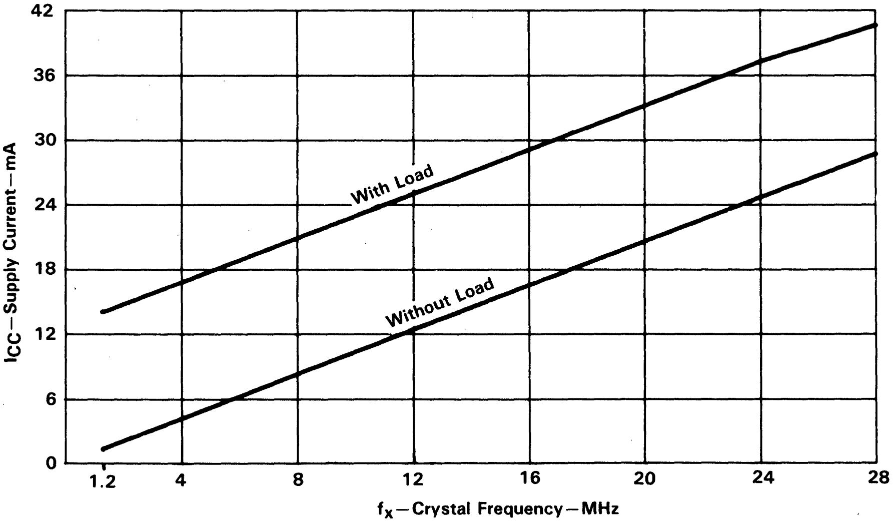
program cycle timing



TYPICAL POWER VS. FREQUENCY GRAPHS



(a)—40°C TO 85°C TEMPERATURE RANGE



(b) VOLTAGE = 5 V; TEMPERATURE = 25°C

FIGURE 9. TYPICAL CMOS I_{CC} VS. FREQUENCY

PACKAGE TYPES

PACKAGE TYPE	SUFFIX	FAMILY MEMBERS	
40-pin plastic DIP (100-mil pin spacing)	N	NMOS: CMOS:	TMS32010 TMS320C10, TMS320C10-14, TMS320C10-25, TMS320C15, TMS320C15-25, TMS320C17
40-pin windowed ceramic DIP (100-mil pin spacing)	JD	CMOS:	TMS320E15, TMS320E15-25, TMS320E17
44-lead PLCC (50-mil pin spacing)	FN	CMOS:	TMS320C10, TMS320C10-25, TMS320C15, TMS320C15-25, TMS320C17
44-lead windowed CER-QUAD (50-mil pin spacing)	FZ	CMOS:	TMS320E15, TMS320E15-25, TMS320E17

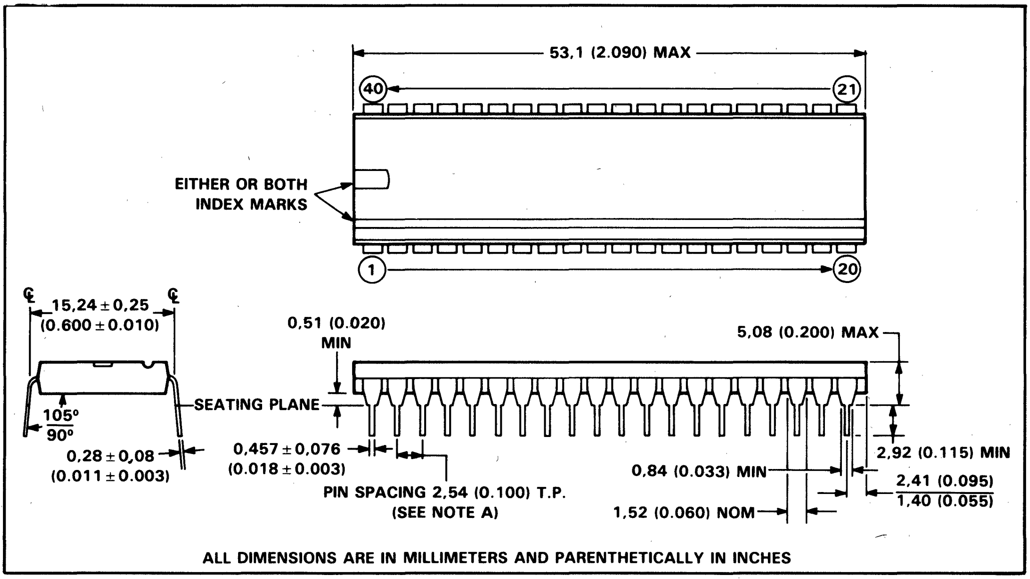
THERMAL DATA

thermal resistance characteristics

PACKAGE	$R_{\theta JA}$ °C/W)	$R_{\theta JC}$ (°C/W)
40-pin plastic dual-in-line package (NMOS)	51.6	16.6
40-pin plastic dual-in-line package (CMOS)	84	26
40-pin windowed ceramic dual-in-line package (CMOS)	40	8
44-lead plastic chip carrier package (CMOS)	60	17
44-lead CER-QUAD chip carrier package (CMOS)	63.8	7.8

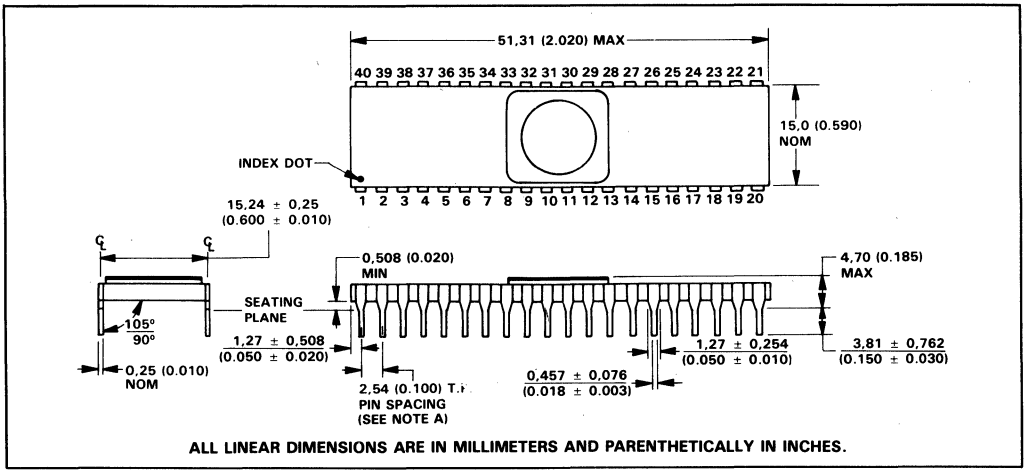
MECHANICAL DATA

40-pin plastic dual-in-line package



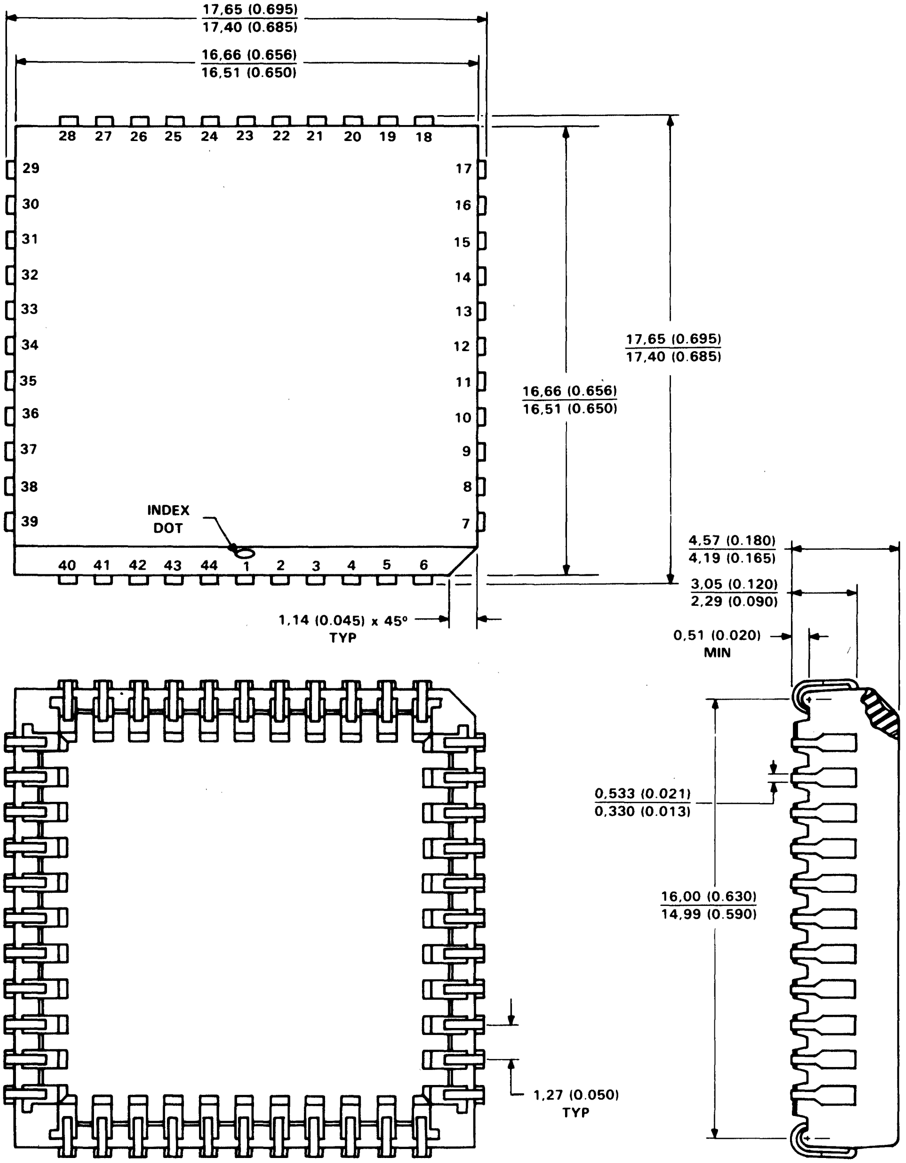
NOTE A: Each pin centerline is located within 0,254 (0.010) of its true longitudinal position.

40-pin windowed ceramic dual-in-line package



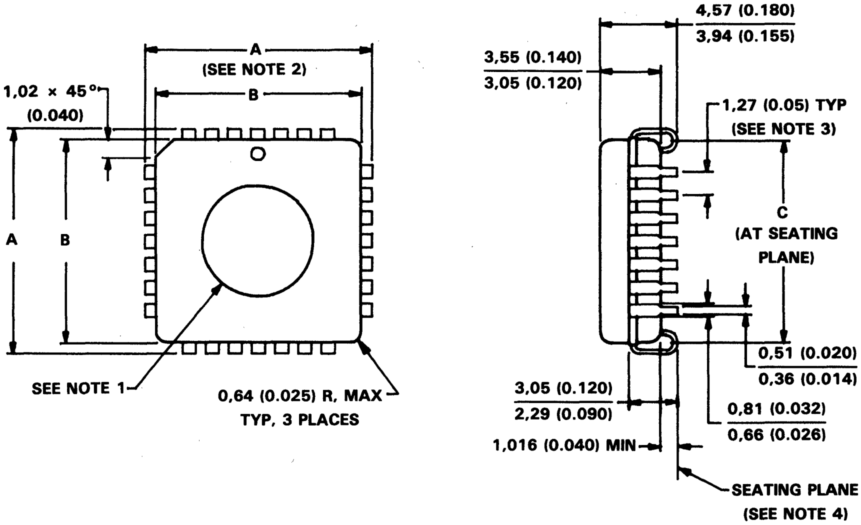
NOTE A: Each pin centerline is located within 0,254 (0.010) of its true longitudinal position.

44-lead plastic chip package



ALL DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES

44-lead FZ CER-QUAD chip carrier package



JEDEC OUTLINE	NO. OF TERMINALS	A		B		C	
		MIN	MAX	MIN	MAX	MIN	MAX
MO-087AA	28	12,32 (0.485)	12,57 (0.495)	10,92 (0.430)	11,56 (0.455)	10,41 (0.410)	10,92 (0.430)
MO-087AB	44	17,40 (0.685)	17,65 (0.695)	16,00 (0.630)	16,64 (0.655)	15,49 (0.610)	16,00 (0.630)
– –	68	25,02 (0.985)	25,27 (0.995)	23,62 (0.930)	24,26 (0.955)	23,11 (0.910)	23,62 (0.930)

ALL LINEAR DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES

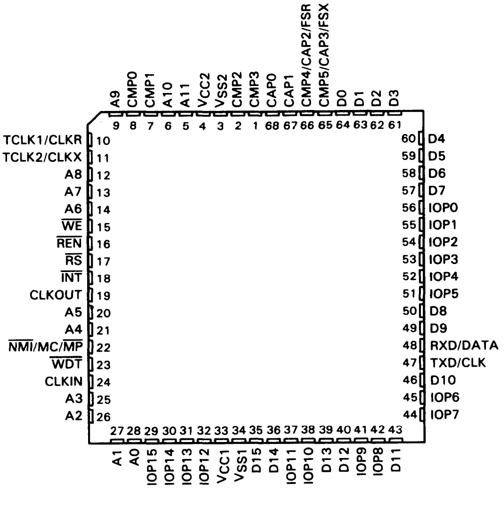
NOTES:
1. Glass is optional, and the diameter is dependent on device application.
2. Centerline of center pin, each side, is within 0,10 (0.004) of package centerline as determined by dimension B.
3. Location of each pin is within 0,127 (0.005) of its true position with respect to center pin on each side.
4. The lead contact points are planar and within 0,15 (0.006).

INDEX

DECEMBER 1988

- 160-ns Instruction Cycle
- 256-Word On-Chip Data RAM
- 4K-Word On-Chip Program ROM (TMS320C14)
- 4K-Word On-Chip Program EPROM (TMS320E14)
- EPROM Code Protection for Copyright Security
- 4K-Word Total External Memory at Full Speed (Microprocessor Mode)
- 32-Bit ALU/Accumulator
- 16 × 16-Bit Multiplier with a 32-Bit Product
- 0 to 16-Bit Barrel Shifter
- Seven Input and Seven Output External Ports
- 16-Bit Bidirectional Data Bus with Greater Than 50-Mbps Transfer Rate
- Bit-Selectable I/O Port (16 Pins)
- Serial Port with Programmable Protocols
- Event Manager with Capture Inputs and Compare Outputs
- Four Independent Timers (Watchdog, General Purpose [2], Serial Port)

TMS320C14, TMS320E14
FN AND FZ PACKAGES
(TOP VIEW)



- Single 5-V Supply
- Packaging: 68-Pin PLCC or CLCC
- 15 Internal/External Interrupts

ADVANCE INFORMATION

introduction

This data sheet provides complete design documentation for the TMS320C14 and TMS320E14 devices, which are a part of the First Generation TMS320 family. The TMS32010, the first digital signal processor of the TMS320 family, was introduced in 1982. Its powerful instruction set, inherent flexibility, high speed number-crunching capabilities, and innovative architecture have made this high performance, cost-effective processor the ideal solution for many commercial, industrial and military applications. Since that time, three generations of the TMS320 family have evolved, each with its own group of related devices. All TMS320 devices combine the flexibility of a high speed controller with the numerical capability of an array processor. This offers an inexpensive alternative to multichip bit-slice processors.

The TMS320C14/E14 devices are 16/32-bit single-chip digital signal processors that are object-code compatible with the TMS32010 device. This allows hardware upgrading without the expense of software re-development. The highly paralleled architecture and efficient instruction set provide the speed and flexibility to execute 6.4 million instructions per second (MIPS). The TMS320C14/E14 devices contain several on-chip peripherals that can reduce and even eliminate interface components and "glue" circuitry, allowing use in space-critical applications.

The TMS320C14/E14 is offered in a 68-pin plastic leaded chip carrier package (FN suffix) rated for operation from 0°C to 70°C (L suffix). It is also offered in a 68-pin ceramic leaded chip carrier package (FZ suffix) carrier rated for operation from 0°C to 70°C (L suffix).

This data sheet is divided into the following major sections: introduction, functional block diagram, architecture, instruction set, development, support products, documentation support, electrical and timing specifications, timing diagrams, and EPROM programming. An index is provided for quick reference to specific information.

ADVANCE INFORMATION documents contain information on new products in the sampling or preproduction phase of development. Characteristic data and other specifications are subject to change without notice.

pin descriptions

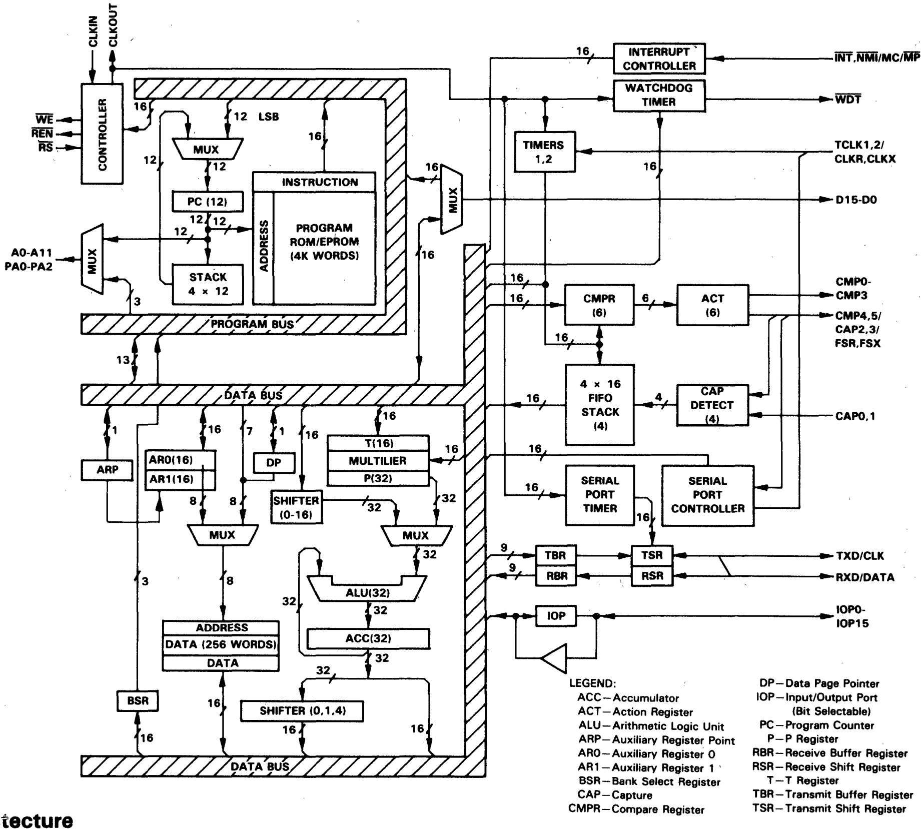
PIN		I/O/Z†	DESCRIPTION
NAME	NO.		ADDRESS/DATA BUSES
A11	5	O/Z	Program memory address bus A11 (MSB) through A0 (LSB) and port addresses PA2 (MSB) through PA0 (LSB). Addresses A11 through A0 are always active and never go to high impedance except during reset. During execution of the IN and OUT instructions, pins 26, 27, and 28 carry the port addresses. Pins A3 through A11 are held high when port accesses are made on pins PA0 through PA2.
A10	6		
A9	9		
A8	12		
A7	13		
A6	14		
A5	20		
A4	21		
A3	25		
A2/PA2	26		
A1/PA1	27		
A0/PA0	28		
D15 MSB	35	I/O/Z	Parallel data bus D15 (MSB) through D0 (LSB). The data bus is always in the high-impedance state except when $\overline{\text{WE}}$ is active (low). The data bus is also active when internal peripherals are written to.
D14	36		
D13	39		
D12	40		
D11	43		
D10	46		
D9	49		
D8	50		
D7	57		
D6	58		
D5	59		
D4	60		
D3	61		
D2	62		
D1	63		
D0 LSB	64		
			INTERRUPT AND MISCELLANEOUS SIGNALS
$\overline{\text{INT}}$	18	I	External interrupt input. The interrupt signal is generated by a low signal on this pin.
$\overline{\text{NMI}}$/MC/$\overline{\text{MP}}$	22	I	Non-maskable interrupt. When this pin is brought low, the device is interrupted irrespective of the state of the INTM bit in status register ST. Microcomputer/microprocessor select. This pin is also sampled when $\overline{\text{RS}}$ is low. If high during reset, internal program memory is selected. If low during reset, external memory will be selected.
$\overline{\text{WE}}$	15	O	Write enable. When active low, $\overline{\text{WE}}$ indicates that device will output data on the bus.
$\overline{\text{REN}}$	16	O	Read enable. When active low, $\overline{\text{REN}}$ indicates that device will accept data from the bus.
$\overline{\text{RS}}$	17	I	Reset. When this pin is low, the device is reset and PC is set to zero.
			SUPPLY/OSCILLATOR SIGNALS
CLKOUT	19	O	System clock output (one fourth CLKIN frequency).
V_{CC}	4,33	I	5-V supply pins.
V_{SS}	3,34	I	Ground pins.
CLKIN	24	I	Master clock input from external clock source.

†Input/Output/High-impedance state.

pin descriptions (concluded)

PIN		I/O	DESCRIPTION
NAME	NO.		SERIAL PORT AND TIMER SIGNALS
RXD/DATA	48	I/O	In the asynchronous and codec modes, this pin is the receive input. In the synchronous mode, this pin is data in while receiving data, and data out while transmitting data.
TXD/CLK	47	I/O	In the asynchronous and codec modes, this pin is the transmit output. In the synchronous mode, this pin is clock input with external clock, and clock output with internal clock.
TCLK1/CLKR	10	I	Timer 1 clock. If external clock is selected, it serves as clock input to Timer 1. Can also be configured as serial port receive clock in codec mode.
TCLK2/CLKX	11	I	Timer 2 clock. If external clock is selected, it serves as clock input to Timer 2. Can also be configured as serial port transmit clock in codec mode.
$\overline{\text{WDT}}$	23	O	Watchdog timer output. An active low is generated on this pin when the watchdog timer times out.
			BIT I/O PINS
IOP15 MSB	29	I/O	16 bit I/O lines that can be individually configured as inputs or outputs and also individually set or reset when configured as outputs.
IOP14	30		
IOP13	31		
IOP12	32		
IOP11	37		
IOP10	38		
IOP9	41		
IOP8	42		
IOP7	44		
IOP6	45		
IOP5	51		
IOP4	52		
IOP3	53		
IOP2	54		
IOP1	55		
IOP0 LSB	56		
			COMPARE AND CAPTURE SIGNALS
CMP0	8	O	Compare outputs. The states of these pins are determined by the combination of compare and action registers.
CMP1	7		
CMP2	2		
CMP3	1		
CAP0	68	I	Capture inputs. A transition on these pins causes the timer register to be captured in FIFO stack.
CAP1	67		
CMP4/CAP2/ FSR	66	I/O	This pin can be configured as compare output, capture input, or as external framing input/output for the receiver section of the serial port in codec mode.
CMP5/CAP3 FSX	65	I/O	This pin can be configured as compare output, capture input, or as external framing input/output for transmit section of the serial port in codec mode.

functional block diagram



architecture

The TMS320 family utilizes a modified Harvard architecture for speed and flexibility. In a strict Harvard architecture, program and data memory lie in two separate spaces, permitting a full overlap of instruction fetch and execution. The TMS320 family's modification of the Harvard architecture allows transfers between program and data spaces, thereby increasing the flexibility of the device. This modification permits coefficients stored in program memory to be read into the RAM, eliminating the need for a separate coefficient ROM. It also makes available immediate instructions and subroutines based on computed values.

32-bit ALU/accumulator

The TMS320C14/E14 devices contain a 32-bit ALU and accumulator for support of double-precision, two's-complement arithmetic. The ALU is a general-purpose arithmetic unit that operates on 16-bit words taken from the data RAM or derived from immediate instructions. In addition to the usual arithmetic instructions, the ALU can perform Boolean operations, providing the bit manipulation ability required of a high-speed controller. The accumulator stores the output from the ALU and is often an input to the ALU. It operates with a 32-bit wordlength. The accumulator is divided into a high-order word (bits 31 through 16) and a low-order word (bits 15 through 0). Instructions are provided for storing the high- and low-order accumulator words in memory.

shifters

Two shifters are available for manipulating data. The ALU barrel shifter performs a left-shift of 0 to 16 places on data memory words loaded into the ALU. This shifter extends the high-order bit of the data word and zero-fills the low-order bits for two's-complement arithmetic. The accumulator parallel shifter performs a left-shift of 0, 1, or 4 places on the entire accumulator and places the resulting high-order accumulator bits into data RAM. Both shifters are useful for scaling and bit extraction.

16 x 16-bit parallel multiplier

The multiplier performs a 16 x 16-bit two's-complement multiplication with a 32-bit result in a single instruction cycle. The multiplier consists of three units: the T Register, P Register, and multiplier array. The 16-bit T Register temporarily stores the multiplicand; the P Register stores the 32-bit product. Multiplier values either come from the data memory or are derived immediately from the MPYK (multiply immediate) instruction word. The fast on-chip multiplier allows the device to perform fundamental operations such as convolution, correlation, and filtering.

data and program memory

Since the TMS320C14/E14 devices use a Harvard architecture, data and program memory reside in two separate spaces. These devices have 256 words of on-chip data RAM and 4K words of on-chip program ROM (TMS320C14) or EPROM (TMS320E14). The EPROM cell utilizes standard PROM programmers and is programed identically to a 64K CMOS EPROM (TMS27C64).

program memory expansion

The first-generation devices are capable of executing up to 4K words of external memory at full speed for those applications requiring external program memory space. This allows for external RAM-based systems to provide multiple functionality.

microcomputer/microprocessor operating modes

The TMS320C14/E14 devices offer two modes of operation defined by the state of the $\overline{\text{NMI}}$/MC/$\overline{\text{MP}}$ pin during reset: the microcomputer mode ($\overline{\text{NMI}}$/MC/$\overline{\text{MP}}$ = 1) or the microprocessor mode ($\overline{\text{NMI}}$/MC/$\overline{\text{MP}}$ = 0). In the microcomputer mode, on-chip ROM is mapped into the memory space with up to 4K words of internal memory available. In the microprocessor mode, all 4K words of memory are external.

interrupts and subroutines

The TMS320C14/E14 devices contain a four-level hardware stack for saving the contents of the program counter during interrupts and subroutine calls. Instructions are available for saving the complete context of the device. PUSH and POP instructions permit a level of nesting restricted only by the amount of available RAM. The TMS320C14/E14 has a total of 16 internal/external interrupts. Fifteen of these are maskable; $\overline{\text{NMI}}$ is the sixteenth.

input/output

The 16-bit parallel data bus can be utilized to access external peripherals. Only the lower three address lines are active, however. The upper nine address lines are driven high.

bit I/O

The TMS320C14/E14 has 16 pins of bit I/O that can be individually configured as inputs or outputs. Each of the pins can be set or cleared without affecting the others. The input pins can also detect and match patterns and generate a maskable interrupt signal to the CPU.

serial port

The TMS320C14/E14 includes an I/O mapped serial port that can operate in one of three modes: asynchronous, synchronous, and codec. Two types of inter-processor communication protocols are supported in all modes. An associated timer provides baud rate/clock generation if required. Depending on the mode, internal/external clock (master/slave) options are available. All communication parameters are software-controlled through a serial control register.

event manager

An event manager is included that provides up to four capture inputs and up to six compare outputs. This peripheral operates with the timers to provide a form of programmable event logging/detection. The six compare outputs can also be configured to produce six channels of high precision PWM.

timers 1 and 2

Two identical 16-bit timers are provided for general purpose applications. Both timers include a 16-bit period register and buffer latch, and can generate a maskable interrupt.

serial port timer

The serial port timer is a 16-bit timer primarily intended for baud rate generation for the serial port. Its architecture is the same as timers 1 and 2, therefore it can serve as a general purpose timer if not needed for serial communication.

watchdog timer

The TMS320C14/E14 contains a 16-bit watchdog timer that can produce a timeout ($\overline{\text{WDT}}$) signal for various applications such as software development and event monitoring. The watchdog timer also generates, at the point of the timeout, a maskable interrupt signal to the CPU.

instruction set

A comprehensive instruction set supports both numeric-intensive operations, such as signal processing, and general-purpose operations, such as high-speed control. All of the first-generation devices are object-code compatible and use the same 60 instructions. The instruction set consists primarily of single-cycle single-word instructions, permitting execution rates of more than six million instructions per second. Only infrequently used branch and I/O instructions are multicycle. Instructions that shift data as part of an arithmetic operation execute in a single cycle and are useful for scaling data in parallel with other operations.

NOTE

The $\overline{\text{BIO}}$ pin on other TMS320C1x devices are not available for use in the TMS320C14/E14. An attempt to execute the BIOZ (Branch on $\overline{\text{BIO}}$ low) instruction will result in a two cycle NOP action.

Three main addressing modes are available with the instruction set: direct, indirect, and immediate addressing.

direct addressing

In direct addressing, seven bits of the instruction word concatenated with the 1-bit data page pointer form the data memory address. This implements a paging scheme in which each page contains 128 words.

indirect addressing

Indirect addressing forms the data memory address from the least-significant eight bits of one of the two auxiliary registers, AR0 and AR1. The Auxiliary Register Pointer (ARP) selects the current auxiliary register. The auxiliary registers can be automatically incremented or decremented and the ARP changed in parallel with the execution of any indirect instruction to permit single-cycle manipulation of data tables. Indirect addressing can be used with all instructions requiring data operands, except for the immediate operand instructions.

immediate addressing

Immediate instructions derive data from part of the instruction word rather than from the data RAM. Some useful immediate instructions are multiply immediate (MPYK), load accumulator immediate (LACK), and load auxiliary register immediate (LARK).

instruction set summary

Table 1 lists the symbols and abbreviations used in Table 2, the instruction set summary. Table 2 contains a short description and the opcode for each TMS320 first-generation instruction. The summary is arranged according to function and alphabetized within each functional group.

TABLE 1. INSTRUCTION SYMBOLS

SYMBOL	MEANING
ACC	Accumulator
D	Data memory address field
I	Addressing mode bit
K	Immediate operand field
PA	3-bit port address field
R	1-bit operand field specifying auxiliary register
S	4-bit left-shift code
X	3-bit accumulator left-shift field

TABLE 2. TMS320 FIRST-GENERATION INSTRUCTION SET SUMMARY

ACCUMULATOR INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABS	Absolute value of accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0
ADD	Add to accumulator with shift	1	1	0	0	0	0	←S→				I	←D→						
ADDH	Add to high-order accumulator bits	1	1	0	1	1	0	0	0	0	0	I	←D→						
ADDS	Add to accumulator with no sign extension	1	1	0	1	1	0	0	0	0	1	I	←D→						
AND	AND with accumulator	1	1	0	1	1	1	1	0	0	1	I	←D→						
LAC	Load accumulator with shift	1	1	0	0	1	0	←S→				I	←D→						
LACK	Load accumulator immediate	1	1	0	1	1	1	1	1	1	0	←K→							
OR	OR with accumulator	1	1	0	1	1	1	1	0	1	0	I	←D→						
SACH	Store high-order accumulator bits with shift	1	1	0	1	0	1	1	←X→			I	←D→						
SACL	Store low-order accumulator bits	1	1	0	1	0	1	0	0	0	0	I	←D→						
SUB	Subtract from accumulator with shift	1	1	0	0	0	1	←S→				I	←D→						
SUBC	Conditional subtract (for divide)	1	1	0	1	1	0	0	1	0	0	I	←D→						
SUBH	Subtract from high-order accumulator bits	1	1	0	1	1	0	0	0	1	0	I	←D→						
SUBS	Subtract from accumulator with no sign extension	1	1	0	1	1	0	0	0	1	1	I	←D→						
XOR	Exclusive OR with accumulator	1	1	0	1	1	1	1	0	0	0	I	←D→						
ZAC	Zero accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1
ZALH	Zero accumulator and load high-order bits	1	1	0	1	1	0	0	1	0	1	I	←D→						
ZALS	Zero accumulator and load low-order bits with no sign extension	1	1	0	1	1	0	0	1	1	0	I	←D→						

AUXILIARY REGISTER AND DATA PAGE POINTER INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LAR	Load auxiliary register	1	1	0	0	1	1	1	0	0	R	I	←D→						
LARK	Load auxiliary register immediate	1	1	0	1	1	1	0	0	0	R	←K→							
LARP	Load auxiliary register pointer immediate	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	K
LDP	Load data memory page pointer	1	1	0	1	1	0	1	1	1	1	I	←D→						
LDPK	Load data memory page pointer immediate	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	K
MAR	Modify auxiliary register and pointer	1	1	0	1	1	0	1	0	0	0	I	←D→						
SAR	Store auxiliary register	1	1	0	0	1	1	0	0	0	R	I	←D→						

TABLE 2. TMS320 FIRST-GENERATION INSTRUCTION SET SUMMARY (continued)

BRANCH INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B	Branch unconditionally	2	2	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BANZ	Branch on auxiliary register not zero	2	2	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BGEZ	Branch if accumulator ≥ 0	2	2	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BGZ	Branch if accumulator > 0	2	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BLEZ	Branch if accumulator ≤ 0	2	2	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BLZ	Branch if accumulator < 0	2	2	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BNZ	Branch if accumulator ≠ 0	2	2	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BV	Branch on overflow	2	2	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BZ	Branch if accumulator - 0	2	2	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
CALA	Call subroutine from accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0
CALL	Call subroutine immediately	2	2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
RET	Return from subroutine or interrupt routine	2	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	1

T REGISTER, P REGISTER, AND MULTIPLY INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APAC	Add P register to accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1
LT	Load T register	1	1	0	1	1	0	1	0	1	0	I	← D →						
LTA	LTA combines LT and APAC into one instruction	1	1	0	1	1	0	1	1	0	0	I	← D →						
LTD	LTD combines LT, APAC, and DMOV into one instruction	1	1	0	1	1	0	1	0	1	1	I	← D →						
MPY	Multiply with T register, store product in P register	1	1	0	1	1	0	1	1	0	1	I	← D →						
MPYK	Multiply T register with immediate operand; store product in P register	1	1	1	0	0	← K →												
PAC	Load accumulator from P register	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	0
SPAC	Subtract P register from accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0

TABLE 2. TMS320 FIRST-GENERATION INSTRUCTION SET SUMMARY (concluded)

CONTROL INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DINT	Disable interrupt	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1
EINT	Enable interrupt	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0
LST	Load status register	1	1	0	1	1	1	1	0	1	1	I	◄——D——►						
NOP	No operation	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
POP	POP stack to accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1
PUSH	PUSH stack from accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0
ROVM	Reset overflow mode	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	0
SOVM	Set overflow mode	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1
SST	Store status register	1	1	0	1	1	1	1	1	0	0	I	◄——D——►						

I/O AND DATA MEMORY OPERATIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMOV	Copy contents of data memory location into next higher location	1	1	0	1	1	0	1	0	0	1	I	◄——D——►						
IN	Input data from port	2	1	0	1	0	0	0	◄PA►			I	◄——D——►						
OUT	Output data to port	2	1	0	1	0	0	1	◄PA►			I	◄——D——►						
TBLR	Table read from program memory to data RAM	3	1	0	1	1	0	0	1	1	1	I	◄——D——►						
TBLW	Table write from data RAM to program memory	3	1	0	1	1	1	1	1	0	1	I	◄——D——►						

development support products

Texas Instruments offers an extensive line of development support products to assist the user in all aspects of TMS320 first-generation-based design and development. These products range from development and application software to complete hardware development and evaluation systems such as the XDS/22. Table 3 lists the software and hardware support products for the first-generation TMS320 devices.

TABLE 3. TMS320C14 SOFTWARE AND HARDWARE SUPPORT

SOFTWARE TOOLS	PART NUMBER
Macro Assembler/Linker	
VAX VMS†	TMDS3242250-08
IBM PC MS-DOS‡§	TMDS3242850-02
VAX ULTRIX†	TMDS3242260-08
SUN-3 UNIX#¶	TMDS3242550-08
CPU Simulator	
VAX VMS†	TMDS3240211-08
IBM PC MS-DOS‡§	TMDS3240811-02
Digital Filter Design Package (DFDP)	
IBM PC MS-DOS‡§	DFDP-IBM002
DSP Software Library	
VAX VMS†	TMDC3240212-18
IBM PC MS-DOS‡§	TMDC3240812-12
HARDWARE TOOLS	**PART NUMBER**
Analog Interface Board (AIB2)	RTC/EVM320C-06
AIB2 Adapter Board	RTC/ADPC14A-06
XDS/22 Emulator	TMDS3262214
EPROM Programmer Adapter Socket	TMDX3270110
TMS320 Design Kit	TMS320DDK

†VAX, VMS, and ULTRIX are trademarks of Digital Equipment Corporation.
‡MS-DOS is a trademark of Microsoft, Incorporated.
§IBM PC is a trademark of IBM Corporation.
¶UNIX is a trademark of AT&T Bell Laboratories.
#SUN is a trademark of Sun Microsystems, Incorporated.

System development begins with the use of the Emulator (XDS). This hardware tool allows the designer to evaluate the processor's performance, benchmark time-critical code, and determine the feasibility of using a TMS320 device to implement a specific algorithm.

Software and hardware can be developed in parallel by using the macro assembler/linker and simulator for software development and the XDS for hardware development. The assembler/linker translates the system's assembly source program into an object module that can be executed by the CPU simulator or XDS. The XDS provides realtime in-circuit emulation and is a powerful tool for debugging and integrating software and hardware modules.

Additional support for the TMS320 products consists of extensive documentation and three-day DSP design workshops offered by the TI Regional Technology Centers (RTCs). The workshops provide hands-on experience with the TMS320 development tools. Refer to the *TMS320 Family Development Support Reference Guide* (SPRU011) for further information about TMS320 development support products and DSP workshops. When technical questions arise regarding the TMS320, contact the Texas Instruments TMS320 DSP Hotline, (713) 274-2320.

documentation support

Extensive documentation supports the first-generation TMS320 devices from product announcement through applications development. The types of documentation include data sheets with design specifications, complete user's guides, and 750 pages of application reports published in the book *Digital Signal Processing Applications with the TMS320 Family* (SPRA012A).

A series of DSP textbooks is being published by both Prentice Hall and John Wiley and Sons to support digital signal processing research and education. Prentice Hall (201) 767-5937 offers among others: *Practical Approaches to Speech Coding*, and *A DSP Laboratory Using the TMS32010*. John Wiley and Sons (800) 526-5368 has published such books as *Digital Filter Design, DFT/FFT and Convolution Algorithms,* and *A Practical Guide to Adaptive Filter Design*. The TMS320 newsletter, *Details on Signal Processing,* is published quarterly and distributed to update TMS320 customers on product information. The TMS320 DSP bulletin board service provides access to large amounts of information pertaining to the TMS320 family.

Refer to the *TMS320 Family Development Support Reference Guide* for further information about TMS320 documentation. To receive copies of first-generation TMS320 literature, call the Customer Response Center at 1-800-232-3200.

ADVANCE INFORMATION

electrical specifications

This section contains all the electrical specifications for the TMS320C14/E14 devices, including test parameter measurement information. Parameters with pp subscript apply only to TMS320E14 in EPROM programming mode.

absolute maximum ratings over specified temperature range (unless otherwise noted)†

Supply voltage range, V_{CC}‡	−0.3 V to 7 V
Supply voltage range, V_{PP}‡	−0.6 V to 14 V
Input voltage range	−0.3 V to 14 V
Output voltage range	−0.3 V to 7 V
Continuous power dissipation	0.5 W
Air temperature range above operating device: L version	0 °C to 70 °C
Storage temperature range	−55 °C to +150 °C

† Stresses beyond 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 beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

‡ All voltage values are with respect to V_{SS}.

recommended operating conditions

		MIN	NOM	MAX	UNIT
V_{CC} Supply voltage	EPROM devices	4.75	5	5.25	V
	EPROM devices while Fast programming	5.75	6.0	6.25	
	EPROM devices while SNAP! programming	6.25	6.5	6.75	
	All other devices	4.5	5	5.5	
V_{PP} Supply voltage for Fast programming (see Note 1)		12.25	12.5	12.75	V
V_{PP} Supply voltage for SNAP! programming (see Note 1)		12.75	13.0	13.25	V
V_{SS} Supply voltage			0		V
V_{IH} High-level input voltage	CLKIN	3			V
	CLKIN, CAP0, CAP1, CMP4/CAP2/FSR, CMP5/CAP3/FSX, $\overline{RS}$		4		
	All remaining inputs	2			
V_{IL} Low-level input voltage, all inputs except as noted				0.8	V
V_{IL} CAP0, CAP1, CMP4/CAP2/FSR, CMP5/CAP3/FSX, $\overline{RS}$			1		V
I_{OH} High-level output current, all outputs				−300	μA
I_{OL} Low-level output current, all outputs				2	mA
T_A Operating free-air temperature, L version		0		70	°C

NOTE 1: V_{PP} can be connected directly (except in the program mode). V_{CC} supply current in this case would be I_{CC} + I_{PP}.

electrical characteristics over specified temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{OH} High-level output voltage		I_{OH} = MAX		2.4	3		V
		I_{OH} = 20 μA (see Note 2)		$V_{CC}-0.4$‡			V
V_{OL} Low-level output voltage		I_{OL} = MAX			0.3	0.5	V
I_{OZ} Off-state output current		V_{CC} = MAX	V_O = 2.4 V			20	μA
			V_O = 0.4 V			−20	
I_I Input current		$V_I = V_{SS}$ to V_{CC}	All inputs except CLKIN			±20	μA
			CLKIN			±50	
I_{CC}§ Supply current	EPROM	f = 25.6 MHz, V_{CC} = 5.25 V, T_A = 0°C to 70°C			65		mA
	ROM	f = 25.6 MHz, V_{CC} = 5.25 V, T_A = 0°C to 70°C			55		mA
I_{PP1} Vpp supply current		$V_{PP} = V_{CC}$ = 5.5 V				100	μA
I_{PP2} Vpp supply current (during program pulse)		V_{PP} = 13 V			30	50	mA
C_i Input capacitance	Data bus	f = 1 MHz, All other pins 0 V			25‡		pF
	All others				15‡		
C_o Output capacitance	Data bus				25‡		pF
	All others				10‡		

†All typical values are at V_{CC} = 5 V, T_A = 25°C, except I_{CC} at 70°C.
‡Values derived from characterization data and not tested.
§I_{CC} characteristics are inversely proportional to temperature.
NOTE 2: This voltage specification is included for interface to HC logic. However, note that all of the other timing parameters defined in this data sheet are specified for TTL logic levels and will differ for HC logic levels.

ADVANCE INFORMATION

PARAMETER MEASUREMENT INFORMATION

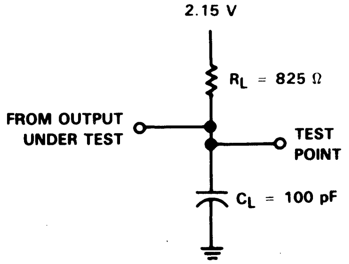


FIGURE 1. TEST LOAD CIRCUIT

EXTERNAL CLOCK REQUIREMENTS

The TMS320C14/E14 uses an external frequency source for a clock. This source is applied to the CLKIN pin, and must conform to the specifications in the table below.

PARAMETER	TEST CONDITIONS	MIN	NOM	MAX	UNIT
CLKIN input clock frequency	T_A = 0°C to 70°C	6.7		25.6	MHz

CLOCK TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	NOM	MAX	UNIT
$t_{c(C)}$	CLKOUT cycle time[‡]	R_L = 825 Ω, C_L = 100 pF, See Figure 1.	156.25	160	597	ns
$t_{r(C)}$	CLKOUT rise time			10[†]		ns
$t_{f(C)}$	CLKOUT fall time			8[†]		ns
$t_{w(CL)}$	Pulse duration, CLKOUT low			72[†]		ns
$t_{w(CH)}$	Pulse duration, CLKOUT high			70[†]		ns
$t_{d(MCC)}$	Delay time CLKIN↑ to CLKOUT↓			45[†]		ns

[†] Values derived from characterization data and not tested.

[‡] $t_{c(C)}$ is the cycle time of CLKOUT, i.e., 4 × $t_{c(MC)}$ (4 times CLKIN cycle time if an external oscillator is used).

timing requirements over recommended operating conditions

		MIN	NOM	MAX	UNIT
$t_{c(MC)}$	Master clock cycle time	39.06	40	150	ns
$t_{r(MC)}$	Rise time master clock input		5[†]	10[†]	ns
$t_{f(MC)}$	Fall time master clock input		5[†]	10[†]	ns
$t_{w(MCP)}$	Pulse duration master clock	$0.45t_{c(MC)}$[†]		$0.55t_{c(MC)}$[†]	ns
$t_{w(MCL)}$	Pulse duration master clock low		15[†]		ns
$t_{w(MCH)}$	Pulse duration master clock high		15[†]		ns

[†] Values derived from characterization data and not tested.

MEMORY READ AND INSTRUCTION TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d1}	Delay time CLKOUT↓ to address bus valid	R_L = 825 Ω, C_L = 100 pF, See Figure 1.	10[†]		40	ns
t_{d2}	Delay time CLKOUT↓ to $\overline{\text{REN}}$↓ (memory access)		$0.25t_{c(C)} - 5$[†]		$0.25t_{c(C)} + 12$	ns
t_{d3}	Delay time CLKOUT↓ to $\overline{\text{REN}}$↑ (memory access)		−10[†]		12	ns
t_{d4}	Delay time CLKOUT↓ to $\overline{\text{REN}}$↓ (I/O access)		$0.25t_{c(C)} - 5$[†]		$0.25t_{c(C)} + 12$	ns
t_{d5}	Delay time CLKOUT↓ to $\overline{\text{REN}}$↑ (I/O access)		−10[†]		12	ns
t_{d6}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↓		$0.5t_{c(C)} - 5$[†]		$0.5t_{c(C)} + 12$	ns
t_{d7}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↑		−10[†]		12	ns
t_{d8}	Delay time CLKOUT↓ to data bus OUT valid				$0.25t_{c(C)} + 52$	ns
t_{d9}	Time after CLKOUT↓ that data bus starts to be driven		$0.25t_{c(C)} - 5$[†]			ns
t_{d10}	Time after CLKOUT↓ that data bus stops being driven				$0.25t_{c(C)} + 30$[†]	ns
t_v	Data bus OUT valid after CLKOUT↓		$0.25t_{c(C)} - 10$			ns
$t_{h(A-WR)}$	Address hold time after $\overline{\text{WE}}$↑, $\overline{\text{REN}}$↑		0[†]			ns
$t_{su(A-REN)}$	Address bus setup time prior to $\overline{\text{REN}}$↓		$0.25t_{c(C)} - 35$			ns

[†] Values derived from characterization data and not tested.

timing requirements over recommended operating conditions

		TEST CONDITIONS	MIN	NOM	MAX	UNIT
$t_{su(D)}$	Setup time data bus valid prior to CLKOUT↓	R_L = 825 Ω, C_L = 100 pF, See Figure 1.	40			ns
$t_{h(D)}$	Hold time data bus held valid after CLKOUT↓ (see Note 3)		0			ns

NOTE 3: Data may be removed from the data bus upon $\overline{\text{REN}}$↑ preceding CLKOUT↓.

RESET ($\overline{RS}$) TIMING

switching characteristics over recommended operating conditions

PARAMETER	TEST CONDITIONS	MIN TYP MAX	UNIT
t_{d11} Delay time $\overline{WE}$↑, and $\overline{REN}$↑ from $\overline{RS}$	R_L = 825 Ω, C_L = 100 pF, See Figure 1.	$0.5t_{c(C)}+50$†	ns
$t_{dis(R)}$ Data bus disable time after $\overline{RS}$		$0.25t_{c(C)}+50$†	ns
$t_{dis(A)}$ Address bus disable time after $\overline{RS}$ low		$0.25t_{c(C)}+50$†	ns
$t_{en(A)}$ Address bus enable time after $\overline{RS}$ high		$0.25t_{c(C)}+50$†	ns

†These values were derived from characterization data and not tested.

timing requirements over recommended operating conditions

	MIN	NOM	MAX	UNIT
$t_{su(R)}$ Reset ($\overline{RS}$) setup time prior to CLKOUT (see Note 4)	40			ns
$t_{w(R)}$ $\overline{RS}$ pulse duration	$5t_{c(C)}$			ns

NOTE 4: $\overline{RS}$ can occur anytime during a clock cycle. Time given is minimum to ensure synchronous operation.

MICROCOMPUTER/MICROPROCESSOR MODE ($\overline{NMI}$/MC/$\overline{MP}$)

timing requirements over recommended operating conditions

	MIN	NOM	MAX	UNIT
$t_{h(MC/MP)}$‡ Hold time after RS high	$1.25t_c$			ns

‡Hold time to put device in microprocessor mode.

INTERRUPT ($\overline{INT}$)/NON-MASKABLE INTERRUPT ($\overline{NMI}$)

timing requirements over recommended operating conditions (see Note 5)

	MIN	NOM	MAX	UNIT
$t_{f(INT)}$ Fall time $\overline{INT}$			15†	ns
$t_{f(NMI)}$ Fall time $\overline{NMI}$			15†	ns
$t_{w(INT)}$ Pulse duration $\overline{INT}$	$t_{c(C)}$			ns
$t_{w(NMI)}$ Pulse duration $\overline{NMI}$	$t_{c(C)}$			ns
$t_{su(INT)}$ Setup time $\overline{INT}$ before CLKOUT low	40			ns
$t_{su(NMI)}$ Setup time $\overline{NMI}$ before CLKOUT low	40			ns

†These values were derived from characterization data and not tested.
NOTE 5: $\overline{INT}$ and $\overline{NMI}$ are synchronous inputs and can occur at any time during the cycle. $\overline{NMI}$ and $\overline{INT}$ are edge triggered only.

ADVANCE INFORMATION

ADVANCE INFORMATION

BIT I/O TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{rfP}	Rise and fall time outputs	R_L = 825 Ω, C_L = 100 pF, See Figure 1.			20†	ns
$t_{d(IOP)}$	CLKOUT low to data valid outputs				$.25t_{c(C)}+20$	ns

timing requirements over recommended operating conditions

		TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{rfI(IOP)}$	Rise and fall time inputs	R_L = 825 Ω, C_L = 100 pF, See Figure 1.			20†	ns
$t_{su(IOP)}$	Data setup time before CLKOUT time		20†			ns
$t_{wI(IOP)}$	Input pulse duration		$t_{c(C)}+20$			ns

†These values were derived from characterization data and not tested.

GENERAL PURPOSE TIMERS

timing requirements over recommended operating conditions

		TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{r(TIM)}$	TCLK1, TCLK2 rise time	R_L = 825 Ω, C_L = 100 pF, See Figure 1.			20†	ns
$t_{f(TIM)}$	TCLK1, TCLK2 fall time				20†	ns
$t_{hL(TIM)}$	Hold time TCLK1, TCLK2 low		$t_{c(C)}+20$			ns
$t_{hH(TIM)}$	Hold time TCLK1, TCLK2 high		$t_{c(C)}+20$			ns

†These values were derived from characterization data and not tested.

WATCHDOG TIMER TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{f(WDT)}$	Fall time, $\overline{\text{WDT}}$	R_L = 825 Ω, C_L = 100 pF, See Figure 1.			20†	ns
$t_{d(WDT)}$	CLKOUT to $\overline{\text{WDT}}$ valid		$0.25t_{c(C)}+20$			ns
$t_{w(WDT)}$	$\overline{\text{WDT}}$ output pulse duration		$8t_{c(C)}-20$		$8t_c+20$	ns

†These values were derived from characterization data and not tested.

EVENT MANAGER TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{f(CMP)}$	Fall time, CMP0-CMP5	R_L = 825 Ω, C_L = 100 pF, See Figure 1.			20†	ns
$t_{r(CMP)}$	Rise time, CMP0-CMP5				20†	ns

†These values were derived from characterization data and not tested.

timing requirements over recommended operating conditions

		TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{w(CAP)}$	CAP0-CAP3 input pulse duration	R_L = 825 Ω, C_L = 100 pF, See Figure 1.	$t_{c(C)}+20$			ns
$t_{su(CAP)}$	Capture input setup time before CLKOUT low		20†			ns

†These values were derived from characterization data and not tested.

SERIAL PORT-SYNCHRONOUS MODE TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{c(CLK-S)}$	Serial port clock cycle time†	R_L = 825 Ω, C_L = 100 pF, See Figure 1.	$t_{c(C)}$	$65{,}536t_{c(C)}$	ns
$t_{f(CLK-S)}$	TXD/CLK fall time†			20	ns
$t_{r(CLK-S)}$	TXD/CLK rise time†			20	ns
$t_{wL(CLK-S)}$	TXD/CLK low time†		$0.5t_{c(CLK-S)} - 20$	$0.5t_{c(CLK-S)} + 20$	ns
$t_{wH(CLK-S)}$	TXD/CLK high time†		$0.5t_{c(CLK-S)} - 20$	$0.5t_{c(CLK-S)} + 20$	ns
$t_{d(TX-S)}$	RXD/DATA output valid before TXD/CLK low†		$t_{wH(CLK-S)} - 20$		ns
$t_{h(TX-S)}$	RXD/DATA hold after TXD/CLK (internal) low†			$t_{d(TX-S)} + 20$	ns
$t_{d(TX-S)}$	RSD/DATA output valid before TXD/CLK low‡		$t_{wH} - 1.75t_{c(C)} + 20$		ns
$t_{h(TX-S)}$	RXD/DATA hold after after TXD/CLK low‡			$t_{wL} + 1.75t_{c(C)} + 20$	ns

†Internal clock
‡External clock

timing requirements over recommended operating conditions

		TEST CONDITIONS	MIN	MAX	UNIT
$t_{wL(CLK-S)}$	TXD/CLK low time (external)‡	R_L = 825 Ω, C_L = 100 pF, See Figure 1.	$2t_{c(C)}$		ns
$t_{wH(CLK-S)}$	TXD/CLK high time (external)‡		$2t_{c(C)}$		ns
$t_{su(RX-S)}$	RXD/DATA input setup before TXD/CLK low‡		0		ns
$t_{h(RX-S)}$	RXD/DATA input hold after TXD/CLK low‡			$2t_{c(C)} - 20$	ns
$t_{su(RX-S)}$	RXD/DATA input setup before TXD/CLK external low†		20		ns
$t_{h(RX-S)}$	RXD/DATA input hold after TXD/CLK external low†		$0.25t_{c(C)} + 20$		ns

†Internal clock
‡External clock

SERIAL PORT-CODEC MODE TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
$t_{d(TXD-C)}$	TXD output valid before CLKX low	R_L = 825 Ω, C_L = 100 pF, See Figure 1.	$0.5t_{c(C)} - 20$		ns
$t_{h(TXD-C)}$	TXD output hold after CLKX low			$t_{wL} + 1.75t_{c(C)} - 20$	ns

timing requirements over recommended operating conditions

		TEST CONDITIONS	MIN	MAX	UNIT
$t_{c(CLK-C)}$	CLKR, CLKX cycle time	R_L = 825 Ω, C_L = 100 pF, See Figure 1.	$3t_{c(C)}$§		ns
$t_{f(CLK-C)}$	CLKR, CLKX fall time			20§	ns
$t_{r(CLK-C)}$	CLKR/CLKX rise time			20§	ns
$t_{wL(CLK-C)}$	CLKR, CLKX high time		$1.5t_{c(C)} - 20$¶		ns
$t_{wH(CLK-C)}$	CLKR, CLKX low time		$1.5t_{c(C)} - 20$¶		ns
$t_{su(FSX)}$	FSX valid before CLKX low		$0.5t_{c(C)} - 20$		ns
$t_{su(FSR)}$	FSR valid before CLKR low		$0.5t_{c(C)} - 20$		ns
$t_{su(RXD-C)}$	RXD input setup time before CLKR low		0		ns
$t_{h(TXD-C)}$	RXD input hold time after CLKR low			$t_{c(C)} + 20$	ns

§These values were derived from characterization data and not tested.
¶This cycle time is only possible when CLK(R) and CLK(X) are synchronized with CLKOUT.

ADVANCE INFORMATION

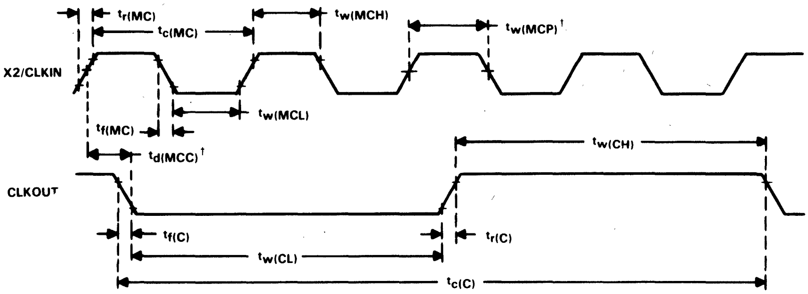
ADVANCE INFORMATION

timing diagrams

This section contains all the timing diagrams for the TMS320C14/E14 devices.

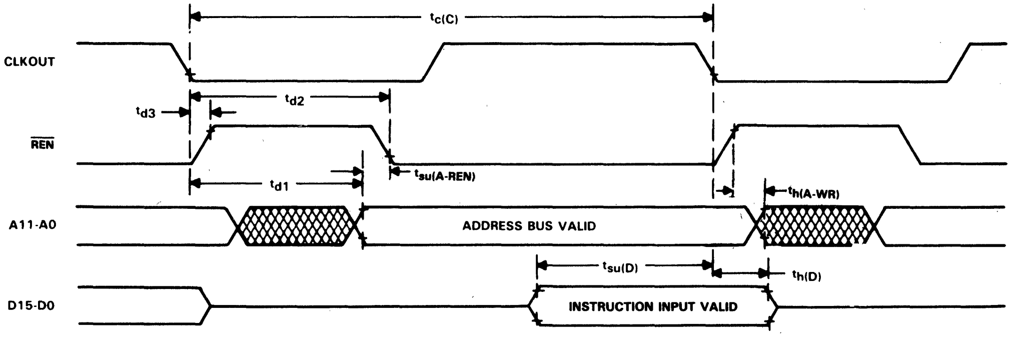
Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

clock timing

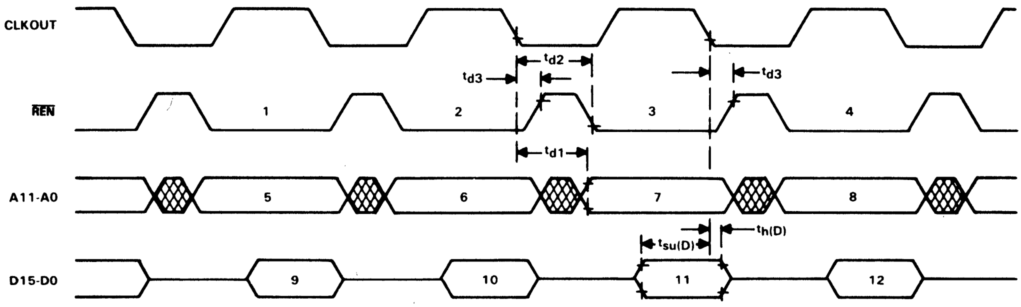


†$t_{d(MCC)}$ and $t_{w(MCP)}$ are referenced to an intermediate level of 1.5 volts on the CLKIN waveform.

memory read timing



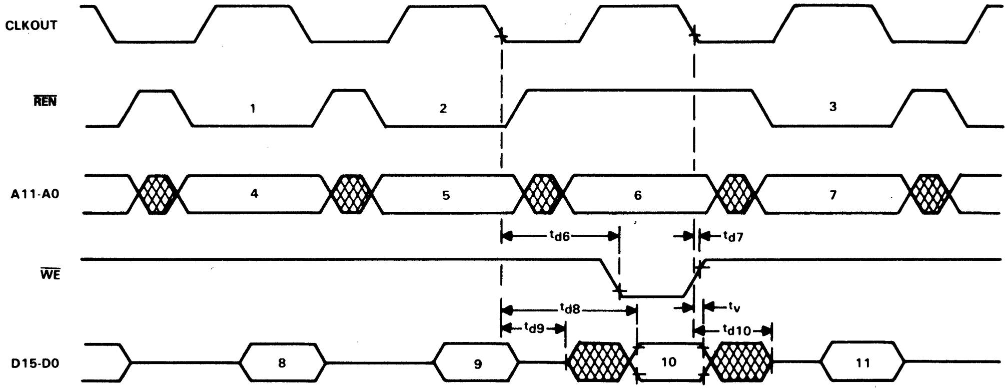
TBLR instruction timing



LEGEND:

1. TBLR INSTRUCTION PREFETCH
2. DUMMY PREFETCH
3. DATA FETCH
4. NEXT INSTRUCTION PREFETCH
5. ADDRESS BUS VALID
6. ADDRESS BUS VALID
7. ADDRESS BUS VALID
8. ADDRESS BUS VALID
9. INSTRUCTION INPUT VALID
10. INSTRUCTION INPUT VALID
11. DATA INPUT VALID
12. INSTRUCTION INPUT VALID

TBLW instruction timing



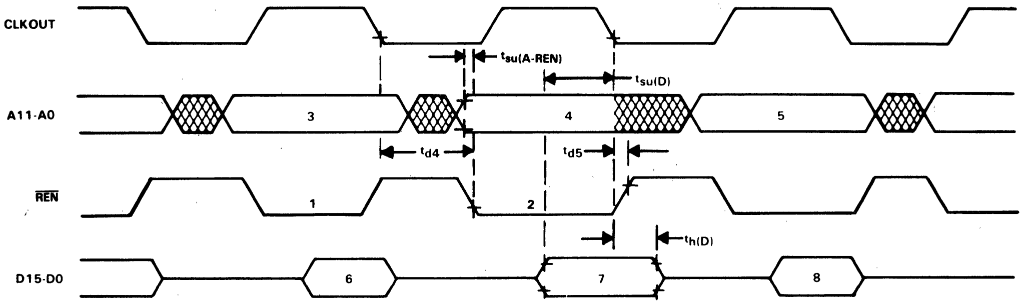
LEGEND:

1. TBLW INSTRUCTION PREFETCH
2. DUMMY PREFETCH
3. NEXT INSTRUCTION PREFETCH
4. ADDRESS BUS VALID
5. ADDRESS BUS VALID
6. ADDRESS BUS VALID
7. ADDRESS BUS VALID
8. INSTRUCTION INPUT VALID
9. INSTRUCTION INPUT VALID
10. DATA OUTPUT VALID
11. INSTRUCTION INPUT VALID

ADVANCE INFORMATION

ADVANCE INFORMATION

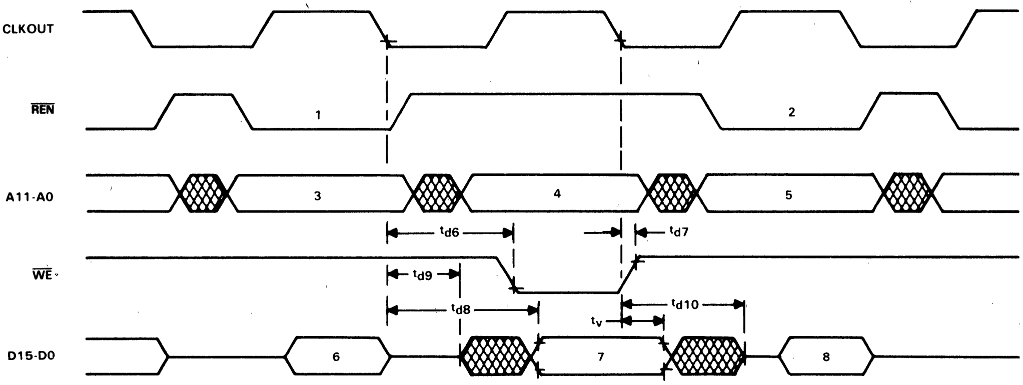
IN instruction timing



LEGEND:

1. IN INSTRUCTION PREFETCH
2. NEXT INSTRUCTION PREFETCH
3. ADDRESS BUS VALID
4. PERIPHERAL ADDRESS VALID
5. ADDRESS BUS VALID
6. INSTRUCTION INPUT VALID
7. DATA INPUT VALID
8. INSTRUCTION INPUT VALID

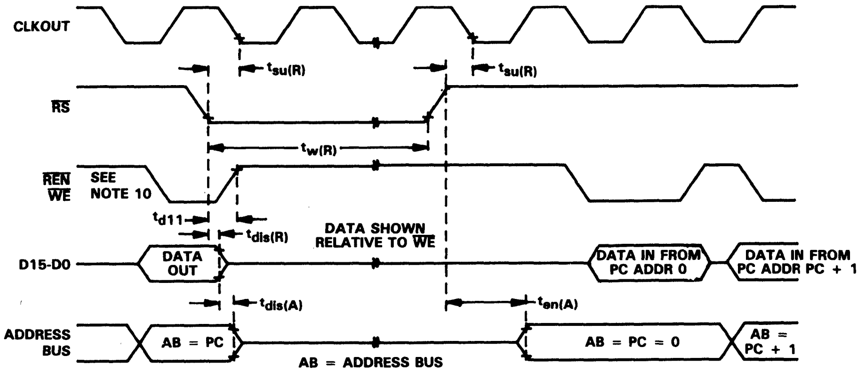
OUT instruction timing



LEGEND:

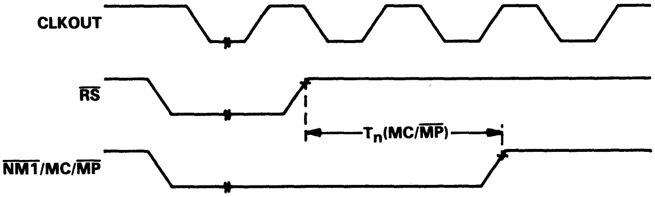
1. OUT INSTRUCTION PREFETCH
2. NEXT INSTRUCTION PREFETCH
3. ADDRESS BUS VALID
4. PERIPHERAL ADDRESS VALID
5. ADDRESS BUS VALID
6. INSTRUCTION INPUT VALID
7. DATA OUTPUT VALID
8. INSTRUCTION INPUT VALID

reset timing



NOTES:
6. $\overline{RS}$ forces $\overline{REN}$ and $\overline{WE}$ high and places data bus D0-D15 and address bus A0-A11 in a high-impedance state. AB outputs (and program counter) are synchronously cleared to zero after the next complete CLK cycle from $\overline{RS}$↑.
7. $\overline{RS}$ must be maintained for a minimum of five clock cycles.
8. Resumption of normal program will commence after one complete CLK cycle from $\overline{RS}$↑.
9. Due to the synchronizing action on $\overline{RS}$, time to execute the function can vary dependent upon when $\overline{RS}$↑ or $\overline{RS}$↓ occur in the CLK cycle.
10. Diagram shown is for definition purpose only. $\overline{WE}$ and $\overline{REN}$ are mutually exclusive.

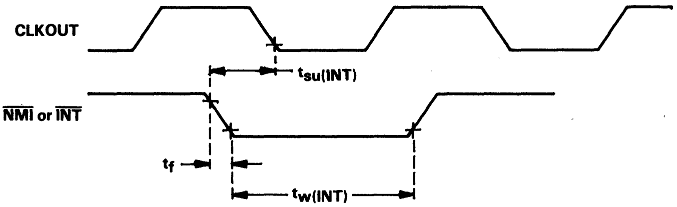
microcomputer/microprocessor mode timing diagram



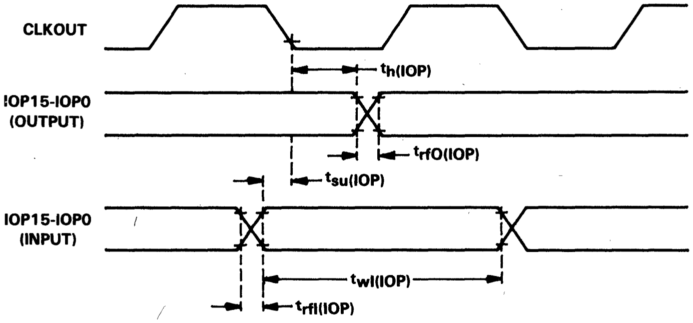
ADVANCE INFORMATION

ADVANCE INFORMATION

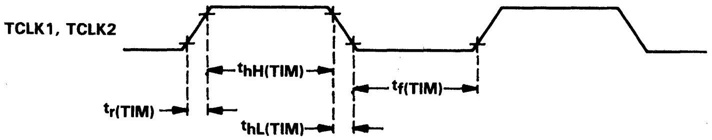
interrupt timing



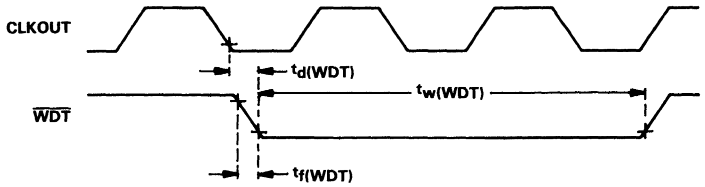
bit I/O timing



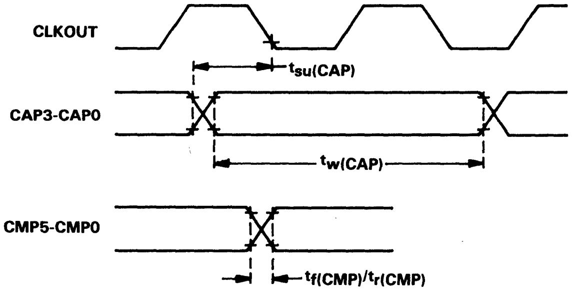
general purpose timers



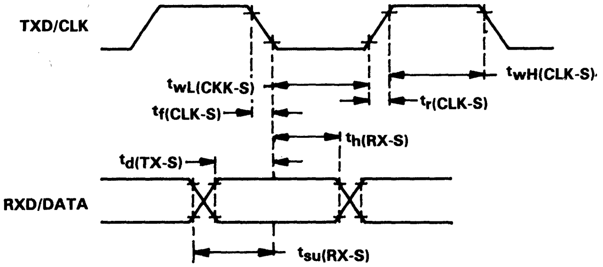
watchdog timer



event manager

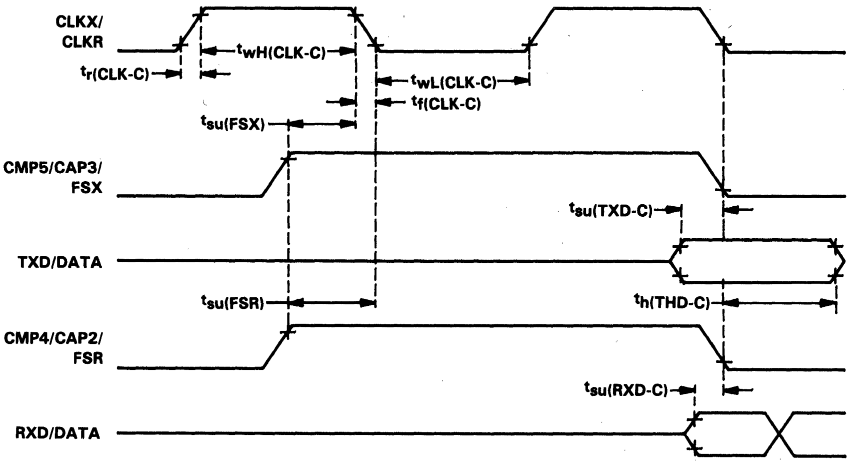


serial port - synchronous mode timing



ADVANCE INFORMATION

serial port - codec mode timing



ADVANCE INFORMATION

EPROM programming

The TMS320E14 includes a 4K x 16-bit industry-standard EPROM cell for prototyping and low-volume production. The TMS320C14 with a 4K-word masked ROM then provides a migration path for cost-effective production. An EPROM adapter socket (part #TMDX3270110), shown in Figure 2, is available to provide 68-pin to 28-pin conversion for programming the TMS320E14.

Key features of the EPROM cell include the normal programming operation as well as verification. The EPROM cell also includes a code protection feature that allows code to be protected against copyright violations.

The TMS320E14 EPROM cell is programmed using the same family and device codes as the TMS27C64 8K × 8-bit EPROM. The TMS27C64 EPROM series are ultraviolet-light erasable, electrically programmable, read-only memories, fabricated using HVCMOS technology. They are pin-compatible with existing 28-pin ROMs and EPROMs. These EPROMs operate from a single 5-V supply in the read mode; however, a 12.5-V supply is needed for programming. All programming signals are TTL level. For programming outside the system, existing EPROM programmers can be used. Locations may be programmed singly, in blocks, or at random.

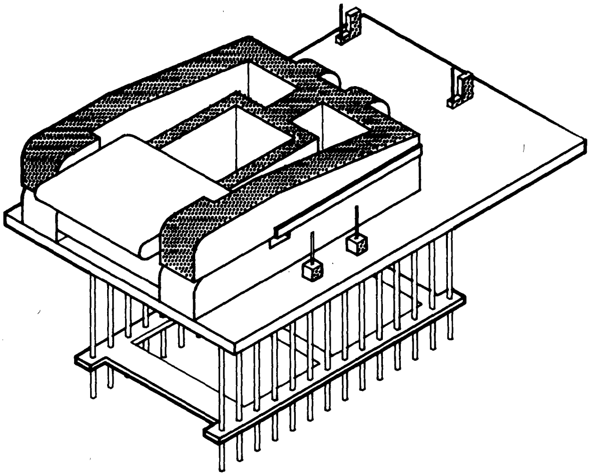


FIGURE 2. EPROM ADAPTER SOCKET

The TMS320E14 uses 12 address lines plus $\overline{WE}$ to address the 4K-word memory in byte format (8K-byte memory). In word format, the most-significant byte of each word is assigned an even address and the least-significant byte an odd address in the byte format. Programming information should be downloaded to EPROM programmer memory in a high-byte to low-byte order for proper programming of the devices (see Figure 3.)

TMS320C14 On-Chip Program Memory (Word Format)		TMS320E14 On-Chip Program Memory (Byte Format)		EPROM Programmer Memory Byte Format with Adapter Socket	
0(0000h)	1234h	0(0000h)	34h	0(0000h)	12h
1(0001h)	5678h	1(0001h)	12h	1(0001h)	34h
2(0002h)	9ABCh	2(0002h)	78h	2(0002h)	56h
3(0003h)	DEF0h	3(0003h)	56h	3(0003h)	78h
.	.	4(0004h)	BCh	4(0004h)	9Ah
.	.	5(0005h)	9Ah	5(0005h)	BCh
.	.	6(0006h)	F0h	6(0006h)	DEh
4095(0FFh)		7(0007h)	DEh	7(0007h)	F0h
		.	.	.	.
		.	.	.	.
		.	.	.	.
				8191(1FFFh)	

FIGURE 3. EPROM PROGRAMMING DATA FORMAT

Figure 4 shows the wiring conversion to program the TMS320E14 using the 28-pin pinout of the TMS27C64. The table of pin nomenclature provides a description of the TMS27C64 pins.

CAUTION

The TMS320E14 does not support the signature mode available with some EPROM programmers. The signature mode puts a high voltage (12.5 V DC) on pin A9. The TMS320E14 EPROM cell is not designed for this feature and will be damaged if subjected to it. A 3.9 kΩ resistor is standard on the TI programmer socket between pin A9 and the programmer. This protects the device from unintentional use of the signature mode.

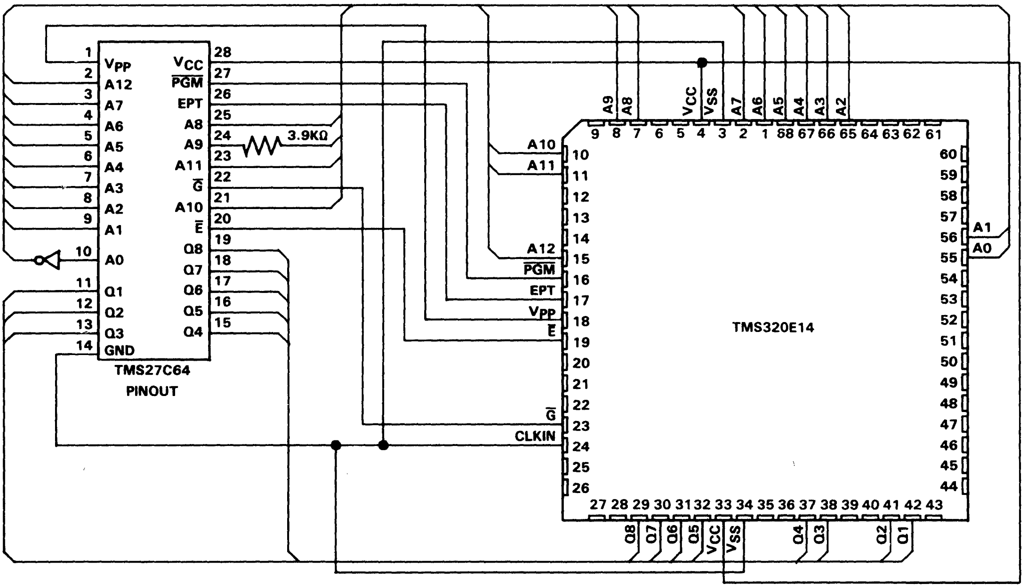


FIGURE 4. TMS320E14 EPROM PROGRAMMING CONVERSION TO TMS27C64 EPROM PINOUT

PIN NOMENCLATURE (TMS320E14)

NAME	I/O	DEFINITION
A12(MSB)-A0(LSB)	I	On-chip EPROM programming address lines
CLKIN	I	Clock oscillator input
$\overline{\text{E}}$	I	EPROM chip enable
EPT	I	EPROM test mode select
$\overline{\text{G}}$	I	EPROM output enable
GND	I	Ground
$\overline{\text{PGM}}$	I	EPROM write/program select
Q8(MSB)-Q1(LSB)	I/O	Data lines for byte-wide programming of on-chip 8K bytes of EPROM
$\overline{\text{RS}}$	I	Reset for initializing the device
V_{CC}	I	5-V to 6.5-V power supply
V_{PP}	I	12.5-V to 13-V power supply

ADVANCE INFORMATION

ADVANCE INFORMATION

Table 4 shows the programming levels required for programming, verifying, reading, and protecting the EPROM cell.

TABLE 4. TMS320E14 PROGRAMMING MODE LEVELS

SIGNAL NAME†	TMS320E14 PIN	TMS27C64 PIN	PROGRAM	PROGRAM VERIFY	READ	EPROM PROTECT	PROTECT VERIFY
$\overline{E}$	19	20	V_{IL}	V_{IL}	V_{IL}	V_{IH}	V_{IL}
$\overline{G}$	23	22	V_{IH}	$\overline{\text{PULSE}}$	$\overline{\text{PULSE}}$	V_{IH}	V_{IL}
$\overline{\text{PGM}}$	16	27	$\overline{\text{PULSE}}$	V_{IH}	V_{IH}	V_{IH}	V_{IH}
V_{PP}	18	1	V_{PP}	V_{PP}	V_{CC}	V_{PP}	V_{CCP}
V_{CC}	4,33	28	V_{CCP}	V_{CCP}	V_{CC}	V_{CCP}	V_{CCP}
V_{SS}	3,34	14	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
CLKIN	24	14	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
EPT	17	26	V_{SS}	V_{SS}	V_{SS}	V_{PP}	V_{PP}
Q1-Q8	42,41,38,37, 32-29	19-15,13-11	D_{IN}	Q_{OUT}	Q_{OUT}	$Q_8 = \overline{\text{PULSE}}$	$Q_8 =$ RBIT
A12-A7	15,11,10, 8,7,2	25,24,23, 21,3,2	ADDR	ADDR	ADDR	X	X
A6	1	4	ADDR	ADDR	ADDR	X	V_{IL}
A5	68	5	ADDR	ADDR	ADDR	X	X
A4	67	6	ADDR	ADDR	ADDR	V_{IH}	X
A3-A0	66,65,56,55	7-10	ADDR	ADDR	ADDR	X	X

†Signal names shown for TMS320E14 EPROM programming mode only.

LEGEND:
V_{IH} = TTL high level; V_{IL} = TTL low level; ADDR = byte address bit; V_{PP} = 12.5 V ± 0.25 V (FAST) or 13.0 V ± 0.25 V (SNAP!).
V_{CC} = 5 V ± 0.25 V; X = don't care; $\overline{\text{PULSE}}$ = low-going TTL pulse.
D_{IN} = byte to be programmed at ADDR; Q_{OUT} = byte stored at ADDR.
V_{CCP} = 6.0 V ± 0.25 V (FAST) or 6.5 V ± 0.25 V (SNAP!)

programming

Since every memory bit in the cell is a logic 1, the programming operation reprograms certain bits to 0. Once programmed, these bits can only be erased using ultraviolet light. The correct byte is placed on the data bus with V_{PP} set to the 12.5-V level. The $\overline{\text{PGM}}$ pin is then pulsed low to program in the zeroes.

erasure

Before programming, the device must be erased by exposing it to ultraviolet light. The recommended minimum exposure dose (UV-intensity X exposure-time) is 15 watt-seconds per square centimeter. A typical 12 milliwatt-seconds per square centimeter, filterless UV lamp will erase the device in 21 minutes. The lamp should be located about 2.5 centimeters above the chip during erasure. After exposure, all bits are in the high state.

verify/read

To verify correct programming, the EPROM cell can be read using either the verify or read line definitions shown in Table 4, assuming the inhibit bit has not been programmed.

program inhibit

Programming may be inhibited by maintaining a high level input on the $\overline{E}$ pin or $\overline{PGM}$ pin.

standard programming procedure

Before programming, the device must first be completely erased. Then the device can be programmed with the correct code. It is advisable to program unused sections with zeroes as a further security measure. After the programming is complete, the code programmed into the cell should be verified. If the cell passes verification, the next step is to program the ROM protect bit (RBIT). Once the RBIT programming is verified, an opaque label should be placed over the window to protect the EPROM cell from inadvertent erasure by ambient light. At this point, the programming is complete, and the device is ready to be placed into its destination circuit.

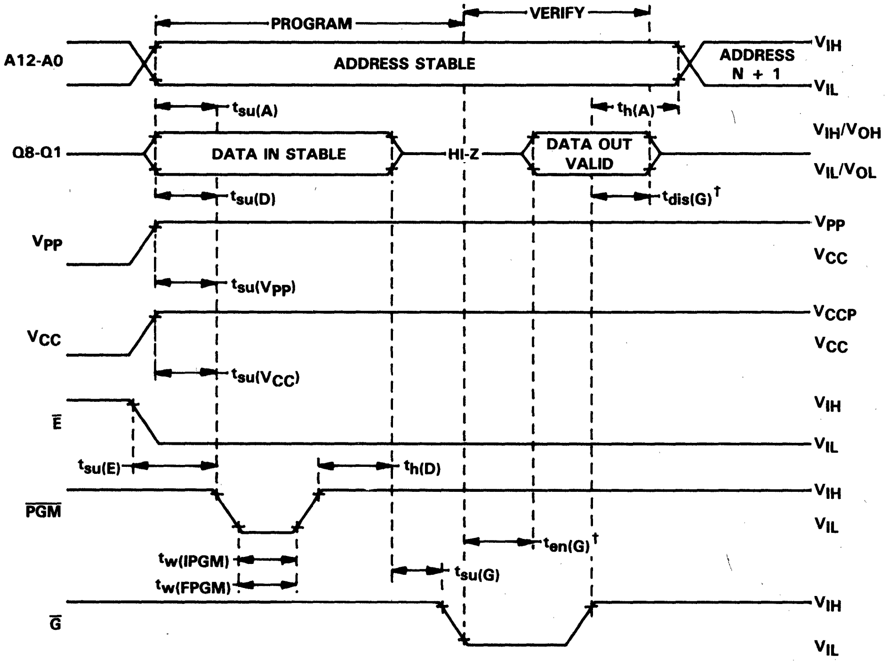
Refer to Appendix F of the TMS320C14/E14 User's Guide for additional information on EPROM programming.

recommended timing requirements for programming: V_{CC} = 6 V and V_{PP} = 12.5 V (Fast) or V_{CC} = 6.5 V and V_{PP} = 13.0 V (SNAP! Pulse), T_A = 25 °C (see Note 6)

			MIN	NOM	MAX	UNIT
$t_{w(IPGM)}$	Initial program pulse duration	Fast programming algorithm	0.95	1	1.05	ms
		SNAP! Pulse programming algorithm	95	100	105	µs
$t_{w(FPGM)}$	Final pulse duration	Fast programming only	2.85		78.75	ms
$t_{su(A)}$	Address setup time		2			µs
$t_{su(E)}$	$\overline{E}$ setup time		2			µs
$t_{su(G)}$	$\overline{G}$ setup time		2			µs
$t_{su(D)}$	Data setup time		2			µs
$t_{su}(VPP)$	V_{PP} setup time		2			µs
$t_{su}(VCC)$	V_{CC} setup time		2			µs
$t_{h(A)}$	Address hold time		0			µs
$t_{h(D)}$	Data hold time		2			µs

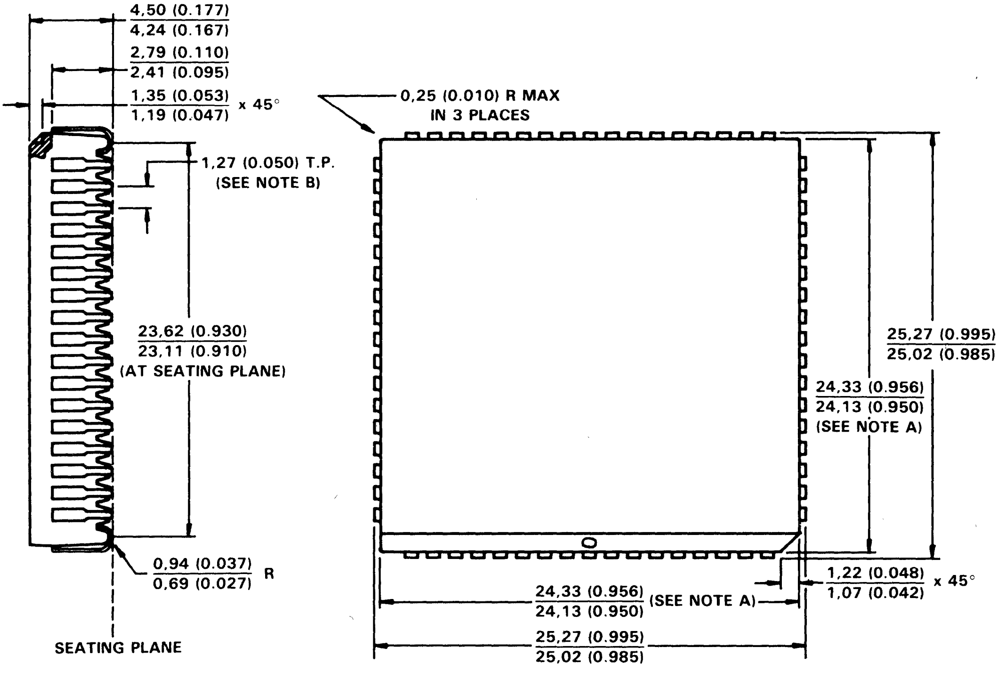
NOTE: 6. For all switching characteristics and timing measurements, input pulse levels are 0.40 V to 2.4 V and V_{PP} = 12.5 V ± 0.5 V during programming.

program cycle timing



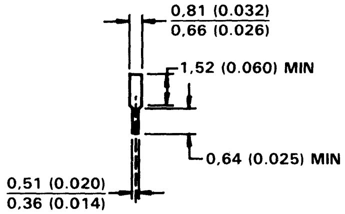
†$t_{dis(G)}$ and $t_{en(G(}$ are characteristics of the device but must be accommodated by the programmer.

68-lead plastic chip carrier package (FN suffix)



THERMAL RESISTANCE CHARACTERISTICS (SEE NOTE C)

$R_{\theta JA}$ °C/W	$R_{\theta JC}$ °C/W
60	17



LEAD DETAIL

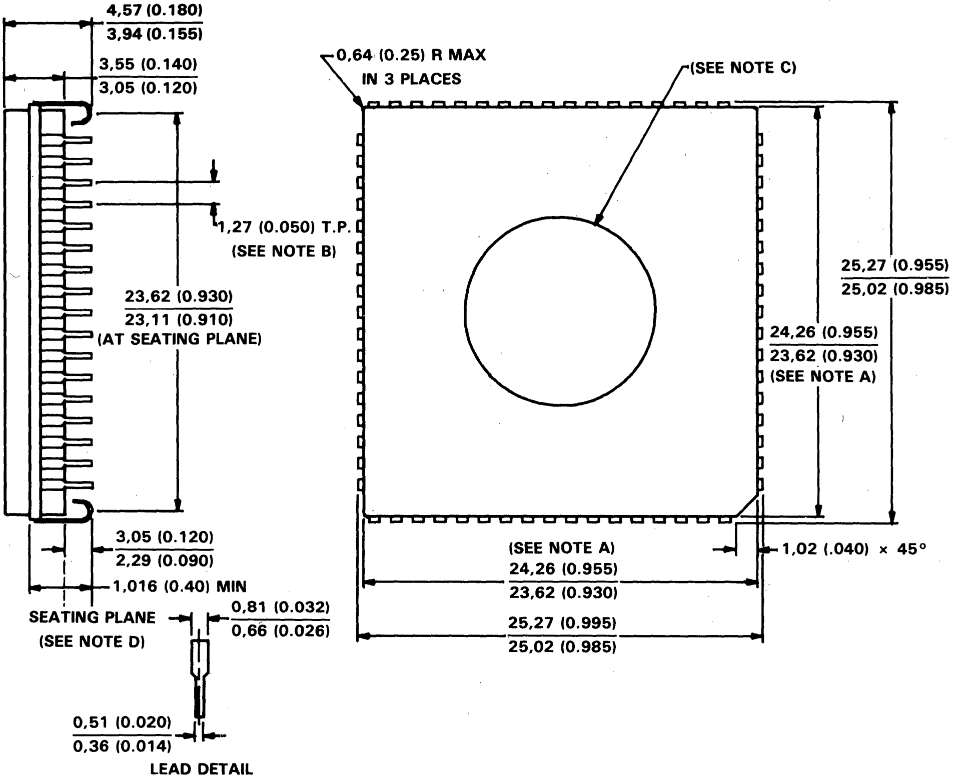
ADVANCE INFORMATION

NOTES:
A. Centerline of center pin each side is within 0,10 (0.004) of package centerline as determined by this dimension
B. Location of each pin is within 0,27 (0.005) of true position with respect to center pin on each side.
C. Thermal resistance calculations based on I_{CC} = 65 mA TYP at T_A = 70°C.

ALL LINEAR DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES.

ADVANCE INFORMATION

68-lead cerquad chip carrier package (FZ suffix)



ALL LINEAR DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES.

NOTES:
- A. Centerline of center pin each side is within 0,10 (0.004) of package centerline as determined by this dimension.
- B. Location of each pin is within 0,27 (0.005) of true position with respect to center pin on each side.
- C. Glass is optional.
- D. The lead contact points are planar within 0,15 (0.006).

INDEX

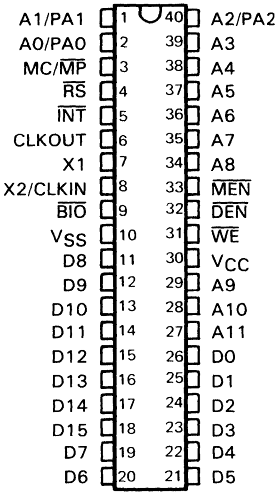
ADVANCE INFORMATION

SMJ32010
DIGITAL SIGNAL PROCESSOR

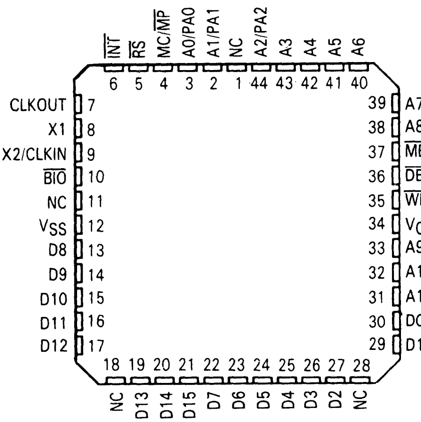
MAY 1983—REVISED MAY 1989

- **High Reliability Class B Processing**
- **200-ns Instruction Cycle**
- **144-Word On-Chip Data RAM**
- **Currently Microprocessor Mode Only (All Program Memory is Extended)**
- **External Memory Expansion to Total of 4K Words at Full Speed**
- **16-Bit Instruction/Data Word**
- **32-Bit ALU/Accumulator**
- **16 × 16-Bit Multiply in One Instruction Cycle**
- **0 to 16-Bit Barrel Shifter**
- **Eight Input and Eight Output Channels**
- **16-Bit Bidirectional Data Bus with 40-Megabits-per-Second Transfer Rate**
- **Interrupt with Full Context Save**
- **Signed Two's-Complement Fixed-Point Arithmetic**
- **2.4-Micron NMOS Technology**
- **Single 5-V Supply**
- **Operating Temperature Range . . . −55 °C to 100 °C (S Suffix)**

SMJ32010 . . . JD PACKAGE
(TOP VIEW)



SMJ32010 . . . FD PACKAGE
(TOP VIEW)



description

The SMJ32010 is a member of the TMS320 digital signal processing family, designed to support a wide range of high-speed or numeric-intensive applications. This 16/32-bit single-chip microcomputer combines the flexibility of a high-speed controller with the numerical capability of an array processor, thereby offering an inexpensive alternative to multichip bit-slice processors. The TMS320 family contains the first MOS microcomputers capable of executing five million instructions per second. This high throughput is the result of the comprehensive, efficient, and easily programmed instruction set and of the highly pipelined architecture. Special instructions have been incorporated to speed the execution of digital signal processing (DSP) algorithms.

The TMS320 family's unique versatility and power give the design engineer a new approach to a variety of complicated applications. In addition, these microcomputers are capable of providing the multiple functions often required for a single application. For example, the TMS320 family can synthesize and recognize speech, sense objects with radar or optical intelligence, and perform mechanical operations through digital servo loop computations.

architecture

The TMS320 family utilizes a modified Harvard architecture for speed and flexibility. In a strict Harvard architecture, program and data memory lie in two separate spaces, permitting a full overlap of the instruction fetch and execution. The TMS320 family's modification of the Harvard architecture allows transfers between program and data spaces, thereby increasing the flexibility of the device. This modification permits coefficients stored in program memory to be read into the RAM, eliminating the need for a separate coefficient ROM. It also makes available immediate instructions and subroutines based on computed values.

The SMJ32010 utilizes hardware to implement functions that other processors typically perform in software. For example, this device contains a hardware multiplier to perform a multiplication in a single 200-ns cycle. There is also a hardware barrel shifter for shifting data on its way into the ALU. Extra hardware has been included so that auxiliary registers, which provide indirect data RAM addresses, can be configured in an autoincrement/decrement mode for single-cycle manipulation of data tables. This hardware-intensive approach gives the design engineer the type of power previously unavailable on a single chip.

PIN NOMENCLATURE

SIGNATURE	I/O/Z†	DEFINITION
A11-A0/ PA2-PA0	O	External address bus. I/O port address multiplexed over PA2-PA0.
$\overline{\text{BIO}}$	I	External polling input.
CLKOUT	O	System clock output, ¼ crystal/CLKIN frequency.
D15-D0	I/O/Z	16-bit data bus.
$\overline{\text{DEN}}$	O	Data enable indicates the processor accepting input data on D15-D0.
$\overline{\text{INT}}$	I	Interrupt.
MC/$\overline{\text{MP}}$	I	Memory mode select pin. High selects microcomputer mode. Low selects microprocessor mode.
$\overline{\text{MEN}}$	O	Memory enable indicates that D15-D0 will accept external memory instruction.
$\overline{\text{RS}}$	I	Reset used to initialize the device.
V_{CC}	I	Power.
V_{SS}	I	Ground.
$\overline{\text{WE}}$	O	Write-enable indicates valid data on D15-D0.
X1	O	Crystal output.
X2/CLKIN	I	Crystal input or external clock input.

†Input/Output/High-impedance state

32-bit ALU/accumulator

The SMJ32010 contains a 32-bit ALU and accumulator that support double-precision arithmetic. The ALU operates on 16-bit words taken from the data RAM or derived from immediate instructions. Besides the usual arithmetic instructions, the ALU can perform Boolean operations, providing the bit manipulation ability required of a high-speed controller.

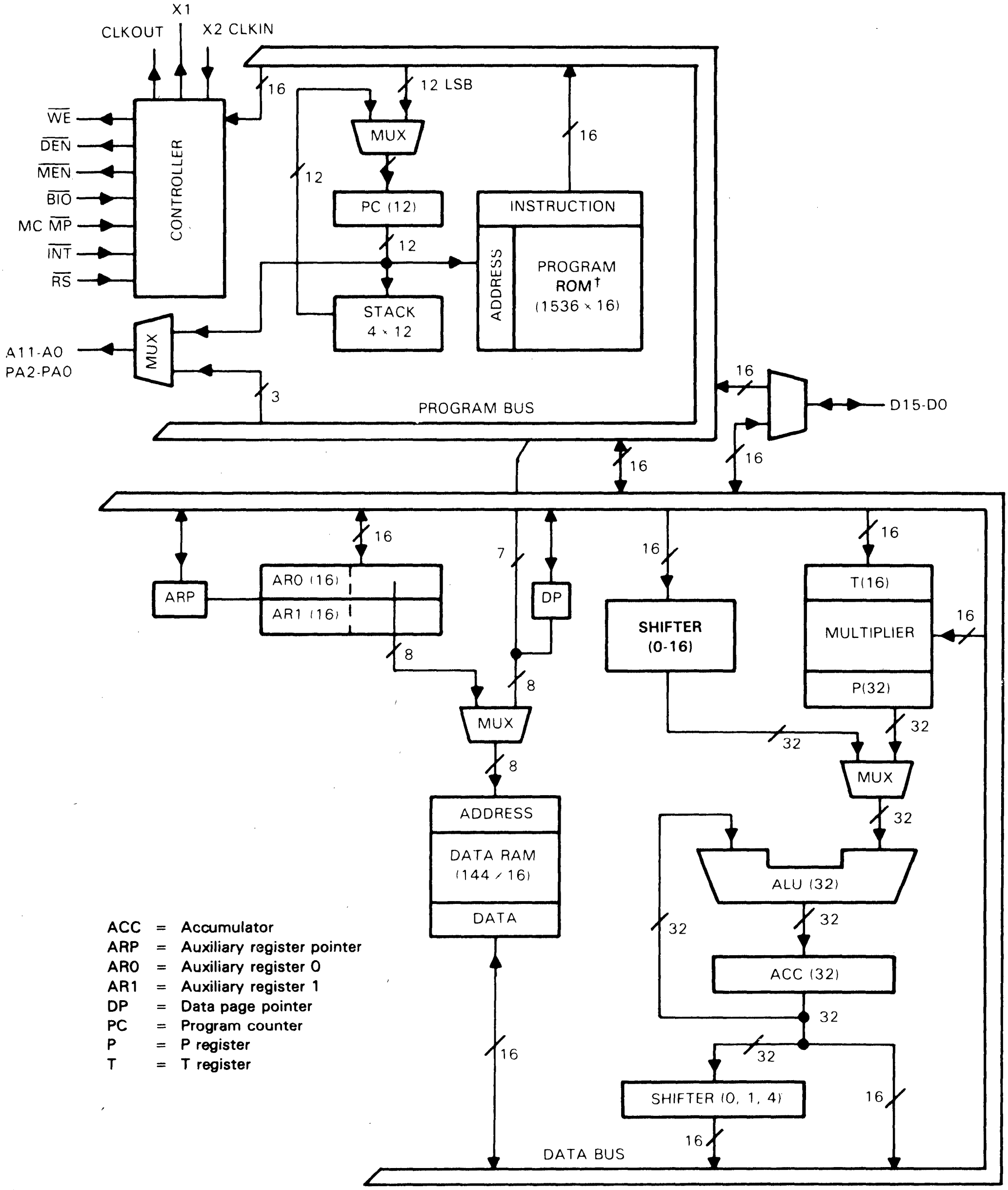
shifters

A barrel shifter is available for left-shifting data 0 to 16 places before it is loaded into, subtracted from, or added to the accumulator. This shifter extends the high-order bit of the data word and zero-fills the low-order bits for two's-complement arithmetic. A second shifter left-shifts the upper half of the accumulator 0, 1, or 4 places while it is being stored in the data RAM. Both shifters are useful for scaling and bit extraction.

16 × 16-bit parallel multiplier

The SMJ32010's multiplier performs a 16 × 16-bit, two's-complement multiplication in one 200-ns instruction cycle. The 16-bit T Register temporarily stores the multiplicand; the P Register stores the 32-bit result. Multiplier values either come from the data memory or are derived immediately from the MPYK (multiply immediate) instruction word. The fast on-chip multiplier allows the SMJ32010 to perform such fundamental operations as convolution, correlation, and filtering at the rate of 2.5 million samples per second.

functional block diagram



ACC	=	Accumulator
ARP	=	Auxiliary register pointer
AR0	=	Auxiliary register 0
AR1	=	Auxiliary register 1
DP	=	Data page pointer
PC	=	Program counter
P	=	P register
T	=	T register

†The Program ROM is available in SMJ320M10 only.

input/output

The SMJ32010's 16-bit parallel data bus can be utilized to perform I/O functions at burst rates of 40 million bits per second. Available for interfacing to peripheral devices are 128 input and 128 output bits consisting of eight 16-bit multiplexed input ports and eight 16-bit multiplexed output ports. In addition, a polling input for bit test and jump operations ($\overline{\text{BIO}}$) and an interrupt pin ($\overline{\text{INT}}$) have been incorporated for multi-tasking.

interrupts and subroutines

The SMJ32010 contains a four-level hardware stack for saving the contents of the program counter during interrupts and subroutine calls. Instructions are available for saving the SMJ32010's complete context. The instructions, PUSH stack from accumulator, and POP stack to accumulator, permit a level of nesting restricted only by the amount of available RAM. The interrupts used in the SMJ32010 are maskable.

instruction set

The SMJ32010's comprehensive instruction set supports both numeric-intensive operations, such as signal processing, and general purpose operations, such as high-speed control. The instruction set, explained in Tables 1 and 2, consists primarily of single-cycle single-word instructions, permitting execution rates of up to five million instructions per second. Only frequently used branch and I/O instructions are multicycle.

The SMJ32010 also contains a number of instructions that shift data a part of an arithmetic operation. These all execute in a single cycle and are very useful for scaling data in parallel with other operations.

Three main addressing modes are available with the SMJ32010 instruction set: direct, indirect, and immediate addressing.

direct addressing

In direct addressing, seven bits of the instruction word concatenated with the data page pointer form the data memory address. This implements a paging scheme in which the first page contains 128 words and the second page contains 16 words. In a typical application, infrequently accessed variables, such as those used for servicing an interrupt, are stored on the second page. The instruction format for direct addressing is shown below.

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">OPCODE</td><td>0</td><td colspan="7">dma</td></tr>
</table>

Bit 7 = 0 defines direct addressing mode. The opcode is contained in bits 15 through 8. Bits 6 through 0 contain data memory address.

The seven bits of the data memory address (dma) field can directly address up to 128 words (1 page) of data memory. Use of the data memory page pointer is required to address the full 144 words of data memory.

Direct addressing can be used with all instructions requiring data operands except for the immediate operand instructions.

indirect addressing

Indirect addressing forms the data memory address from the least significant eight bits of one of two auxiliary registers, AR0 and AR1. The auxiliary register pointer (ARP) selects the current auxiliary register. The auxiliary registers can be automatically incremented or decremented in parallel with the execution of any indirect instruction to permit single-cycle manipulation of data tables. The instruction format for indirect addressing is as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPCODE								1	0	INC	DEC	NAR	0	0	ARP

Bit 7 = 1 defines indirect addressing mode. The opcode is contained in bits 15 through 8. Bits 6 through 0 contain indirect addressing contol bits.

Bit 3 and bit 0 control the Auxiliary Register Pointer (ARP). If bit 3 = 0, the contents of bit 0 are loaded into the ARP after execution of the current instruction. If bit 3 = 1, the contents of the ARP remain unchanged. ARP = 0 defines the contents of AR0 as a memory address. ARP = 1 defines the contents of AR1 as a memory address. Note that NAR indicates the new auxiliary register control bit.

Bit 5 and bit 4 control the auxiliary registers. If bit 5 = 1, the current auxiliary register is incremented by 1 after execution. If bit 4 = 1, the current auxiliary register is decremented by 1 after execution. If bit 5 and bit 4 are 0, then neither auxiliary register is incremented nor decremented. Bits 6, 2, and 1 are reserved and should always be programmed to 0.

Indirect addressing can be used with all instructions requiring data operands, except for the immediate operand instructions.

immediate addressing

The SMJ32010 instruction set contains special "immediate" instructions. These instructions derive data from part of the instruction word rather than from the data RAM. Some very useful immediate instructions are multiply immediate (MPYK), load accumulator immediate (LACK), and load auxiliary register immediate (LARK).

instruction set summary

Table 1 lists the symbols and abbreviations used in Table 2, the instruction set summary. Table 2 contains a short description and the opcode for each TMS320 instruction. The summary is arranged according to function and alphabetized within each functional group.

TABLE 1. INSTRUCTION SYMBOLS

SYMBOL	MEANING
ACC	Accumulator
D	Data memory address field
I	Addressing mode bit
K	Immediate operand field
PA	3-bit port address field
R	1-bit operand field specifying auxiliary register
S	4-bit left-shift code
X	3-bit accumulator left-shift field

TABLE 2. TMS320 FIRST-GENERATION INSTRUCTION SET SUMMARY

ACCUMULATOR INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABS	Absolute value of accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0
ADD	Add to accumulator with shift	1	1	0	0	0	0	◄—S—►				I	◄——D——►						
ADDH	Add to high-order accumulator bits	1	1	0	1	1	0	0	0	0	0	I	◄——D——►						
ADDS	Add to accumulator with no sign extension	1	1	0	1	1	0	0	0	0	1	I	◄——D——►						
AND	AND with accumulator	1	1	0	1	1	1	1	0	0	1	I	◄——D——►						
LAC	Load accumulator with shift	1	1	0	0	1	0	◄—S—►				I	◄——D——►						
LACK	Load accumulator immediate	1	1	0	1	1	1	1	1	1	0	◄——K——►							
OR	OR with accumulator	1	1	0	1	1	1	1	0	1	0	I	◄——D——►						
SACH	Store high-order accumulator bits with shift	1	1	0	1	0	1	1	◄X►			I	◄——D——►						
SACL	Store low-order accumulator bits	1	1	0	1	0	1	0	0	0	0	I	◄——D——►						
SUB	Subtract from accumulator with shift	1	1	0	0	0	1	◄—S—►				I	◄——D——►						
SUBC	Conditional subtract (for divide)	1	1	0	1	1	0	0	1	0	0	I	◄——D——►						
SUBH	Subtract from high-order accumulator bits	1	1	0	1	1	0	0	0	1	0	I	◄——D——►						
SUBS	Subtract from accumulator with no sign extension	1	1	0	1	1	0	0	0	1	1	I	◄——D——►						
XOR	Exclusive OR with accumulator	1	1	0	1	1	1	1	0	0	0	I	◄——D——►						
ZAC	Zero accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1
ZALH	Zero accumulator and load high-order bits	1	1	0	1	1	0	0	1	0	1	I	◄——D——►						
ZALS	Zero accumulator and load low-order bits with no sign extension	1	1	0	1	1	0	0	1	1	0	I	◄——D——►						

AUXILIARY REGISTER AND DATA PAGE POINTER INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LAR	Load auxiliary register	1	1	0	0	1	1	1	0	0	R	I	◄——D——►						
LARK	Load auxiliary register immediate	1	1	0	1	1	1	0	0	0	R	◄——K——►							
LARP	Load auxiliary register pointer immediate	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	K
LDP	Load data memory page pointer	1	1	0	1	1	0	1	1	1	1	I	◄——D——►						
LDPK	Load data memory page pointer immediate	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	K
MAR	Modify auxiliary register and pointer	1	1	0	1	1	0	1	0	0	0	I	◄——D——►						
SAR	Store auxiliary register	1	1	0	0	1	1	0	0	0	R	I	◄——D——►						

TABLE 2. TMS320 FIRST-GENERATION INSTRUCTION SET SUMMARY (CONTINUED)

BRANCH INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B	Branch unconditionally	2	2	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BANZ	Branch on auxiliary register not zero	2	2	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BGEZ	Branch if accumulator ≥ 0	2	2	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BGZ	Branch if accumulator > 0	2	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BIOZ	Branch on $\overline{\mathrm{BIO}}$ = 0	2	2	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BLEZ	Branch if accumulator ≤ 0	2	2	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BLZ	Branch if accumulator < 0	2	2	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BNZ	Branch if accumulator ≠ 0	2	2	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BV	Branch on overflow	2	2	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BZ	Branch if accumulator = 0	2	2	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
CALA	Call subroutine from accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0
CALL	Call subroutine immediately	2	2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
RET	Return from subroutine or interrupt routine	2	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	1

T REGISTER, P REGISTER, AND MULTIPLY INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APAC	Add P register to accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1
LT	Load T register	1	1	0	1	1	0	1	0	1	0	I	← D →						
LTA	LTA combines LT and APAC into one instruction	1	1	0	1	1	0	1	1	0	0	I	← D →						
LTD	LTD combines LT, APAC, and DMOV into one instruction	1	1	0	1	1	0	1	0	1	1	I	← D →						
MPY	Multiply with T register, store product in P register	1	1	0	1	1	0	1	1	0	1	I	← D →						
MPYK	Multiply T register with immediate operand; store product in P register	1	1	1	0	0	← K →												
PAC	Load accumulator from P register	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	0
SPAC	Subtract P register from accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0

TABLE 2. TMS320 FIRST-GENERATION INSTRUCTION SET SUMMARY (CONCLUDED)

CONTROL INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DINT	Disable interrupt	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1
EINT	Enable interrupt	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0
LST	Load status register	1	1	0	1	1	1	1	0	1	1	I	←— D —→						
NOP	No operation	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
POP	POP stack to accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1
PUSH	PUSH stack from accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0
ROVM	Reset overflow mode	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	0
SOVM	Set overflow mode	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1
SST	Store status register	1	1	0	1	1	1	1	1	0	0	I	←— D —→						

I/O AND DATA MEMORY OPERATIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMOV	Copy contents of data memory location into next higher location	1	1	0	1	1	0	1	0	0	1	I	←— D —→						
IN	Input data from port	2	1	0	1	0	0	0	←PA→			I	←— D —→						
OUT	Output data to port	2	1	0	1	0	0	1	←PA→			I	←— D —→						
TBLR	Table read from program memory to data RAM	3	1	0	1	1	0	0	1	1	1	I	←— D —→						
TBLW	Table write from data RAM to program memory	3	1	0	1	1	1	1	1	0	1	I	←— D —→						

development support

Together, Texas Instruments and its authorized third-party suppliers offer an extensive line of development support products to assist the user in all aspects of TMS320 first-generation-based design and development. These products range from development and application software to complete hardware development and evaluation systems. Table 3 lists the development support products for the first-generation TMS320 devices.

System development may begin with the use of the simulator, evaluation module (EVM), or emulator (XDS), along with an assembler/linker. These tools give the TMS320 user various means of evaluation, from software simulation of the first-generation TMS320s (simulator) to full-speed in-circuit emulation with hardware and software break point trace and timing capabilities (XDS).

Software and hardware can be developed simultaneously by using the macro assembler/linker or simulator for software development, the XDS for hardware development, and the EVM for both software development and limited hardware development.

Many third-party vendors offer additional development support for the first-generation TMS320s, including assembler/linkers, simulators, high-level languages, applications software, algorithm development tools, application boards, software development boards, and in-circuit emulators. Refer to the *TMS320 Family Development Support Reference Guide* (SPRU011A) for further information about TMS320 development support products offered by both Texas Instruments and its third-party suppliers.

Additional support for the TMS320 products consists of an extensive library of product and applications documentation. Three-day DSP design workshops are offered by the TI Regional Technology Centers (RTCs). These workshops provide insight into the architecture and the instruction set of the first-generation TMS320s as well as hands-on training with the TMS320 development tools. When technical questions arise in regard to a TMS320 Family member, contact Texas Instruments TMS320 Hotline via (713) 274-2320. Or, keep informed on the latest TI and third-party development support tools by accessing the DSP Bulletin Board Service (BBS) via (713) 274-2323. Application source code may also be accessed through the BBS via a 2400-, 1200-, or 300-bps modem.

TABLE 3. TMS320 FIRST-GENERATION SOFTWARE AND HARDWARE SUPPORT

SOFTWARE TOOLS	PART NUMBER
Macro Assembler/Linker	
PC/MS-DOS	TMDS3242850-02
VAX/VMS	TMDS3242250-08
VAX ULTRIX	TMDS3242260-08
SUN-3 UNIX	TMDS3242550-08
Simulator	
PC/MS-DOS	TMDS3240811-02
VAX/VMS	TMDS3240211-08
Digital Filter Design Package (DFDP)	
IBM PC PC-DOS	DFDP/IBM002
DSP Software Library	
PC/MS-DOS	TMDC3240812-12
VAX/VMS	TMDC3240212-18
TMS320 Bell 212A Modem Software	
PC/MS-DOS	TMDX3240813-12
Data Encryption Standard Software	
PC/MS-DOS	TMDX3240814-12
HARDWARE TOOLS	**PART NUMBER**
Evaluation Tools	
Evaluation Module (EVM)	RTC/EVM320A-03
Analog Interface Board 1 (AIB1)	RTC/EVM320C-06
Analog Interface Board 2 (AIB2)	RTC/AIB320A-06
EPROM DSP Starter Kit (TMS320E15)	RTC/EVM320E-15
XDS/22 Emulators	
TMS320C10/C15	TMDS3262211
TMS320C14	TMDX3262214
TMS320C17	TMDX3262217
XDS/22 Upgrade Kits	
TMS32010 → TMS320C10/C15	TMDS3282215
TMS320C10/C15 → TMS320C14	TMDX3285010 and TMDX3285018
TMS320C10/C15 → TMS320C17	TMDX3285014 and TMDX3285018
EPROM Programming Adaptor Sockets	
40- to 28-pin (TMS320E15/E17)	RTC/PGM320A-06
44- to 28-pin (TMS320E15/E17)	RTC/PGM320C-06
68- to 28-pin (TMS320E14)	TMDX3270110
Additional Target Connector	
44-pin PLCC (TMS320C10)	TMDX3288810

documentation support

Extensive documentation supports the first-generation TMS320 devices from product announcement through applications development. The types of documentation include data sheets with design specifications, complete user's guides, and 750 pages of application reports published in the book, *Digital Signal Processing Applications with the TMS320 Family* (SPRA012A).

A series of DSP textbooks is being published by Prentice-Hall and John Wiley & Sons to support digital signal processing research and education. The TMS320 newsletter, *Details on Signal Processing,* is published quarterly and distributed to update TMS320 customers on product information. The TMS320 DSP bulletin board service provides access to large amounts of information pertaining to the TMS320 family.

Refer to the *TMS320 Family Development Support Reference Guide* (SPRU011A) for further information about TMS320 documentation. To receive copies of first-generation TMS320 literature, call the Customer Response Center at 1-800-232-3200.

absolute maximum ratings over specified temperature range (unless otherwise noted)†

Supply voltage range, V_{CC}‡	−0.3 V to 7 V
Input voltage range	−0.3 V to 7 V
Output voltage range	−0.3 V to 7 V
Continuous power dissipation	1.5 W
Maximum operating case temperature	100°C
Minimum operating free-air temperature	−55°C
Storage temperature range	−65°C to 150°C

† Stresses beyond 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 beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

‡ All voltage values are with respect to V_{SS}.

recommended operating conditions

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage		4.5	5	5.5	V
V_{SS}	Supply voltage			0		V
V_{IH}	High-level input voltage	All inputs except CLKIN	2			V
		CLKIN	2.8			
V_{IL}	Low-level input voltage	X2/CLKIN and data			0.8	V
		$\overline{BIO}$, $\overline{INT}$, MC/$\overline{MP}$, $\overline{RS}$			0.7	
I_{OH}	High-level output current (all outputs)				300	µA
I_{OL}	Low-level output current (all outputs)				2	mA
T_C	Maximum operating case temperature				100	°C
T_A	Minimum free-air temperature		−55			°C

electrical characteristics over specified temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS		MIN	TYP[†]	MAX	UNIT
V_{OH}	High-level output voltage		I_{OH} = MAX		2.4	3		V
V_{OL}	Low-level output voltage		I_{OL} = MAX			0.3	0.5	V
I_{OZ}	Off-state output current		V_{CC} = MAX	V_O = 2.4 V			20	μA
				V_O = 0.4 V			−20	
I_I	Input current		V_I = V_{SS} to V_{CC}				±50	μA
I_{CC}	Supply current		V_{CC} = MAX, f_x = MAX,			180	275	mA
C_i	Input capacitance	Data bus	f = 1 MHz, All other pins 0 V			25		pF
		All others				15		
C_o	Output capacitance	Data bus				25		
		All others				10		

[†]All typical values are at V_{CC} = 5 V, T_A = 25°C.

CLOCK CHARACTERISTICS AND TIMING

The SMJ32010 can use either its internal oscillator or an external frequency source for a clock.

internal clock option

The internal oscillator is enabled by connecting a crystal across X1 and X2/CLKIN (see Figure 1). The frequency of CLKOUT is one-fourth the crystal fundamental frequency. The crystal should be fundamental mode, and parallel resonant, with an effective series resistance of 30 ohms, a power dissipation of 1 mW, and be specified at a load capacitance of 20 pF.

PARAMETER	TEST CONDITIONS	MIN	NOM	MAX	UNIT
Crystal frequency, f_x[†]	−55°C to 100°C	6.7[‡]		20[‡]	MHz
C1, C2			10		pF

[†]An 8 MHz crystal was used in the test.
[‡]Value derived from characterization data. The value is guaranteed bu+ not tested.

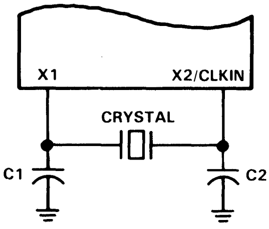


FIGURE 1. INTERNAL CLOCK OPTION

external clock option

An external frequency source can be used by injecting the frequency directly into X2/CLKIN with X1 left unconnected. The external frequency injected must conform to the specifications listed in the following table.

timing requirements over recommended operating conditions

PARAMETER		MIN	NOM	MAX	UNIT
$t_{c(MC)}$	Master clock cycle time	50		150	ns
$t_{r(MC)}$	Rise time master clock input		5	10§	ns
$t_{f(MC)}$	Fall time master clock input		5	10§	ns
$t_{w(MCL)}$	Pulse duration master clock low, $t_{c(MC)}$ = 50 ns		20		ns
$t_{w(MCH)}$	Pulse duration master clock high, $t_{c(MC)}$ = 50 ns		20		ns

§CLKIN rise and fall times must be less than 10 ns.

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	NOM	MAX	UNIT
$t_{c(C)}$	CLKOUT cycle time	R_L = 825 Ω, C_L = 100 pF, See Figure 2	200			ns
$t_{r(C)}$	CLKOUT rise time			10		ns
$t_{f(C)}$	CLKOUT fall time			8		ns
$t_{w(CL)}$	Pulse duration, CLKOUT low			92		ns
$t_{w(CH)}$	Pulse duration, CLKOUT high			90		ns

PARAMETER MEASUREMENT INFORMATION

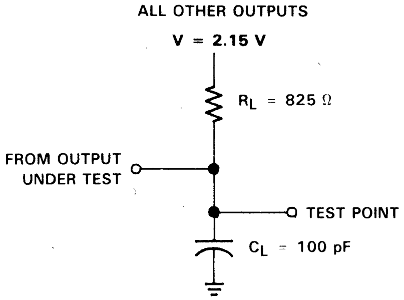
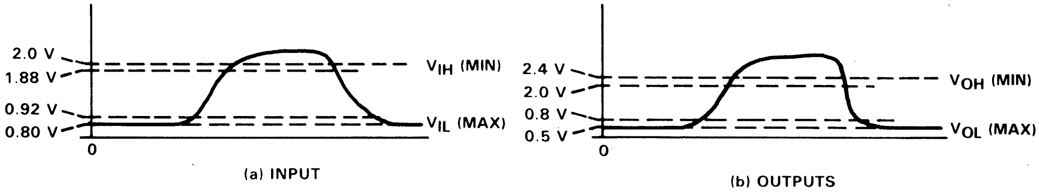


FIGURE 2. TEST LOAD CIRCUIT



(a) INPUT

(b) OUTPUTS

FIGURE 3. VOLTAGE REFERENCE LEVELS

input synchronization requirements

For systems using asynchronous inputs to the $\overline{\text{INT}}$ and $\overline{\text{BIO}}$ pins on the SMJ32010, the external hardware shown in the Figure 4 is recommended to ensure proper execution of interrupts and the BIOZ instruction. This hardware synchronizes the $\overline{\text{INT}}$ and $\overline{\text{BIO}}$ input signals with the rising edge of CLKOUT on the SMJ32010. The pulse width required for these input signals is $t_{c(C)}$, which is one SMJ32010 clock cycle, plus sufficient setup time for the flip-flop (dependent upon the flip-flop used). Note that these input synchronization requirements apply only to NMOS versions of the SMJ32010 and not to other members of the SMJ320 family.

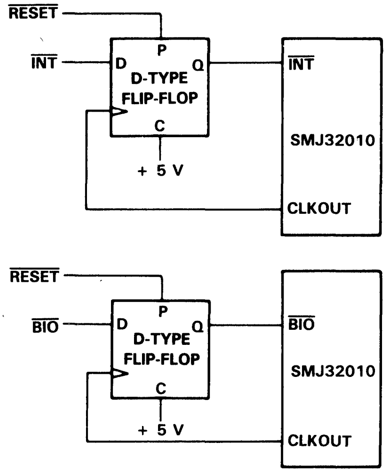
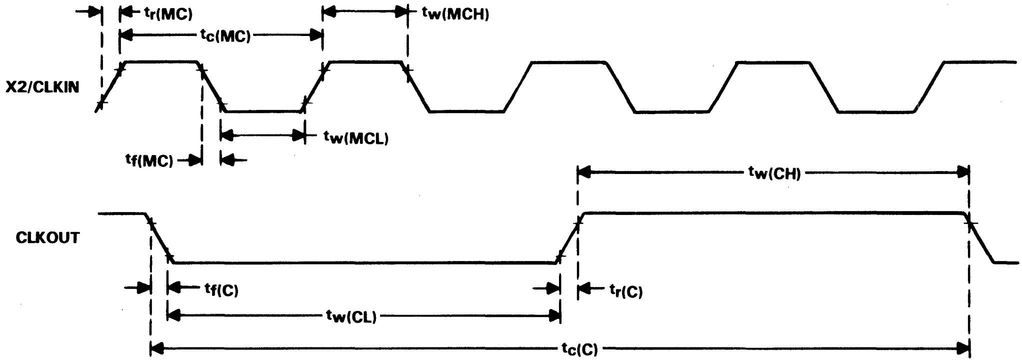


FIGURE 4. ASYNCHRONOUS INPUT SYNCHRONIZATION CIRCUITS

clock timing



NOTE 1: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

MEMORY AND PERIPHERAL INTERFACE TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d1}	Delay time CLKOUT↓ to address bus valid (see Note 2)	R_L = 825 Ω, C_L = 100 pF, See Figure 2	10†		60	ns
t_{d2}	Delay time CLKOUT↓ to $\overline{\text{MEN}}$↓		¼$t_{c(C)}$ − 10†		¼$t_{c(C)}$ + 15	ns
t_{d3}	Delay time CLKOUT↓ to $\overline{\text{MEN}}$↑		−15†		15	ns
t_{d4}	Delay time CLKOUT↓ to $\overline{\text{DEN}}$↓		¼$t_{c(C)}$ − 10†		¼$t_{c(C)}$ + 15	ns
t_{d5}	Delay time CLKOUT↓ to $\overline{\text{DEN}}$↑		−15†		15	ns
t_{d6}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↓		½$t_{c(C)}$ − 10†		½$t_{c(C)}$ + 15	ns
t_{d7}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↑		−10†		15	ns
t_{d8}	Delay time CLKOUT↓ OUT valid to data bus				¼$t_{c(C)}$ + 65	ns
t_{d9}	Time after CLKOUT↓ that data bus starts to be driven		¼$t_{c(C)}$ − 10†			ns
t_{d10}	Time after CLKOUT↓ that data bus stops being driven				¼$t_{c(C)}$ + 30†	ns
t_v	Data bus OUT valid after CLKOUT↓		¼$t_{c(C)}$ − 10			ns
$t_{h(A\text{-}WMD)}$	Address bus hold time after $\overline{\text{WE}}$↑, $\overline{\text{MEN}}$↑, or $\overline{\text{DEN}}$↑		0†			ns
$t_{su(A\text{-}MD)}$	Address bus setup prior to $\overline{\text{MEN}}$↓ and $\overline{\text{DEN}}$↓		¼$t_{c(C)}$ − 45†			ns

NOTE 2: Address bus will be valid upon $\overline{\text{WE}}$↑, $\overline{\text{DEN}}$↑, or $\overline{\text{MEN}}$↑, and address bus will be valid upon $\overline{\text{MEN}}$↓ or $\overline{\text{DEN}}$↓.

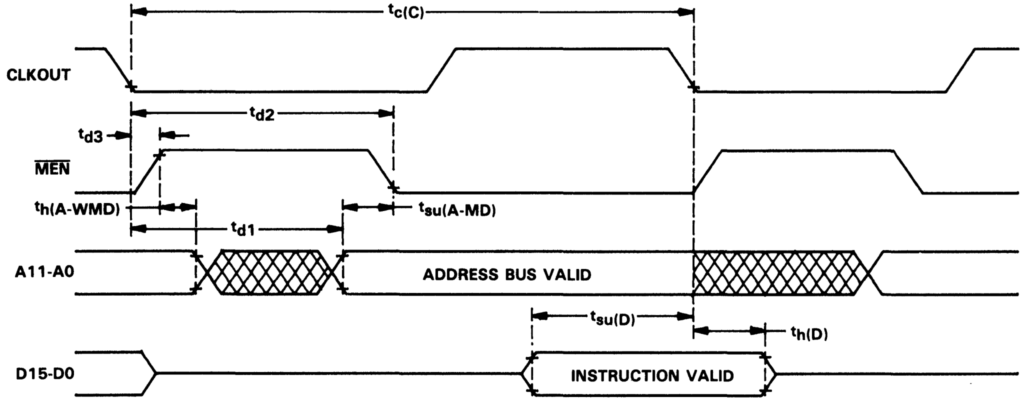
†These values were derived from characterization data. The values are guaranteed but not tested.

timing requirements over recommended operating conditions

		TEST CONDITIONS	MIN	NOM	MAX	UNIT
$t_{su(D)}$	Setup time data bus valid prior to CLKOUT↓	R_L = 825 Ω, C_L = 100 pF, See Figure 2	50			ns
$t_{h(D)}$	Hold time data bus held valid after CLKOUT↓ (see Note 3)		0			ns

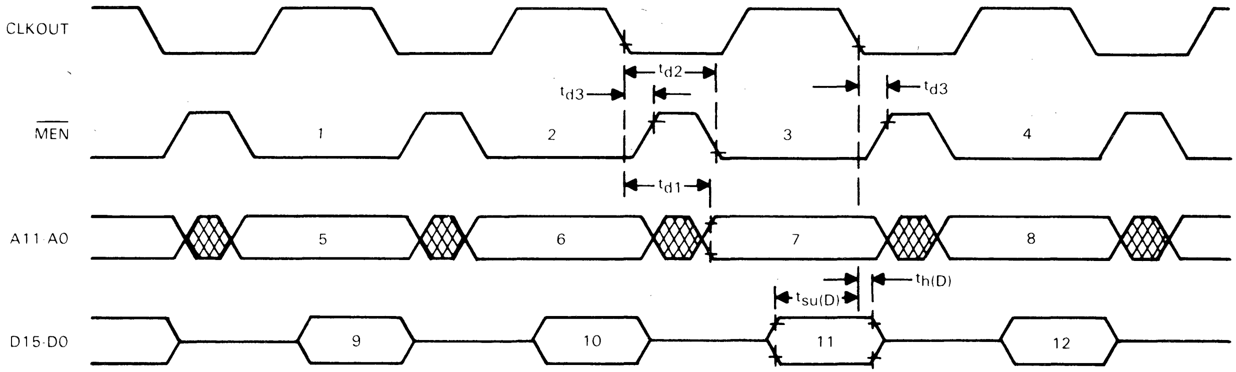
NOTE 3: Data may be removed from the data bus upon $\overline{\text{MEN}}$↑ or $\overline{\text{DEN}}$↑ preceding CLKOUT↓.

memory read



NOTE 1: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

TBLR instruction timing

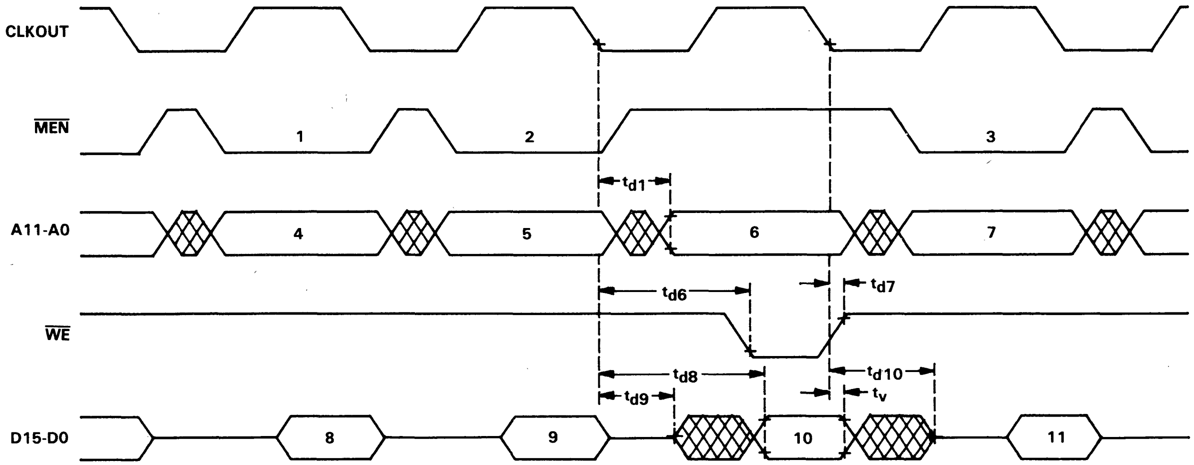


LEGEND:

1. TBLR INSTRUCTION PREFETCH
2. DUMMY PREFETCH
3. DATA FETCH
4. NEXT INSTRUCTION PREFETCH
5. ADDRESS BUS VALID
6. ADDRESS BUS VALID
7. ADDRESS BUS VALID
8. ADDRESS BUS VALID
9. INSTRUCTION VALID
10. INSTRUCTION VALID
11. DATA INPUT VALID
12. INSTRUCTION VALID

NOTE 1: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

TBLW instruction timing

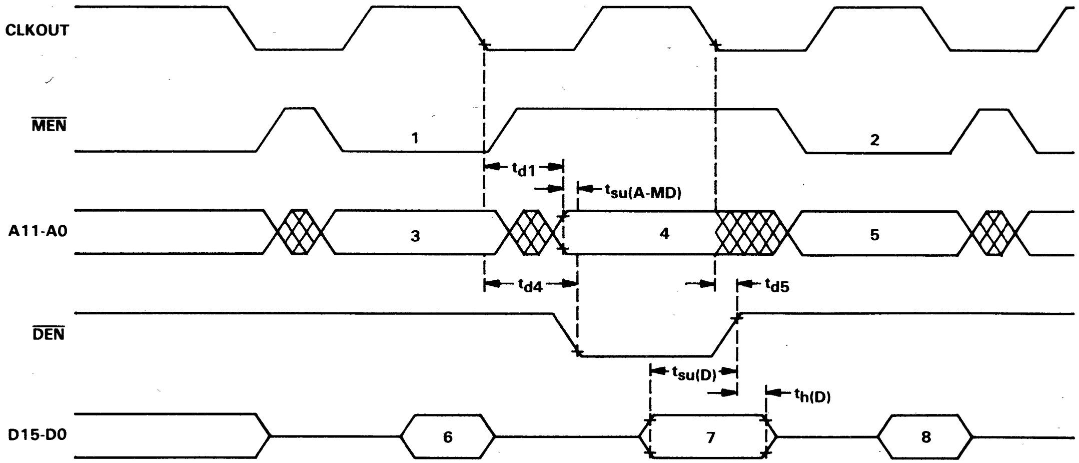


LEGEND:

1. TBLW INSTRUCTION PREFETCH
2. DUMMY PREFETCH
3. NEXT INSTRUCTION PREFETCH
4. ADDRESS BUS VALID
5. ADDRESS BUS VALID
6. ADDRESS BUS VALID
7. ADDRESS BUS VALID
8. INSTRUCTION VALID
9. INSTRUCTION VALID
10. DATA OUTPUT VALID
11. INSTRUCTION VALID

NOTE 1: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

IN instruction timing

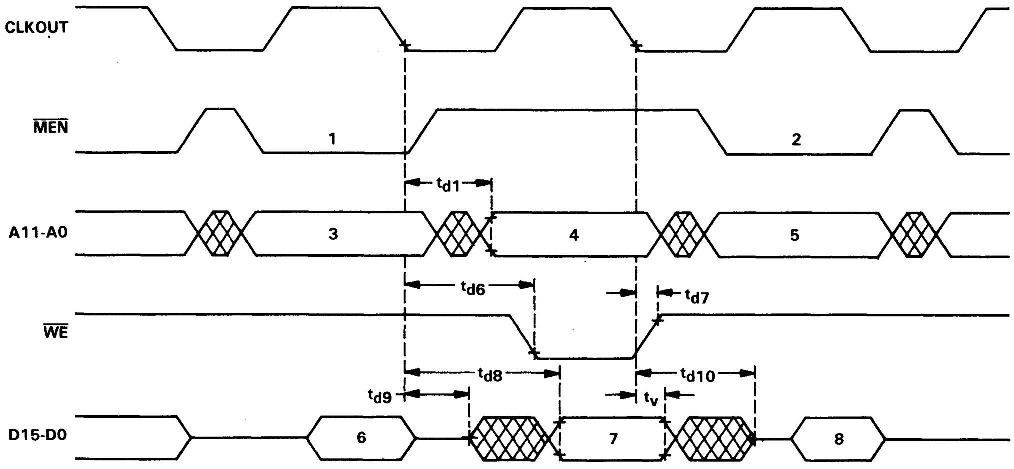


LEGEND:

1. IN INSTRUCTION PREFETCH
2. NEXT INSTRUCTION PREFETCH
3. ADDRESS BUS VALID
4. PERIPHERAL BUS VALID
5. ADDRESS BUS VALID
6. INSTRUCTION VALID
7. DATA INPUT VALID
8. INSTRUCTION VALID

NOTE 1: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

OUT instruction timing



LEGEND:

1. OUT INSTRUCTION PREFETCH
2. NEXT INSTRUCTION PREFETCH
3. ADDRESS BUS VALID
4. PERIPHERAL ADDRESS VALID
5. ADDRESS BUS VALID
6. INSTRUCTION VALID
7. DATA OUTPUT VALID
8. INSTRUCTION VALID

NOTE 1: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

RESET ($\overline{RS}$) TIMING

timing requirements over recommended operating conditions

		MIN	NOM	MAX	UNIT
$t_{su(R)}$	Reset ($\overline{RS}$) setup time prior to CLKOUT (see Note 4)	50			ns
$t_{w(R)}$	$\overline{RS}$ pulse duration	$5t_{c(C)}$			ns

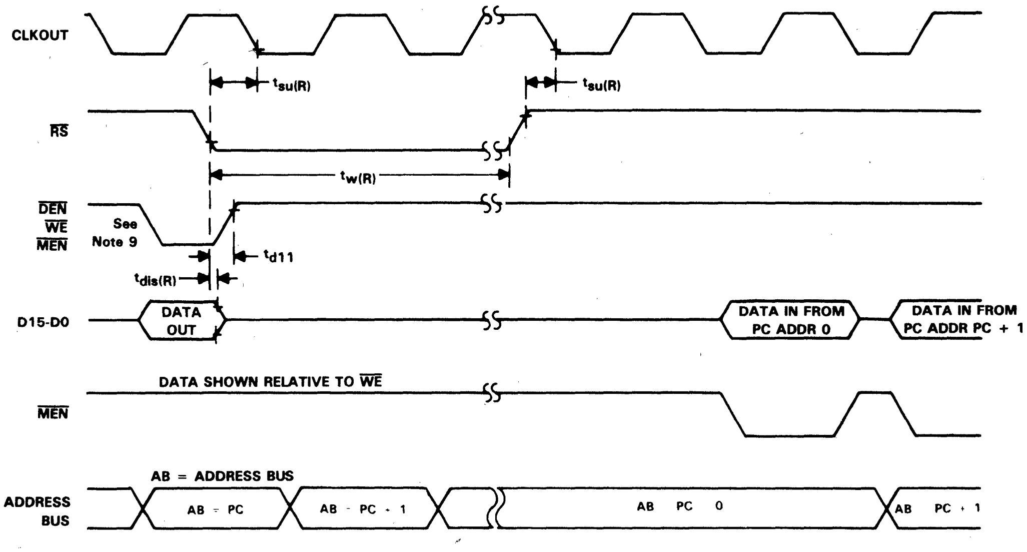
switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d11}	Delay time $\overline{DEN}$↑, $\overline{WE}$↑, and $\overline{MEN}$↑ from $\overline{RS}$	R_L = 825 Ω, C_L = 100 pF, See Figure 2			$½t_{c(C)}+50^†$	ns
$t_{dis(R)}$	Data bus disable time after $\overline{RS}$				$¼t_{c(C)}+50^†$	ns

NOTE 4: $\overline{RS}$ can occur anytime during a clock cycle. Time given is minimum to ensure synchronous operation.

†These values were derived from characterization data. The values are guaranteed but not tested.

reset timing



NOTES:
1. Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.
5. $\overline{RS}$ forces $\overline{DEN}$, $\overline{WE}$, and $\overline{MEN}$ high and three-states data bus D0 through D15. AB outputs (and program counter) are synchronously cleared to zero after the next complete CLK cycle from ↓$\overline{RS}$.
6. $\overline{RS}$ must be maintained for a minimum of five clock cycles.
7. Resumption of normal program will commence after one complete CLK cycle from ↑$\overline{RS}$.
8. Due to the synchronizing action on $\overline{RS}$, time to execute the function can vary dependent upon when ↑$\overline{RS}$ or ↓$\overline{RS}$ occur in the CLK cycle.
9. Diagram shown is for definition purpose only. $\overline{DEN}$, $\overline{WE}$, $\overline{MEN}$ are mutually exclusive.
10. During a write cycle, $\overline{RS}$ may produce an invalid write address.

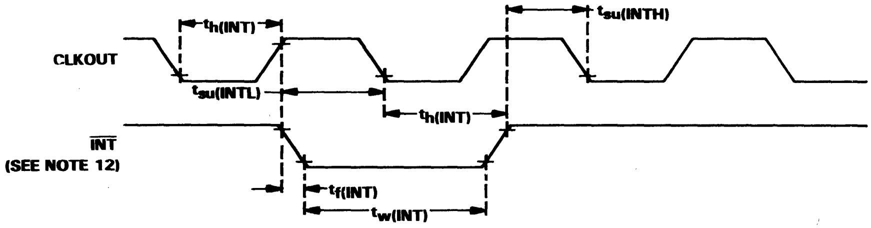
INTERRUPT ($\overline{\text{INT}}$) TIMING

timing requirements over recommended operating conditions

PARAMETER		MIN	TYP	MAX	UNIT
$t_{f(INT)}$	Fall time $\overline{\text{INT}}$ (see Note 11)		10		ns
$t_{w(INT)}$	Pulse duration $\overline{\text{INT}}$	$t_{c(C)}$			ns
$t_{su(INTL)}$	Setup time $\overline{\text{INT}}$↓ before CLKOUT↓	50			ns
$t_{su(INTH)}$	Setup time $\overline{\text{INT}}$↑ before CLKOUT↓	50†			ns
$t_{h(INT)}$	Hold time $\overline{\text{INT}}$↑ or $\overline{\text{INT}}$↓ after CLKOUT↓	50†			ns

†Value derived from characterization data. The value is guaranteed but not tested.

interrupt timing



NOTES: 1. Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.
11. $\overline{\text{INT}}$ fall time must be less than 15 ns.
12. The interrupt signal, $\overline{\text{INT}}$, must not transition within ± 50 ns from CLKOUT↓.

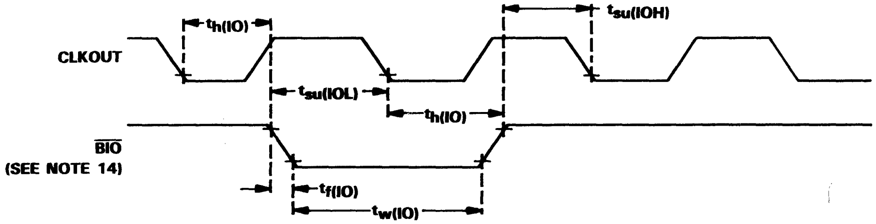
I/O ($\overline{\text{BIO}}$) TIMING

timing requirements over recommended operating conditions

PARAMETER		MIN	TYP	MAX	UNIT
$t_{f(IO)}$	Fall time $\overline{\text{BIO}}$ (see Note 13)		10		ns
$t_{w(IO)}$	Pulse duration $\overline{\text{BIO}}$	$t_{c(C)}$			ns
$t_{su(IOL)}$	Setup time $\overline{\text{BIO}}$↓ before CLKOUT↓	50			ns
$t_{su(IOH)}$	Setup time $\overline{\text{BIO}}$↑ before CLKOUT↓	50†			ns
$t_{h(IO)}$	Hold time BIO↑ or BIO↓ after CLKOUT↓	50†			ns

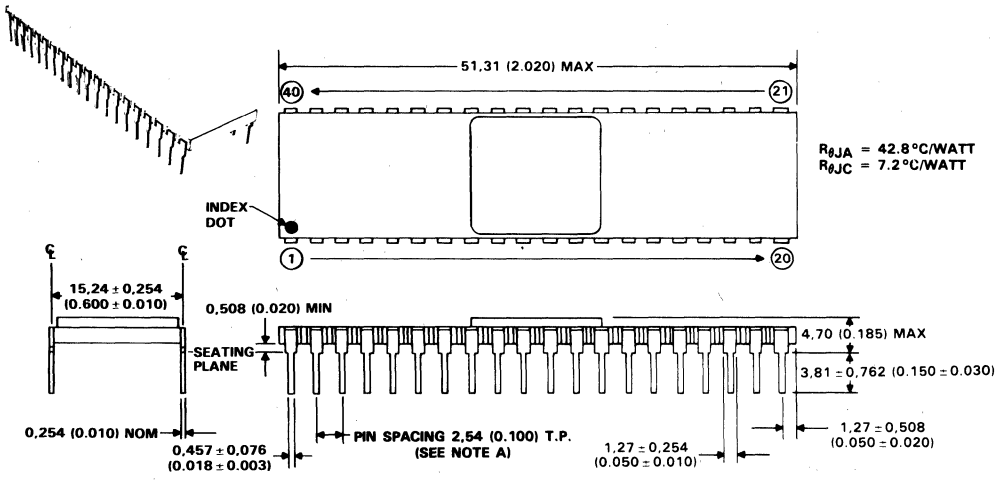
†Value derived from characterization data and not tested.

$\overline{\text{BIO}}$ timing



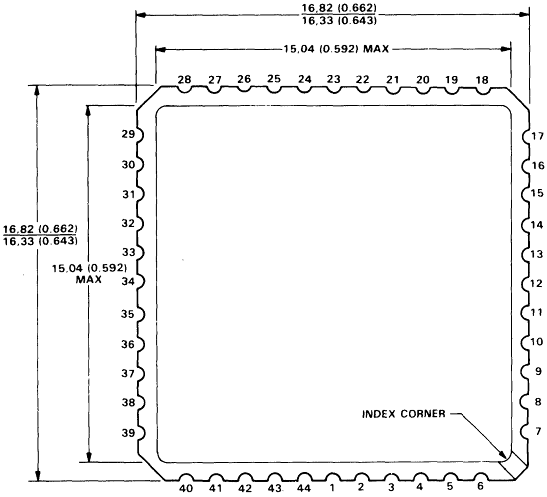
NOTES: 1. Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.
13. $\overline{\text{BIO}}$ fall time must be less than 15 ns.
14. The branch control signal, $\overline{\text{BIO}}$, must not transition within ± 50 ns from CLKOUT↓.

40-pin JD ceramic dual-in-line package

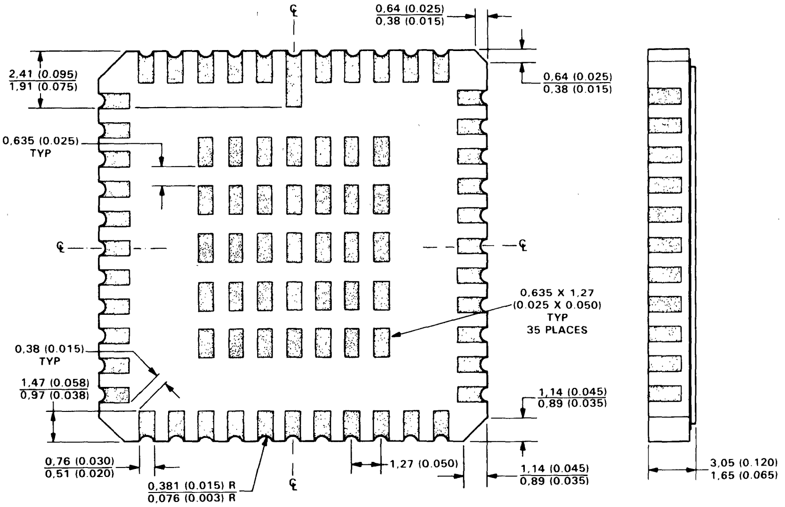


NOTE A: Each pin centerline is located within 0,254 (0.010) of its true longitudinal position.
ALL DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES

44-pad ceramic chip carrier package



$R_{\theta JA}$ = 44.7°C/WATT
$R_{\theta JC}$ = 13.3°C/WATT



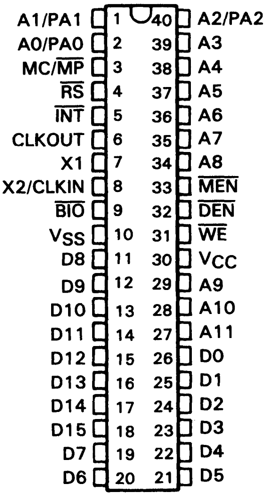
The checkerboard pattern is aligned vertically and is symmetrical horizontally as shown.
ALL DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES
Texas Instruments reserves the right to make changes at any time in order to improve design and to supply the best product possible.

SMJ320C10
DIGITAL SIGNAL PROCESSOR

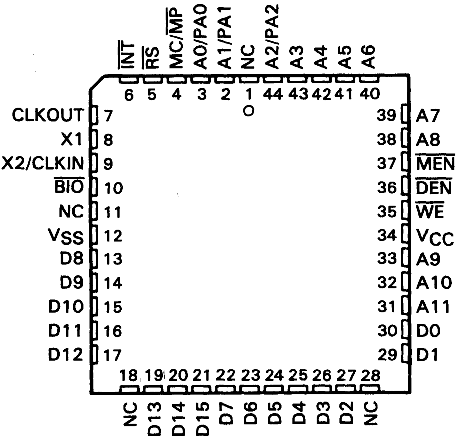
MAY 1987 — REVISED MAY 1989

- 200-ns Instruction Cycle
- 144-Word On-Chip Data RAM
- ROMless Version — SMJ320C10
- 1.5K-Word On-Chip Program ROM — SMJ320M10
- External Memory Expansion to a Total of 4K Words at Full Speed
- 16-Bit Instruction/Data Word
- 32-Bit ALU/Accumulator
- 16 × 16-Bit Multiply in One Instruction Cycle
- 0 to 16-Bit Barrel Shifter
- Eight Input and Eight Output Channels
- 16-Bit Bidirectional Data Bus with 40-Megabits-per-Second Transfer Rate
- Interrupt with Full Context Save
- Signed Two's-Complement Fixed-Point Arithmetic
- CMOS Technology
- Single 5-V Supply

SMJ320C10 . . . JD PACKAGE
(TOP VIEW)



SMJ320C10 . . . FD PACKAGE
(TOP VIEW)

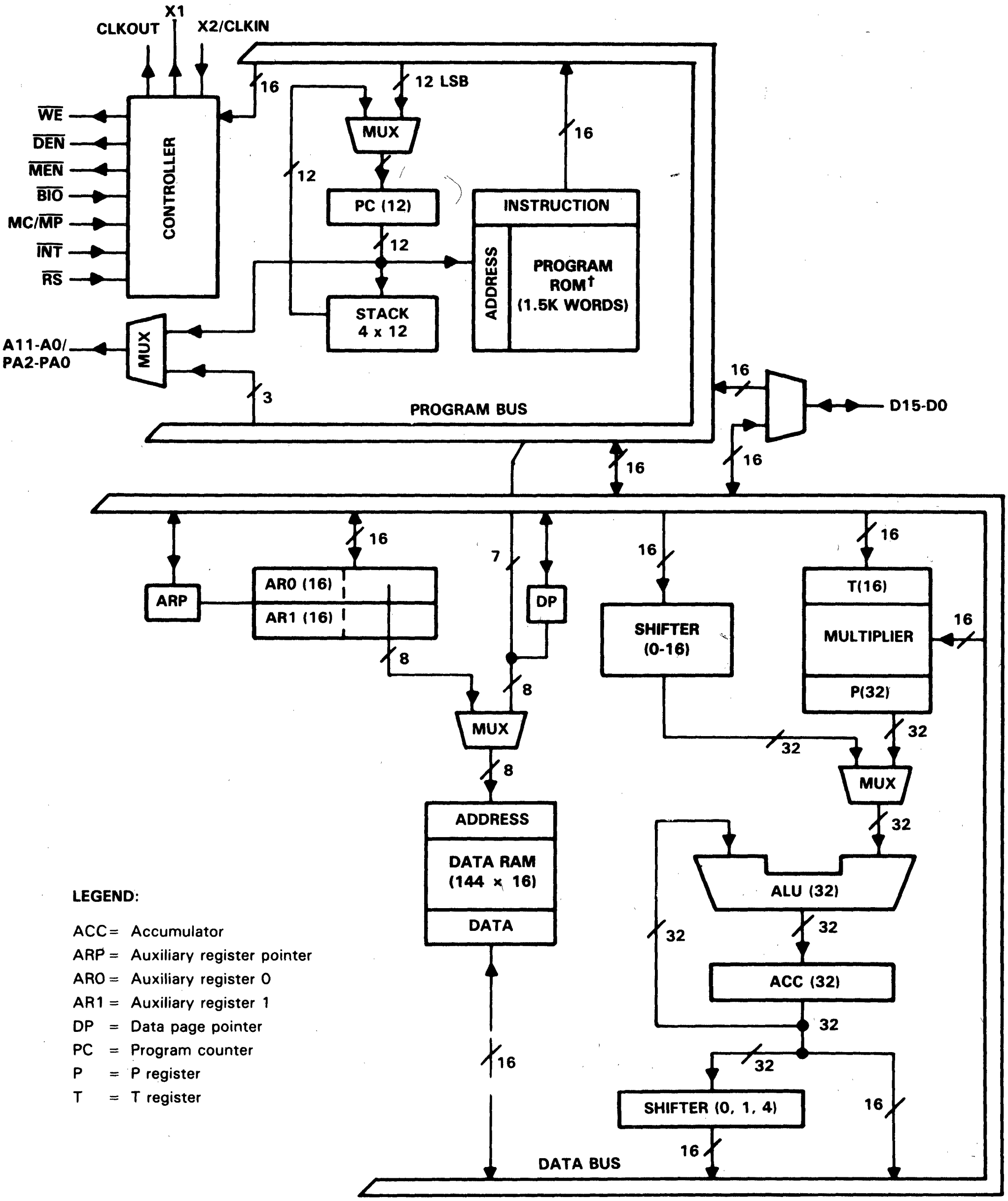


description

The SMJ320C10 is the first low-power CMOS member of the Texas Instruments SMJ320 family of Digital Signal Processors. This device is a CMOS pin-for-pin compatible version of the industry-standard TMS32010 Digital Signal Processor. The 165-mW typical power dissipation of the SMJ320C10 enables power-sensitive applications to take advantage of the SMJ320C10's high performance. The 16/32-bit microcomputer was designed to support a wide range of high-speed and numeric-intensive applications. The SMJ320C10 combines the flexibility of a high-speed controller with the numerical capability of an array processor, thereby offering an inexpensive alternative to multichip bit-slice processors. The highly pipelined architecture and efficient instruction set of the SMJ320C10 provide the capability of executing more than five million instructions per second. The instruction set is easily programmed and contains general-purpose as well as digital signal processing instructions.

TEXAS INSTRUMENTS
POST OFFICE BOX 1443 • HOUSTON, TEXAS 77001

functional block diagram



LEGEND:

ACC	=	Accumulator
ARP	=	Auxiliary register pointer
AR0	=	Auxiliary register 0
AR1	=	Auxiliary register 1
DP	=	Data page pointer
PC	=	Program counter
P	=	P register
T	=	T register

†The Program ROM is available in SMJ320M10 only.

PIN NOMENCLATURE

NAME	I/O	DEFINITION
A11-A0/PA2-PA0	O	External address bus. I/O port address multiplexed over PA2-PA0.
$\overline{\text{BIO}}$	I	External polling input
CLKOUT	O	System clock output, ¼ crystal/CLKIN frequency
D15-D0	I/O	16-bit parallel data bus
$\overline{\text{DEN}}$	O	Data enable for device input data on D15-D0
$\overline{\text{INT}}$	I	External interrupt input
MC/$\overline{\text{MP}}$	I	Memory mode select pin. High selects microcomputer mode. Low selects microprocessor mode.
$\overline{\text{MEN}}$	O	Memory enable indicates that D15-D0 will accept external memory instruction.
NC	O	No connection
$\overline{\text{RS}}$	I	Reset for initializing the device
V_{CC}	I	+5 V supply
V_{SS}	I	Ground
$\overline{\text{WE}}$	O	Write enable for device output data on D15-D0
X1	O	Crystal output for internal oscillator
X2/CLKIN	I	Crystal input for internal oscillator or external system clock input

architecture

The SMJ320 family utilizes a modified Harvard architecture for speed and flexibility. In a strict Harvard architecture, program and data memory lie in two separate spaces, permitting a full overlap of instruction fetch and execution. The SMJ320 family's modification of the Harvard architecture allows transfers between program and data spaces, thereby increasing the flexibility of the device. This modification permits coefficients stored in program memory to be read into the RAM, eliminating the need for a separate coefficient ROM. It also makes available immediate instructions and subroutines based on computed values.

32-bit ALU/accumulator

The SMJ320 first-generation devices contain a 32-bit ALU and accumulator for support of double-precision, two's-complement arithmetic. The ALU is a general-purpose arithmetic unit that operates on 16-bit words taken from the data RAM or derived from immediate instructions. In addition to the usual arithmetic instructions, the ALU can perform Boolean operations, providing the bit manipulation ability required of a high-speed controller. The accumulator stores the output from the ALU and is often an input to the ALU. It operates with a 32-bit wordlength. The accumulator is divided into a high-order word (bits 31 through 16) and a low-order word (bits 15 through 0). Instructions are provided for storing the high- and low-order accumulator words in memory.

shifters

Two shifters are available for manipulating data. The ALU barrel shifter performs a left-shift of 0 to 16 places on data memory words loaded into the ALU. This shifter extends the high-order bit of the data word and zero-fills the low-order bits for two's-complement arithmetic. The accumulator parallel shifter performs a left-shift of 0, 1, or 4 places on the entire accumulator and places the resulting high-order accumulator bits into data RAM. Both shifters are useful for scaling and bit extraction.

16 × 16-bit parallel multiplier

The multiplier performs a 16 x 16-bit two's-complement multiplication with a 32-bit result in a single instruction cycle. The multiplier consists of three units: the T Register, P Register, and multiplier array. The 16-bit T Register temporarily stores the multiplicand; the P Register stores the 32-bit product. Multiplier values either come from the data memory or are derived immediately from the MPYK (multiply immediate) instruction word. The fast on-chip multiplier allows the device to perform fundamental operations such as convolution, correlation, and filtering.

data and program memory

Since the SMJ320 devices use a Harvard architecture, data and program memory reside in two separate spaces. The SMJ320C10 device has 144 words of on-chip data RAM and 1.5K words of on-chip program ROM. The SMJ320C10 is capable of executing up to 4K words of external memory at full speed for those applications requiring external program memory space. This allows for external RAM-based systems to provide multiple functionality.

microcomputer/microprocessor operating modes

The SMJ320C10 offers two modes of operation defined by the state of the MC/$\overline{\text{MP}}$ pin: the microcomputer mode (MC/$\overline{\text{MP}}$ = 1) or the microprocessor mode (MC/$\overline{\text{MP}}$ = 0). In the microcomputer mode, the on-chip ROM is mapped into the memory space with up to 1.5K words of internal memory and 2.5K words of external memory. In the microprocessor mode, 4K words of memory are external.

interrupts and subroutines

The SMJ320 first-generation devices contain a four-level hardware stack for saving the contents of the program counter during interrupts and subroutine calls. Instructions are available for saving the device's complete context. PUSH and POP instructions permit a level of nesting restricted only by the amount of available RAM. The interrupts used in these devices are maskable.

input/output

The 16-bit parallel data bus can be utilized to perform I/O functions in two cycles. The I/O ports are addressed by the three LSBs on the address lines. In addition, a polling input for bit test and jump operations ($\overline{\text{BIO}}$) and an interrupt pin ($\overline{\text{INT}}$) have been incorporated for multitasking.

instruction set

A comprehensive instruction set supports both numeric-intensive operations, such as signal processing, and general-purpose operations, such as high-speed control. All of the first-generation devices are object-code compatible and use the same 60 instructions. The instruction set consists primarily of single-cycle single-word instructions, permitting execution rates of more than six million instructions per second. Only infrequently used branch and I/O instructions are multicycle. Instructions that shift data as part of an arithmetic operation execute in a single cycle and are useful for scaling data in parallel with other operations.

Three main addressing modes are available with the instruction set: direct, indirect, and immediate addressing.

direct addressing

In direct addressing, seven bits of the instruction word concatenated with the 1-bit data page pointer form the data memory address. This implements a paging scheme in which the first page contains 128 words, and the second page contains up to 16 words.

indirect addressing

Indirect addressing forms the data memory address from the least-significant eight bits of one of the two auxiliary registers, AR0 and AR1. The Auxiliary Register Pointer (ARP) selects the current auxiliary register. The auxiliary registers can be automatically incremented or decremented and the ARP changed in parallel with the execution of any indirect instruction to permit single-cycle manipulation of data tables. Indirect addressing can be used with all instructions requiring data operands, except for the immediate operand instructions.

immediate addressing

Immediate instructions derive data from part of the instruction word rather than from the data RAM. Some useful immediate instructions are multiply immediate (MPYK), load accumulator immediate (LACK), and load auxiliary register immediate (LARK).

instruction set summary

Table 1 lists the symbols and abbreviations used in Table 2, the instruction set summary. Table 2 contains a short description and the opcode for each SMJ320 first-generation instruction. The summary is arranged according to function and alphabetized within each functional group.

TABLE 1. INSTRUCTION SYMBOLS

SYMBOL	MEANING
ACC	Accumulator
D	Data memory address field
I	Addressing mode bit
K	Immediate operand field
PA	3-bit port address field
R	1-bit operand field specifying auxiliary register
S	4-bit left-shift code
X	3-bit accumulator left-shift field

TABLE 2. SMJ320 FIRST-GENERATION INSTRUCTION SET SUMMARY

ACCUMULATOR INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABS	Absolute value of accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0
ADD	Add to accumulator with shift	1	1	0	0	0	0	←S→				I	←D→						
ADDH	Add to high-order accumulator bits	1	1	0	1	1	0	0	0	0	0	I	←D→						
ADDS	Add to accumulator with no sign extension	1	1	0	1	1	0	0	0	0	1	I	←D→						
AND	AND with accumulator	1	1	0	1	1	1	1	0	0	1	I	←D→						
LAC	Load accumulator with shift	1	1	0	0	1	0	←S→				I	←D→						
LACK	Load accumulator immediate	1	1	0	1	1	1	1	1	1	0	←K→							
OR	OR with accumulator	1	1	0	1	1	1	1	0	1	0	I	←D→						
SACH	Store high-order accumulator bits with shift	1	1	0	1	0	1	1	←X→			I	←D→						
SACL	Store low-order accumulator bits	1	1	0	1	0	1	0	0	0	0	I	←D→						
SUB	Subtract from accumulator with shift	1	1	0	0	0	1	←S→				I	←D→						
SUBC	Conditional subtract (for divide)	1	1	0	1	1	0	0	1	0	0	I	←D→						
SUBH	Subtract from high-order accumulator bits	1	1	0	1	1	0	0	0	1	0	I	←D→						
SUBS	Subtract from accumulator with no sign extension	1	1	0	1	1	0	0	0	1	1	I	←D→						
XOR	Exclusive OR with accumulator	1	1	0	1	1	1	1	0	0	0	I	←D→						
ZAC	Zero accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1
ZALH	Zero accumulator and load high-order bits	1	1	0	1	1	0	0	1	0	1	I	←D→						
ZALS	Zero accumulator and load low-order bits with no sign extension	1	1	0	1	1	0	0	1	1	0	I	←D→						

AUXILIARY REGISTER AND DATA PAGE POINTER INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LAR	Load auxiliary register	1	1	0	0	1	1	1	0	0	R	I	←D→						
LARK	Load auxiliary register immediate	1	1	0	1	1	1	0	0	0	R	←K→							
LARP	Load auxiliary register pointer immediate	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	K
LDP	Load data memory page pointer	1	1	0	1	1	0	1	1	1	1	I	←D→						
LDPK	Load data memory page pointer immediate	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	K
MAR	Modify auxiliary register and pointer	1	1	0	1	1	0	1	0	0	0	I	←D→						
SAR	Store auxiliary register	1	1	0	0	1	1	0	0	0	R	I	←D→						

TABLE 2. SMJ320 FIRST-GENERATION INSTRUCTION SET SUMMARY (CONTINUED)

BRANCH INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B	Branch unconditionally	2	2	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BANZ	Branch on auxiliary register not zero	2	2	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BGEZ	Branch if accumulator ≥ 0	2	2	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BGZ	Branch if accumulator > 0	2	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BIOZ	Branch on $\overline{\text{BIO}}$ = 0	2	2	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BLEZ	Branch if accumulator ≤ 0	2	2	1	1	1	1	1	0	1	1	C	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BLZ	Branch if accumulator < 0	2	2	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BNZ	Branch if accumulator ≠ 0	2	2	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BV	Branch on overflow	2	2	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BZ	Branch if accumulator = 0	2	2	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
CALA	Call subroutine from accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0
CALL	Call subroutine immediately	2	2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
RET	Return from subroutine or interrupt routine	2	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	1

T REGISTER, P REGISTER, AND MULTIPLY INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APAC	Add P register to accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1
LT	Load T register	1	1	0	1	1	0	1	0	1	0	I	◄— D —►						
LTA	LTA combines LT and APAC into one instruction	1	1	0	1	1	0	1	1	0	0	I	◄— D —►						
LTD	LTD combines LT, APAC, and DMOV into one instruction	1	1	0	1	1	0	1	0	1	1	I	◄— D —►						
MPY	Multiply with T register, store product in P register	1	1	0	1	1	0	1	1	0	1	I	◄— D —►						
MPYK	Multiply T register with immediate operand; store product in P register	1	1	1	0	0	◄— K —►												
PAC	Load accumulator from P register	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	0
SPAC	Subtract P register from accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0

TABLE 2. SMJ320 FIRST-GENERATION INSTRUCTION SET SUMMARY (CONCLUDED)

CONTROL INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DINT	Disable interrupt	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1
EINT	Enable interrupt	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0
LST	Load status register	1	1	0	1	1	1	1	0	1	1	I	◄— D —►						
NOP	No operation	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
POP	POP stack to accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1
PUSH	PUSH stack from accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0
ROVM	Reset overflow mode	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	0
SOVM	Set overflow mode	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1
SST	Store status register	1	1	0	1	1	1	1	1	0	0	I	◄— D —►						

I/O AND DATA MEMORY OPERATIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMOV	Copy contents of data memory location into next higher location	1	1	0	1	1	0	1	0	0	1	I	◄— D —►						
IN	Input data from port	2	1	0	1	0	0	0	◄PA►			I	◄— D —►						
OUT	Output data to port	2	1	0	1	0	0	1	◄PA►			I	◄— D —►						
TBLR	Table read from program memory to data RAM	3	1	0	1	1	0	0	1	1	1	I	◄— D —►						
TBLW	Table write from data RAM to program memory	3	1	0	1	1	1	1	1	0	1	I	◄— D —►						

development support products

Together, Texas Instruments and its authorized third-party suppliers offer an extensive line of development support products to assist the user in all aspects of TMS320 first-generation-based design and development. These products range from development and application software to complete hardware development and evaluation systems. Table 4 lists the development support products for the first-generation TMS320 devices.

System development may begin with the use of the simulator, evaluation module (EVM), or emulator (XDS), along with an assembler/linker. These tools give the TMS320 user various means of evaluation, from software simulation of the first-generation TMS320s (simulator) to full-speed in-circuit emulation with hardware and software break point tracing and timing capabilities (XDS).

Software and hardware can be developed simultaneously by using the macro assembler/linker or simulator for software development, the XDS for hardware development, and the evaluation module for both software development and limited hardware development.

Many third-party vendors offer additional development support for the first-generation TMS320s, including assembler/linkers, simulators, high-level languages, applications software, algorithm development tools, application boards, software development boards, and in-circuit emulators. Refer to the *TMS320 Family Development Support Reference Guide* (SPRU011A) for further information about TMS320 development support products offered by both Texas Instruments and its third-party suppliers.

Additional support for the TMS320 products consists of an extensive library of product and applications documentation. Three-day DSP design workshops are offered by the TI Regional Technology Centers (RTCs). These workshops provide insight into the architecture and the instruction set of the first-generation TMS320s as well as hands-on training with the TMS320 development tools. When technical questions arise in regard to a TMS320 member, contact Texas Instruments TMS320 Hotline at (713) 274-2320. Or, keep informed on the latest TI and third-party development support tools by accessing the libraries of application source code via the DSP Bulletin Board Service (BBS) at (713) 274-2323. The BBS provides access for the 2400-/1200-/300-bps modems.

documentation support

Extensive documentation supports the first-generation TMS320 devices from product announcement through applications development. The types of documentation include data sheets with design specifications, complete user's guides, and 750 pages of application reports published in the book *Digital Signal Processing Applications with the TMS320 Family.*

A series of DSP textbooks is being published to support digital signal processing research and education. The first book, *DFT/FFT and Convolution Algorithms,* is now available. The TMS320 newsletter, *Details on Signal Processing,* is being published quarterly and distributed to update TMS320 customers on product information. The TMS320 DSP bulletin board service provides access to large amounts of information pertaining to the TMS320 family.

Refer to the *TMS320 Family Development Support Reference Guide* for further information about TMS320 documentation. To receive copies of first-generation SMJ320 literature, call the Customer Response Center at 1-800-232-3200.

TABLE 3. TMS320 FIRST-GENERATION SOFTWARE AND HARDWARE SUPPORT

SOFTWARE TOOLS	PART NUMBER
Macro Assembler/Linker	
PC/MS-DOS	TMDS3242850-02
VAX/VMS	TMDS3242250-08
VAX ULTRIX	TMDS3242260-08
SUN-3 UNIX	TMDS3242550-08
Simulator	
PC/MS-DOS	TMDS3240811-02
VAX/VMS	TMDS3240211-08
Digital Filter Design Package (DFDP)	
IBM PC PC-DOS	DFDP/IBM002
DSP Software Library	
PC/MS-DOS	TMDC3240812-12
VAX/VMS	TMDC3240212-18
TMS320 Bell 212A Modem Software	
PC/MS-DOS	TMDX3240813-12
Data Encryption Standard Software	
PC/MS-DOS	TMDX3240814-12
HARDWARE TOOLS	**PART NUMBER**
Evaluation Tools	
Evaluation Module (EVM)	RTC/EVM320A-03
Analog Interface Board 1 (AIB1)	RTC/EVM320C-06
Analog Interface Board 2 (AIB2)	RTC/AIB320A-06
EPROM DSP Starter Kit (TMS320E15)	RTC/EVM320E-15
XDS/22 Emulators	
TMS320C10/C15	TMDS3262211
TMS320C14	TMDX3262214
TMS320C17	TMDX3262217
XDS/22 Upgrade Kits	
TMS32010 → TMS320C10/C15	TMDS3282215
TMS320C10/C15 → TMS320C14	TMDX3285010 and TMDX3285018
TMS320C10/C15 → TMS320C17	TMDX3285014 and TMDX3285018
EPROM Programming Adaptor Sockets	
40- to 28-pin (TMS320E15/E17)	RTC/PGM320A-06
44- to 28-pin (TMS320E15/E17)	RTC/PGM320C-06
68- to 28-pin (TMS320E14)	TMDX3270110
Additional Target Connector	
44-pin PLCC (TMS320C10)	TMDX3288810

absolute maximum ratings over specified temperature range (unless otherwise noted)[†]

Supply voltage range, V_{CC}[‡]	−0.3 V to 7 V
Input voltage range	−0.3 V to 7 V
Output voltage range	−0.3 V to 7 V
Continuous power dissipation:	0.33 W
Maximum operating case temperature	125°C
Minimum operating free-air temperature	−55°C
Storage temperature range	−65°C to +150°C

[†] Stresses beyond 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 beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

[‡] All voltage values are with respect to V_{SS}.

recommended operating conditions

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage		4.5	5	5.5	V
V_{SS}	Supply voltage			0		V
V_{IH}	High-level input voltage	All inputs except CLKIN	2			V
		CLKIN	3			
V_{IL}	Low-level input voltage (All inputs)				0.8	V
I_{OH}	High-level output current (All outputs)				−300	μA
I_{OL}	Low-level output current (All outputs)				2	mA
T_A	Operating free-air temperature		−55			°C
T_C	Operating case temperature				125	°C

electrical characteristics over specified temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS		MIN	TYP[§]	MAX	UNIT
V_{OH}	High-level output voltage		I_{OH} = MAX		2.4	3		V
			I_{OH} = 20 μA (see Note 1)		V_{CC}−0.4 V[#]			
V_{OL}	Low-level output voltage		I_{OL} = MAX			0.3	0.5	V
I_{OZ}	Off-state output current		V_{CC} = MAX	V_O = 2.4 V			20	μA
				V_O = 0.4 V			−20	
I_I	Input current		V_I = V_{SS}, V_{CC} = MAX				±50	μA
I_{CC}[¶]	Supply current	SMJ320C10	V_{CC} = 5.5 V, f_x = MAX				60	mA
		SMJ320C10-14	V_{CC} = 5.5 V, f_x = MAX				50	
C_i	Input capacitance	Data bus	f = 1 MHz, All other pins 0 V			25[#]		pF
		All others				15[#]		
C_o	Output capacitance	Data bus				25[#]		
		All others				10[#]		

[§] All typical values except for I_{CC} are at V_{CC} = 5 V, T_A = 25°C.

[¶] I_{CC} characteristics are inversely proportional to temperature; i.e., I_{CC} decreases approximately linearly with temperature.

[#] Value derived from characterization data and is guaranteed to limit but not tested.

NOTE 1: This voltage specification is included for interface to HC logic. However, note that all other timing parameters defined in this data sheet are specified for TTL logic levels and will differ for HC logic levels.

CLOCK CHARACTERISTICS AND TIMING

The SMJ320C10 can use either its internal oscillator or an external frequency source for a clock.

internal clock option

The internal oscillator is enabled by connecting a crystal across X1 and X2/CLKIN (see Figure 1). The frequency of CLKOUT is one-fourth the crystal fundamental frequency. The crystal should be fundamental mode, and parallel resonant, with an effective series resistance of 30 ohms, a power dissipation of 1 mW, and be specified at a load capacitance of 20 pF.

PARAMETER		TEST CONDITIONS	MIN	NOM	MAX	UNIT
Crystal frequency, f_x[¶]	SMJ320C10	−55°C to 125°C	6.7[#]		20.5[#]	MHz
	SMJ320C10-14		6.7[#]		20.5[#]	
C1, C2				10		pF

[¶] An 8 MHz crystal was used in the test.
[#] Value derived from characterization data and is guaranteed but not tested.

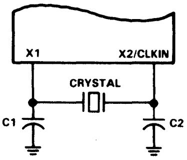


FIGURE 1. INTERNAL CLOCK OPTION

external clock option

An external frequency source can be used by injecting the frequency directly into X2/CLKIN with X1 left unconnected. The external frequency injected must conform to the specifications listed in the table below.

timing requirements over recommended operating conditions

			MIN	NOM	MAX	UNIT
$t_{c(MC)}$	Master clock cycle time	SMJ320C10	48.78		150	ns
		SMJ320C10-14	69.44		150	
$t_{r(MC)}$	Rise time master clock input (see Note 2)			5		ns
$t_{f(MC)}$	Fall time master clock input (see Note 2)			5		ns
$t_{w(MCP)}$	Pulse duration master clock[#]		$0.475t_{c(MC)}$		$0.525t_{c(MC)}$	ns
$t_{w(MCL)}$	Pulse duration master clock low, $t_{c(MC)}$ = 50 ns			20		ns
$t_{w(MCH)}$	Pulse duration master clock high, $t_{c(MC)}$ = 50 ns			20		ns

switching characteristics over recommended operating conditions

PARAMETER			TEST CONDITIONS	MIN	NOM	MAX	UNIT
$t_{c(C)}$	CLKOUT cycle time[‡]	SMJ320C10	R_L = 825 Ω, C_L = 100 pF, See Figure 2	195.12		600	ns
		SMJ320C10-14		278.78		600	
$t_{r(C)}$	CLKOUT rise time				10		ns
$t_{f(C)}$	CLKOUT fall time				8		ns
$t_{w(CL)}$	Pulse duration, CLKOUT low				92		ns
$t_{w(CH)}$	Pulse duration, CLKOUT high				90		ns
$t_{d(MCC)}$	Delay time CLKIN↑ to CLKOUT↓			20		60	ns

[‡] $t_{c(C)}$ is the cycle time of CLKOUT, i.e., $4*t_{c(MC)}$ (4 times CLKIN cycle time if an external oscillator is used).
[#] Value derived from characterization data and is guaranteed but not tested.
NOTE 2: CLKIN rise and fall times must be less than 10 ns.

MEMORY AND PERIPHERAL INTERFACE TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d1}	Delay time CLKOUT↓ to address bus valid (see Note 3)	R_L = 825 Ω, C_L = 100 pF, See Figure 2	10†		50	ns
t_{d2}	Delay time CLKOUT↓ to $\overline{\text{MEN}}$↓		$\frac{1}{4}t_{c(C)} - 5$†		$\frac{1}{4}t_{c(C)} + 15$	ns
t_{d3}	Delay time CLKOUT↓ to $\overline{\text{MEN}}$↑		-10†		15	ns
t_{d4}	Delay time CLKOUT↓ to $\overline{\text{DEN}}$↓		$\frac{1}{4}t_{c(C)} - 5$†		$\frac{1}{4}t_{c(C)} + 15$	ns
t_{d5}	Delay time CLKOUT↓ to $\overline{\text{DEN}}$↑		-10†		15	ns
t_{d6}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↓		$\frac{1}{2}t_{c(C)} - 5$†		$\frac{1}{2}t_{c(C)} + 15$	ns
t_{d7}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↑		-10†		15	ns
t_{d8}	Delay time CLKOUT↓ OUT valid to data bus				$\frac{1}{4}t_{c(C)} + 65$	ns
t_{d9}	Time after CLKOUT↓ that data bus starts to be driven		$\frac{1}{4}t_{c(C)} - 5$†			ns
t_{d10}	Time after CLKOUT↓ that data bus stops being driven				$\frac{1}{4}t_{c(C)} + 40$†	ns
t_v	Data bus OUT valid after CLKOUT↓		$\frac{1}{4}t_{c(C)} - 10$			ns
$t_{su(A\text{-}MD)}$	Address bus setup time prior to $\overline{\text{MEN}}$↓ or $\overline{\text{DEN}}$↓		$\frac{1}{4}t_{c(C)} - 45$			ns
$t_{h(A\text{-}WMD)}$	Address hold time after $\overline{\text{WE}}$↑, $\overline{\text{MEN}}$↑, or $\overline{\text{DEN}}$↑		0†			ns

NOTE 3: Address bus will be valid upon $\overline{\text{WE}}$↑, $\overline{\text{DEN}}$↑, or $\overline{\text{MEN}}$↑.

†Value derived from characterization data and is guaranteed but not tested.

timing requirements over recommended operating conditions

		TEST CONDITIONS	MIN	NOM	MAX	UNIT
$t_{su(D)}$	Setup time data bus valid prior to CLKOUT↓	R_L = 825 Ω, C_L = 100 pF, See Figure 2	50			ns
$t_{h(D)}$	Hold time data bus held valid after CLKOUT↓ (see Note 4)		0			ns

NOTE 4: Data may be removed from the data bus upon $\overline{\text{MEN}}$↑ or $\overline{\text{DEN}}$↑ preceding CLKOUT↓.

RESET ($\overline{\text{RS}}$) TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d11}	Delay time $\overline{\text{DEN}}$↑, $\overline{\text{WE}}$↑, and $\overline{\text{MEN}}$↑ from $\overline{\text{RS}}$	R_L = 825 Ω, C_L = 100 pF, See Figure 2			½$t_{c(C)}$+50†	ns
$t_{dis(R)}$	Data bus disable time after $\overline{\text{RS}}$				¼$t_{c(C)}$+50†	ns

†Value derived from characterization data and is guaranteed but not tested.

timing requirements over recommended operating conditions

		MIN	NOM	MAX	UNIT
$t_{su(R)}$	Reset ($\overline{\text{RS}}$) setup time prior to CLKOUT (see Note 5)	50			ns
$t_{w(R)}$	$\overline{\text{RS}}$ pulse duration	5$t_{c(C)}$			ns

NOTE 5: $\overline{\text{RS}}$ can occur anytime during a clock cycle. Time given is minimum to ensure synchronous operation.

INTERRUPT ($\overline{\text{INT}}$) TIMING

timing requirements over recommended operating conditions

		MIN	NOM	MAX	UNIT
$t_{f(INT)}$	Fall time $\overline{\text{INT}}$ (see Note 6)		10		ns
$t_{w(INT)}$	Pulse duration $\overline{\text{INT}}$	$t_{c(C)}$			ns
$t_{su(INT)}$	Setup time $\overline{\text{INT}}$↓ before CLKOUT↓	50			ns

NOTE 6: $\overline{\text{INT}}$ fall time must be less than 15 ns.

$\overline{\text{BIO}}$ TIMING

timing requirements over recommended operating conditions

		MIN	NOM	MAX	UNIT
$t_{f(IO)}$	Fall time $\overline{\text{BIO}}$ (see Note 7)		10		ns
$t_{w(IO)}$	Pulse duration $\overline{\text{BIO}}$	$t_{c(C)}$			ns
$t_{su(IO)}$	Setup time $\overline{\text{BIO}}$↓ before CLKOUT↓	50			ns

NOTE 7: $\overline{\text{BIO}}$ fall time must be less than 15 ns.

PARAMETER MEASUREMENT INFORMATION

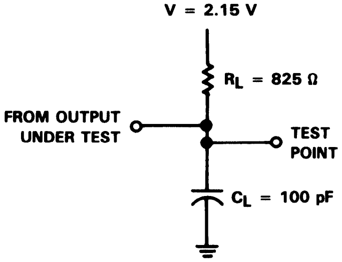
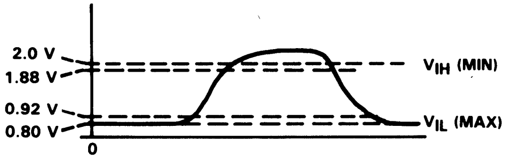
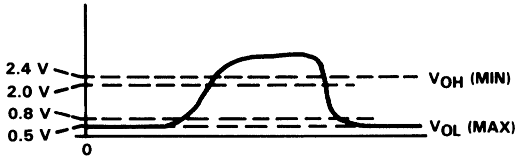


FIGURE 2. TEST LOAD CIRCUIT



(a) INPUT



(b) OUTPUTS

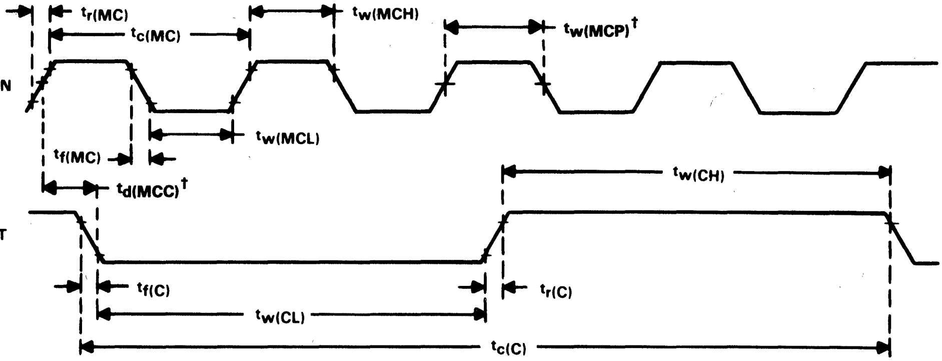
FIGURE 3. VOLTAGE REFERENCE LEVELS

TIMING DIAGRAMS

This section contains all the timing diagrams for the SMJ320 first-generation devices. Refer to the top corner for the specific device.

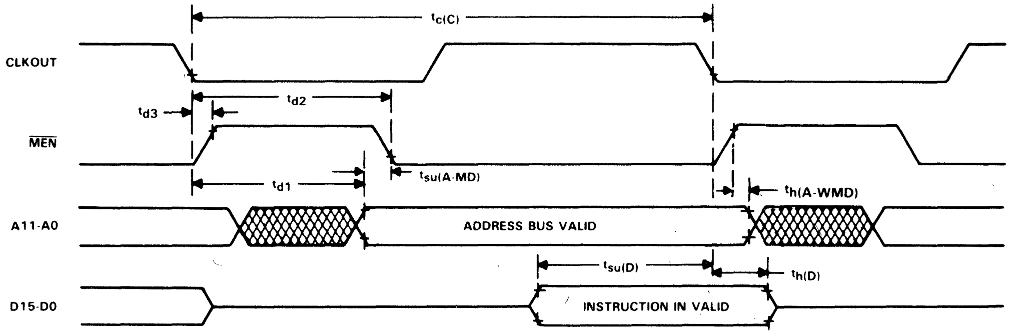
Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

clock timing

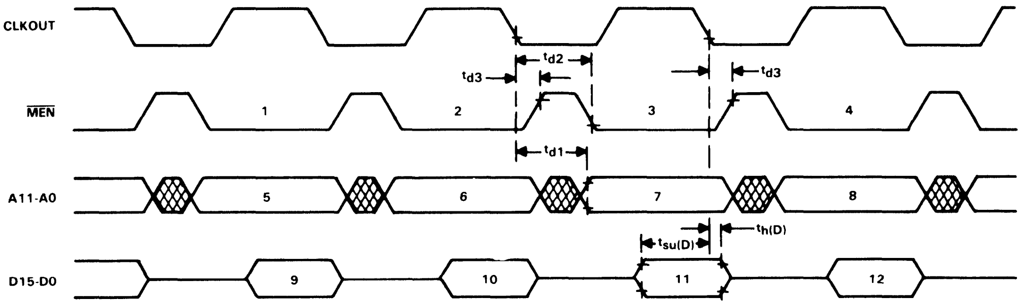


†$t_{d(MCC)}$ and $t_{w(MCP)}$ are referenced to an intermediate level of 1.5 volts on the CLKIN waveform.

memory read timing



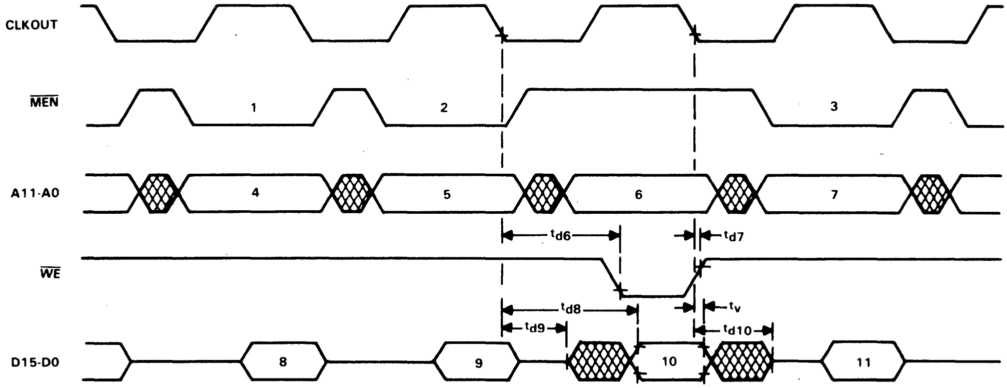
TBLR instruction timing



LEGEND:

1. TBLR INSTRUCTION PREFETCH
2. DUMMY PREFETCH
3. DATA FETCH
4. NEXT INSTRUCTION PREFETCH
5. ADDRESS BUS VALID
6. ADDRESS BUS VALID
7. ADDRESS BUS VALID
8. ADDRESS BUS VALID
9. INSTRUCTION VALID
10. INSTRUCTION VALID
11. DATA INPUT VALID
12. INSTRUCTION VALID

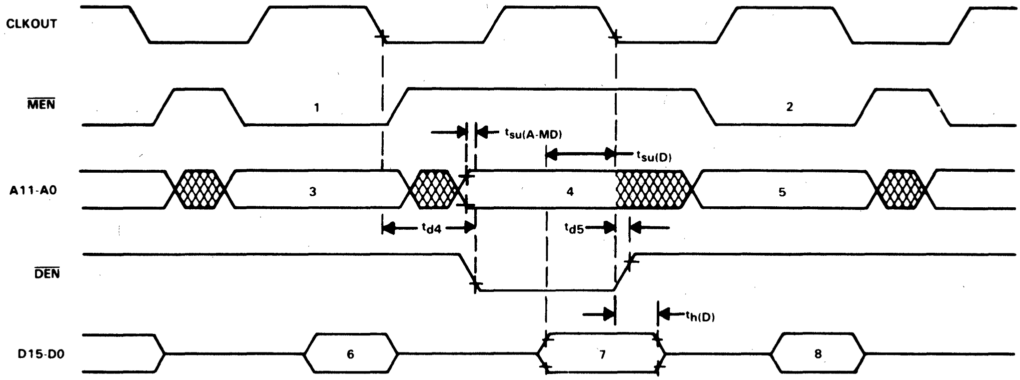
TBLW instruction timing



LEGEND:

1. TBLW INSTRUCTION PREFETCH
2. DUMMY PREFETCH
3. NEXT INSTRUCTION PREFETCH
4. ADDRESS BUS VALID
5. ADDRESS BUS VALID
6. ADDRESS BUS VALID
7. ADDRESS BUS VALID
8. INSTRUCTION VALID
9. INSTRUCTION VALID
10. DATA OUTPUT VALID
11. INSTRUCTION VALID

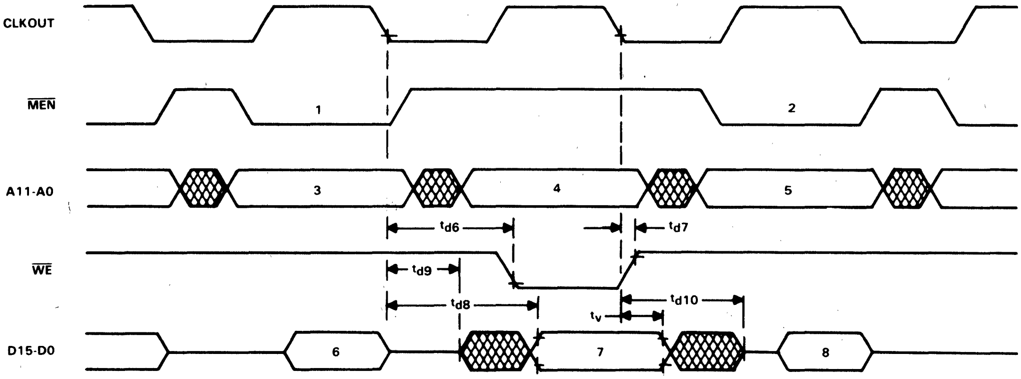
IN instruction timing



LEGEND:

1. IN INSTRUCTION PREFETCH
2. NEXT INSTRUCTION PREFETCH
3. ADDRESS BUS VALID
4. PERIPHERAL ADDRESS VALID
5. ADDRESS BUS VALID
6. INSTRUCTION VALID
7. DATA INPUT VALID
8. INSTRUCTION VALID

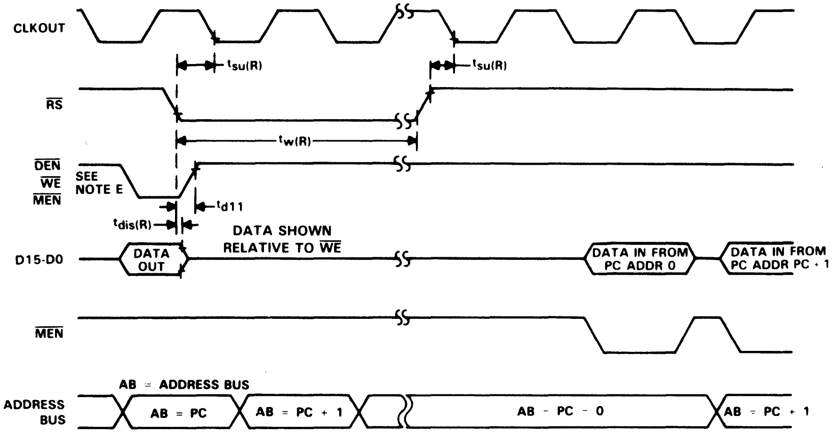
OUT instruction timing



LEGEND:

1. OUT INSTRUCTION PREFETCH
2. NEXT INSTRUCTION PREFETCH
3. ADDRESS BUS VALID
4. PERIPHERAL ADDRESS VALID
5. ADDRESS BUS VALID
6. INSTRUCTION VALID
7. DATA OUTPUT VALID
8. INSTRUCTION VALID

reset timing



NOTES: A. RS forces DEN, WE, and MEN high and places data bus D0 through D15 in a high-impedance state. AB outputs (and program counter) are synchronously cleared to zero after the next complete CLK cycle from RS↓.

B. RS must be maintained for a minimum of five clock cycles.

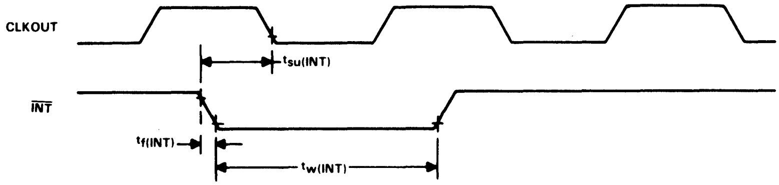
C. Resumption of normal program will commence after one complete CLK cycle from RS↑.

D. Due to the synchronizing action on RS, time to execute the function can vary dependent upon when RS↑ or RS↓ occur in the CLK cycle.

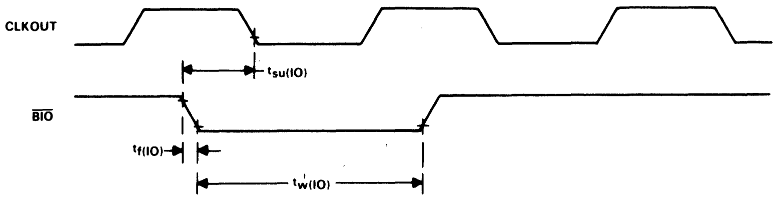
E. Diagram shown is for definition purpose only. DEN, WE, and MEN are mutually exclusive.

F. During a write cycle, RS may produce an invalid write address.

interrupt timing

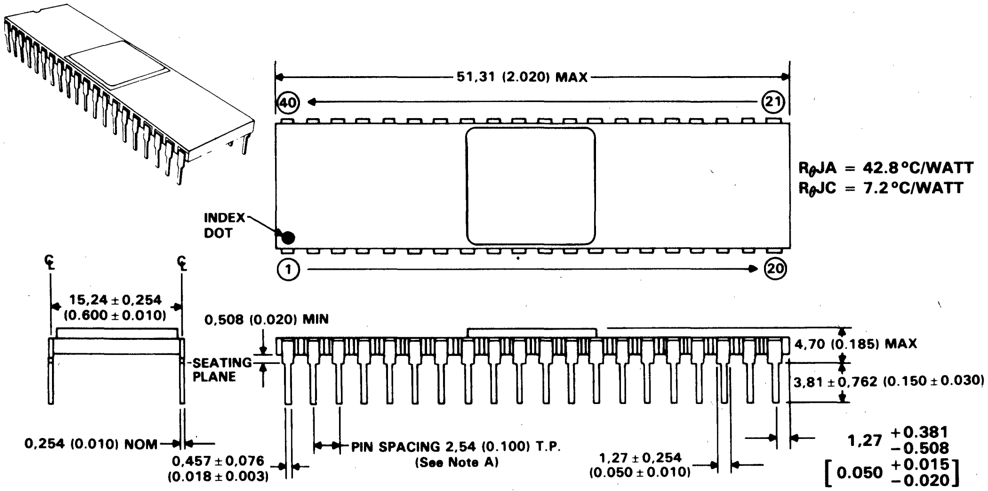


BIO timing



MECHANICAL DATA

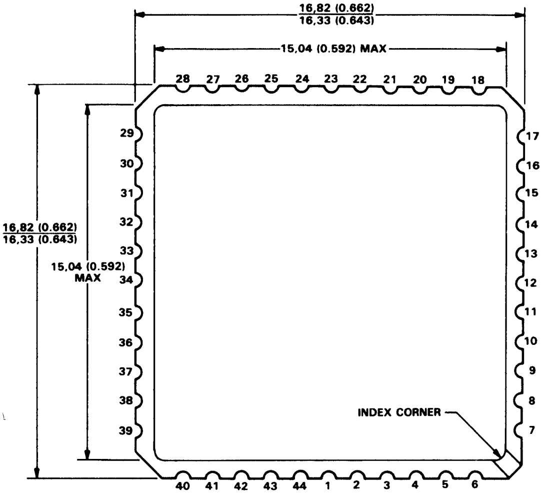
40-pin JD ceramic dual-in-line package



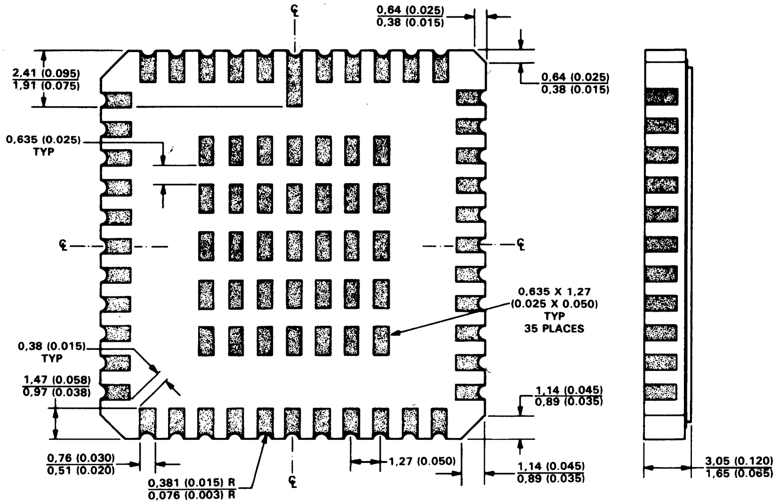
NOTE A: Each pin centerline is located within 0,254 (0.010) of its true longitudinal position.
ALL DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES

MECHANICAL DATA

44-pad ceramic chip carrier package



$R_{\theta}JA$ = 44.7°C/WATT
$R_{\theta}JC$ = 13.3°C/WATT



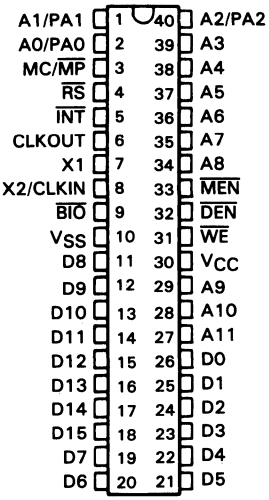
The checkerboard pattern is aligned vertically and is symmetrical horizontally as shown.
ALL DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES
Texas Instruments reserves the right to make changes at any time in order to improve design and to supply the best product possible.

SMJ320C15, SMJ320C15-25 DIGITAL SIGNAL PROCESSORS

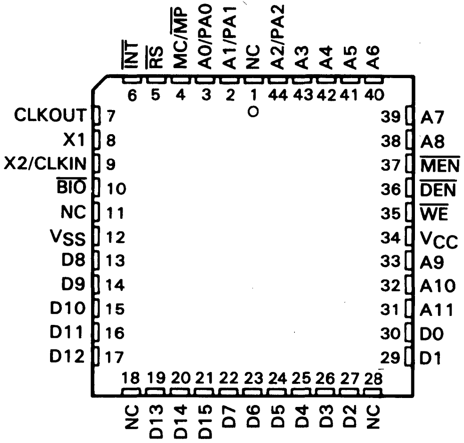
MAY 1989

- Instruction Cycle Timing:
  - 160 ns (SMJ320C15-25)
  - 200 ns (SMJ320C15)
- 256 Words of On-Chip Data RAM
- 4K Words of On-Chip Program ROM
- External Memory Expansion up to 4K Words at Full Speed
- 16-Bit Instruction/Data Word
- Object-Code and Pin-For-Pin Compatible with SMJ32010 and SMJ320C10
- 32-Bit ALU/Accumulator
- 16 × 16-Bit Multiplier with 32-Bit Product
- 0 to 16-Bit Barrel Shifter
- Eight Input and Eight Output Channels
- On-Chip Clock Oscillator
- Single 5-V ± 10% Supply
- Device Packaging:
  - 40-Pin DIP
  - 44-Pad LCCC
- CMOS Technology

JD PACKAGE
(TOP VIEW)



FD PACKAGE
(TOP VIEW)



description

The SMJ320 family of 16/32-bit single-chip digital signal processors combines the flexibility of a high-speed controller with the numerical capability of an array processor, thereby offering an inexpensive alternative to multichip bit-slice processors. The highly paralleled architecture and efficient instruction set provide speed and flexibility to produce a MOS microprocessor family capable of executing 6.4 MIPS (million instructions per second). The SMJ320 family optimizes speed by implementing functions in hardware that other processors implement through microcode or software. This hardware-intensive approach provides the design engineer with processing power previously unavailable on a single chip.

The SMJ320 family consists of two generations of digital signal processors. The first generation contains the SMJ32010, SMJ320C10, and the SMJ320C15. The SMJ32020 and SMJ320C25 are the second-generation processors, designed for higher performance. Many features are common among the SMJ320 processors. Specific features are added in each processor to provide different cost/performance tradeoffs. Each processor has software and hardware tools to facilitate rapid design.

TEXAS INSTRUMENTS
POST OFFICE BOX 1443 • HOUSTON, TEXAS 77001

PIN NOMENCLATURE

NAME	I/O	DEFINITION
A11-A0/PA2-PA0	O	External address bus. I/O port address multiplexed over PA2-PA0.
$\overline{\text{BIO}}$	I	External polling input
CLKOUT	O	System clock output, ¼ crystal/CLKIN frequency
D15-D0	I/O	16-bit parallel data bus
$\overline{\text{DEN}}$	O	Data enable for device input data on D15-D0
$\overline{\text{INT}}$	I	External interrupt input
MC/$\overline{\text{MP}}$	I	Memory mode select pin. High selects microcomputer mode. Low selects microprocessor mode.
$\overline{\text{MEN}}$	O	Memory enable indicates that D15-D0 will accept external memory instruction.
NC	O	No connection
$\overline{\text{RS}}$	I	Reset for initializing the device
V_{CC}	I	+5 V supply
V_{SS}	I	Ground
$\overline{\text{WE}}$	O	Write enable for device output data on D15-D0
X1	O	Crystal output for internal oscillator
X2/CLKIN	I	Crystal input for internal oscillator or external system clock input

description (continued)

The SMJ320C15 CMOS device is object-code and pin-for-pin compatible with the SMJ32010/C10 and offers expanded on-chip RAM of 256 words and on-chip program ROM of 4K words. This device allows the capability of upgrading performance and reducing power, board space, and system cost without hardware redesign. The SMJ320C15 is also available in a 160-ns version, the SMJ320C15-25.

architecture

The SMJ320 family utilizes a modified Harvard architecture for speed and flexibility. In a strict Harvard architecture, program and data memory lie in two separate spaces, permitting a full overlap of instruction fetch and execution. The SMJ320 family's modification of the Harvard architecture allows transfers between program and data spaces, thereby increasing the flexibility of the device. This modification permits coefficients stored in program memory to be read into the RAM, eliminating the need for a separate coefficient ROM. It also makes available immediate instructions and subroutines based on computed values.

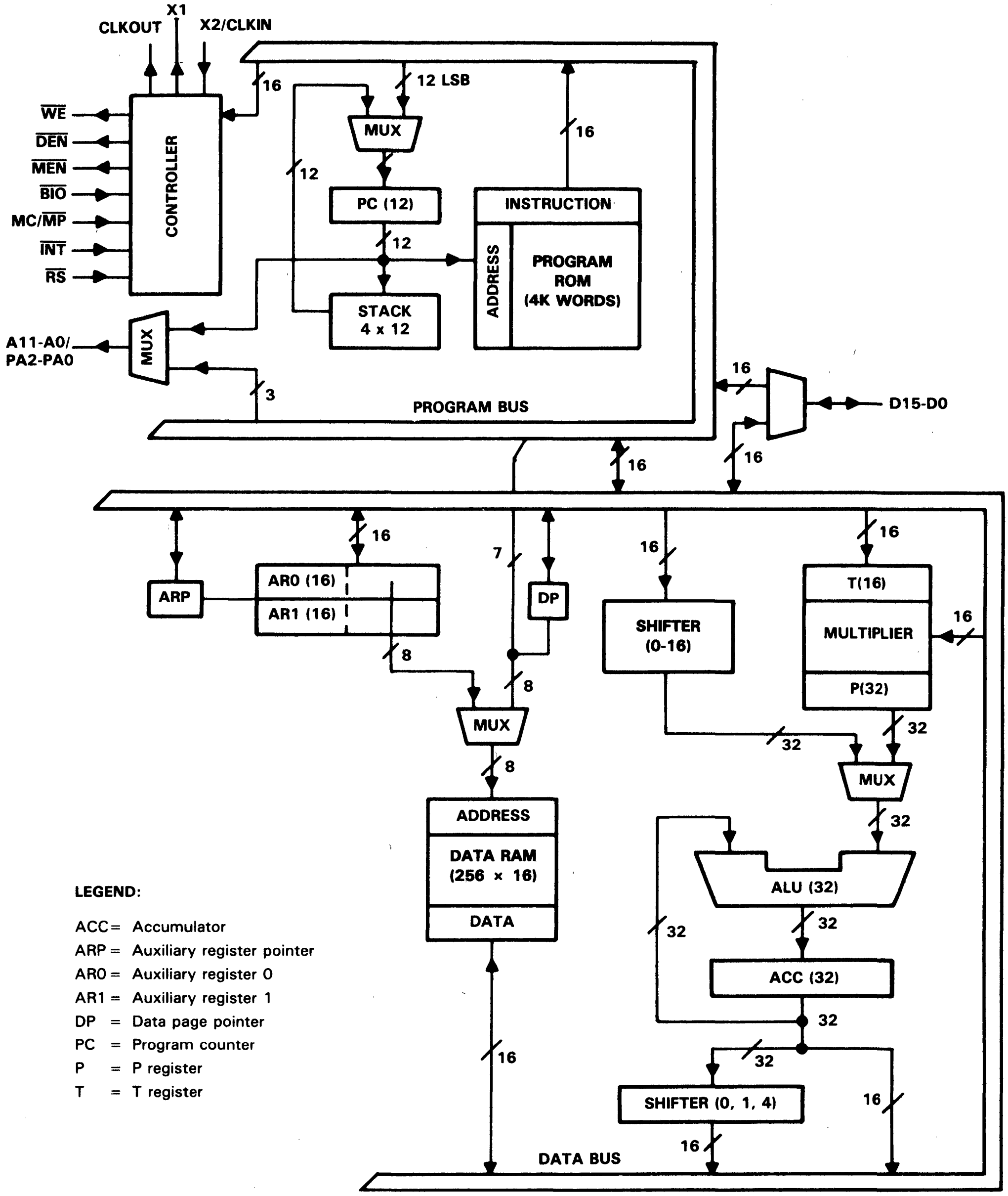
32-bit ALU/accumulator

The SMJ320 first-generation devices contain a 32-bit ALU and accumulator for support of double-precision, two's-complement arithmetic. The ALU is a general-purpose arithmetic unit that operates on 16-bit words taken from the data RAM or derived from immediate instructions. In addition to the usual arithmetic instructions, the ALU can perform Boolean operations, providing the bit manipulation ability required of a high-speed controller. The accumulator stores the output from the ALU and is often an input to the ALU. It operates with a 32-bit wordlength. The accumulator is divided into a high-order word (bits 31 through 16) and a low-order word (bits 15 through 0). Instructions are provided for storing the high- and low-order accumulator words in memory.

shifters

Two shifters are available for manipulating data. The ALU barrel shifter performs a left-shift of 0 to 16 places on data memory words loaded into the ALU. This shifter extends the high-order bit of the data word and zero-fills the low-order bits for two's-complement arithmetic. The accumulator parallel shifter performs a left-shift of 0, 1, or 4 places on the entire accumulator and places the resulting high-order accumulator bits into data RAM. Both shifters are useful for scaling and bit extraction.

functional block diagram



LEGEND:

ACC =	Accumulator
ARP =	Auxiliary register pointer
AR0 =	Auxiliary register 0
AR1 =	Auxiliary register 1
DP =	Data page pointer
PC =	Program counter
P =	P register
T =	T register

16 × 16-bit parallel multiplier

The multiplier performs a 16 x 16-bit two's-complement multiplication with a 32-bit result in a single instruction cycle. The multiplier consists of three units: the T Register, P Register, and multiplier array. The 16-bit T Register temporarily stores the multiplicand; the P Register stores the 32-bit product. Multiplier values either come from the data memory or are derived immediately from the MPYK (multiply immediate) instruction word. The fast on-chip multiplier allows the device to perform fundamental operations such as convolution, correlation, and filtering.

data and program memory

Since the SMJ320 devices use a Harvard architecture, data and program memory reside in two separate spaces. The first-generation devices have 144 or 256 words of on-chip data RAM and 1.5K or 4K words of on-chip program ROM. The SMJ320C15 is capable of executing up to 4K words of external memory at full speed for those applications requiring external program memory space. This allows for external RAM-based systems to provide multiple functionality.

microcomputer/microprocessor operating modes

The SMJ320C15 offers two modes of operation defined by the state of the MC/$\overline{\text{MP}}$ pin: the microcomputer mode (MC/$\overline{\text{MP}}$ = 1) or the microprocessor mode (MC/$\overline{\text{MP}}$ = 0). In the microcomputer mode, on-chip ROM is mapped into the memory space with up to 4K words of internal memory available. In the microprocessor mode, all 4K words of memory are external.

interrupts and subroutines

The SMJ320 first-generation devices contain a four-level hardware stack for saving the contents of the program counter during interrupts and subroutine calls. Instructions are available for saving the device's complete context. PUSH and POP instructions permit a level of nesting restricted only by the amount of available RAM. The interrupts used in these devices are maskable.

input/output

The 16-bit parallel data bus can be utilized to perform I/O functions in two cycles. The I/O ports are addressed by the three LSBs on the address lines. In addition, a polling input for bit test and jump operations ($\overline{\text{BIO}}$) and an interrupt pin ($\overline{\text{INT}}$) have been incorporated for multitasking.

instruction set

A comprehensive instruction set supports both numeric-intensive operations, such as signal processing, and general-purpose operations, such as high-speed control. All of the first-generation devices are object-code compatible and use the same 60 instructions. The instruction set consists primarily of single-cycle single-word instructions, permitting execution rates of more than six million instructions per second. Only infrequently used branch and I/O instructions are multicycle. Instructions that shift data as part of an arithmetic operation execute in a single cycle and are useful for scaling data in parallel with other operations.

Three main addressing modes are available with the instruction set: direct, indirect, and immediate addressing.

direct addressing

In direct addressing, seven bits of the instruction word concatenated with the 1-bit data page pointer form the data memory address. This implements a paging scheme in which the first page contains 128 words, and the second page contains up to 128 words.

indirect addressing

Indirect addressing forms the data memory address from the least-significant eight bits of one of the two auxiliary registers, AR0 and AR1. The Auxiliary Register Pointer (ARP) selects the current auxiliary register. The auxiliary registers can be automatically incremented or decremented and the ARP changed in parallel with the execution of any indirect instruction to permit single-cycle manipulation of data tables. Indirect addressing can be used with all instructions requiring data operands, except for the immediate operand instructions.

immediate addressing

Immediate instructions derive data from part of the instruction woıd rather than from the data RAM. Some useful immediate instructions are multiply immediate (MPYK), load accumulator immediate (LACK), and load auxiliary register immediate (LARK).

instruction set summary

Table 1 lists the symbols and abbreviations used in Table 2, the instruction set summary. Table 2 contains a short description and the opcode for each SMJ320 first-generation instruction. The summary is arranged according to function and alphabetized within each functional group.

TABLE 1. INSTRUCTION SYMBOLS

SYMBOL	MEANING
ACC	Accumulator
D	Data memory address field
I	Addressing mode bit
K	Immediate operand field
PA	3-bit port address field
R	1-bit operand field specifying auxiliary register
S	4-bit left-shift code
X	3-bit accumulator left-shift field

TABLE 2. SMJ320 FIRST-GENERATION INSTRUCTION SET SUMMARY

ACCUMULATOR INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABS	Absolute value of accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0
ADD	Add to accumulator with shift	1	1	0	0	0	0	←S→				I	←D→						
ADDH	Add to high-order accumulator bits	1	1	0	1	1	0	0	0	0	0	I	←D→						
ADDS	Add to accumulator with no sign extension	1	1	0	1	1	0	0	0	0	1	I	←D→						
AND	AND with accumulator	1	1	0	1	1	1	1	0	0	1	I	←D→						
LAC	Load accumulator with shift	1	1	0	0	1	0	←S→				I	←D→						
LACK	Load accumulator immediate	1	1	0	1	1	1	1	1	1	0	←K→							
OR	OR with accumulator	1	1	0	1	1	1	1	0	1	0	I	←D→						
SACH	Store high-order accumulator bits with shift	1	1	0	1	0	1	1	←X→			I	←D→						
SACL	Store low-order accumulator bits	1	1	0	1	0	1	0	0	0	0	I	←D→						
SUB	Subtract from accumulator with shift	1	1	0	0	0	1	←S→				I	←D→						
SUBC	Conditional subtract (for divide)	1	1	0	1	1	0	0	1	0	0	I	←D→						
SUBH	Subtract from high-order accumulator bits	1	1	0	1	1	0	0	0	1	0	I	←D→						
SUBS	Subtract from accumulator with no sign extension	1	1	0	1	1	0	0	0	1	1	I	←D→						
XOR	Exclusive OR with accumulator	1	1	0	1	1	1	1	0	0	0	I	←D→						
ZAC	Zero accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1
ZALH	Zero accumulator and load high-order bits	1	1	0	1	1	0	0	1	0	1	I	←D→						
ZALS	Zero accumulator and load low-order bits with no sign extension	1	1	0	1	1	0	0	1	1	0	I	←D→						

AUXILIARY REGISTER AND DATA PAGE POINTER INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LAR	Load auxiliary register	1	1	0	0	1	1	1	0	0	R	I	←D→						
LARK	Load auxiliary register immediate	1	1	0	1	1	1	0	0	0	R	←K→							
LARP	Load auxiliary register pointer immediate	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	K
LDP	Load data memory page pointer	1	1	0	1	1	0	1	1	1	1	I	←D→						
LDPK	Load data memory page pointer immediate	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	K
MAR	Modify auxiliary register and pointer	1	1	0	1	1	0	1	0	0	0	I	←D→						
SAR	Store auxiliary register	1	1	0	0	1	1	0	0	0	R	I	←D→						

TABLE 2. SMJ320 FIRST-GENERATION INSTRUCTION SET SUMMARY (CONTINUED)

BRANCH INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B	Branch unconditionally	2	2	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BANZ	Branch on auxiliary register not zero	2	2	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BGEZ	Branch if accumulator ≥ 0	2	2	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BGZ	Branch if accumulator > 0	2	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BIOZ	Branch on $\overline{\text{BIO}}$ = 0	2	2	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BLEZ	Branch if accumulator ≤ 0	2	2	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BLZ	Branch if accumulator < 0	2	2	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BNZ	Branch if accumulator ≠ 0	2	2	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BV	Branch on overflow	2	2	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
BZ	Branch if accumulator = 0	2	2	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
CALA	Call subroutine from accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0
CALL	Call subroutine immediately	2	2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	◄— BRANCH ADDRESS —►											
RET	Return from subroutine or interrupt routine	2	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	1

T REGISTER, P REGISTER, AND MULTIPLY INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APAC	Add P register to accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1
LT	Load T register	1	1	0	1	1	0	1	0	1	0	I	◄— D —►						
LTA	LTA combines LT and APAC into one instruction	1	1	0	1	1	0	1	1	0	0	I	◄— D —►						
LTD	LTD combines LT, APAC, and DMOV into one instruction	1	1	0	1	1	0	1	0	1	1	I	◄— D —►						
MPY	Multiply with T register, store product in P register	1	1	0	1	1	0	1	1	0	1	I	◄— D —►						
MPYK	Multiply T register with immediate operand; store product in P register	1	1	1	0	0	◄— K —►												
PAC	Load accumulator from P register	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	0
SPAC	Subtract P register from accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0

TABLE 2. SMJ320 FIRST-GENERATION INSTRUCTION SET SUMMARY (CONCLUDED)

CONTROL INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DINT	Disable interrupt	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1
EINT	Enable interrupt	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0
LST	Load status register	1	1	0	1	1	1	1	0	1	1	I	◄—D—►						
NOP	No operation	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
POP	POP stack to accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1
PUSH	PUSH stack from accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0
ROVM	Reset overflow mode	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	0
SOVM	Set overflow mode	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1
SST	Store status register	1	1	0	1	1	1	1	1	0	0	I	◄—D—►						

I/O AND DATA MEMORY OPERATIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMOV	Copy contents of data memory location into next higher location	1	1	0	1	1	0	1	0	0	1	I	◄—D—►						
IN	Input data from port	2	1	0	1	0	0	0	◄PA►			I	◄—D—►						
OUT	Output data to port	2	1	0	1	0	0	1	◄PA►			I	◄—D—►						
TBLR	Table read from program memory to data RAM	3	1	0	1	1	0	0	1	1	1	I	◄—D—►						
TBLW	Table write from data RAM to program memory	3	1	0	1	1	1	1	1	0	1	I	◄—D—►						

development support products

Together, Texas Instruments and its authorized third-party suppliers offer an extensive line of development support products to assist the user in all aspects of TMS320 first-generation-based design and development. These products range from development and application software to complete hardware development and evaluation systems. Table 4 lists the development support products for the first-generation TMS320 devices.

System development may begin with the use of the simulator, evaluation module (EVM), or emulator (XDS), along with an assembler/linker. These tools give the TMS320 user various means of evaluation, from software simulation of the first-generation TMS320s (simulator) to full-speed in-circuit emulation with hardware and software break point tracing and timing capabilities (XDS).

Software and hardware can be developed simultaneously by using the macro assembler/linker or simulator for software development, the XDS for hardware development, and the evaluation module for both software development and limited hardware development.

Many third-party vendors offer additional development support for the first-generation TMS320s, including assembler/linkers, simulators, high-level languages, applications software, algorithm development tools, application boards, software development boards, and in-circuit emulators. Refer to the *TMS320 Family Development Support Reference Guide* (SPRU011A) for further information about TMS320 development support products offered by both Texas Instruments and its third-party suppliers.

Additional support for the TMS320 products consists of an extensive library of product and applications documentation. Three-day DSP design workshops are offered by the TI Regional Technology Centers (RTCs). These workshops provide insight into the architecture and the instruction set of the first-generation TMS320s as well as hands-on training with the TMS320 development tools. When technical questions arise in regard to a TMS320 member, contact Texas Instruments TMS320 Hotline at (713) 274-2320. Or, keep informed on the latest TI and third-party development support tools by accessing the libraries of application source code via the DSP Bulletin Board Service (BBS) at (713) 274-2323. The BBS provides access for the 2400-/1200-/300-bps modems.

documentation support

Extensive documentation supports the first-generation TMS320 devices from product announcement through applications development. The types of documentation include data sheets with design specifications, complete user's guides, and 750 pages of application reports published in the book *Digital Signal Processing Applications with the TMS320 Family.*

A series of DSP textbooks is being published to support digital signal processing research and education. The first book, *DFT/FFT and Convolution Algorithms,* is now available. The TMS320 newsletter, *Details on Signal Processing,* is published quarterly and distributed to update TMS320 customers on product information. The TMS320 DSP bulletin board service provides access to large amounts of information pertaining to the TMS320 family.

Refer to the *TMS320 Family Development Support Reference Guide* for further information about TMS320 documentation. To receive copies of first-generation SMJ320 literature, call the Customer Response Center at 1-800-232-3200.

TABLE 3. TMS320 FIRST-GENERATION SOFTWARE AND HARDWARE SUPPORT

SOFTWARE TOOLS	PART NUMBER
Macro Assembler/Linker	
PC/MS-DOS	TMDS3242850-02
VAX/VMS	TMDS3242250-08
VAX ULTRIX	TMDS3242260-08
SUN-3 UNIX	TMDS3242550-08
Simulator	
PC/MS-DOS	TMDS3240811-02
VAX/VMS	TMDS3240211-08
Digital Filter Design Package (DFDP)	
IBM PC PC-DOS	DFDP/IBM002
DSP Software Library	
PC/MS-DOS	TMDC3240812-12
VAX/VMS	TMDC3240212-18
TMS320 Bell 212A Modem Software	
PC/MS-DOS	TMDX3240813-12
Data Encryption Standard Software	
PC/MS-DOS	TMDX3240814-12
HARDWARE TOOLS	**PART NUMBER**
Evaluation Tools	
Evaluation Module (EVM)	RTC/EVM320A-03
Analog Interface Board 1 (AIB1)	RTC/EVM320C-06
Analog Interface Board 2 (AIB2)	RTC/AIB320A-06
EPROM DSP Starter Kit (TMS320E15)	RTC/EVM320E-15
XDS/22 Emulators	
TMS320C10/C15	TMDS3262211
TMS320C14	TMDX3262214
TMS320C17	TMDX3262217
XDS/22 Upgrade Kits	
TMS32010 → TMS320C10/C15	TMDS3282215
TMS320C10/C15 → TMS320C14	TMDX3285010 and TMDX3285018
TMS320C10/C15 → TMS320C17	TMDX3285014 and TMDX3285018
EPROM Programming Adaptor Sockets	
40- to 28-pin (TMS320E15/E17)	RTC/PGM320A-06
44- to 28-pin (TMS320E15/E17)	RTC/PGM320C-06
68- to 28-pin (TMS320E14)	TMDX3270110
Additional Target Connector	
44-pin PLCC (TMS320C10)	TMDX3288810

absolute maximum ratings over specified temperature range (unless otherwise noted)†

Supply voltage range, V_{CC}‡	−0.3 V to 7 V
Input voltage range	−0.3 V to 7 V
Output voltage range	−0.3 V to 7 V
Continuous power dissipation: SMJ320C15	275 mW
SMJ320C15-25	330 mW
Maximum operating case temperature	125°C
Minimum operating free-air temperature	−55°C
Storage temperature range	−65°C to +150°C

† Stresses beyond 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 beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

‡ All voltage values are with respect to V_{SS}.

recommended operating conditions

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage		4.5	5	5.5	V
V_{SS}	Supply voltage			0		V
V_{IH}	High-level input voltage	All inputs except CLKIN	2			V
		CLKIN	3			
V_{IL}	Low-level input voltage	All inputs except MC/$\overline{MP}$			0.8	V
		MC/$\overline{MP}$			0.6	
I_{OH}	High-level output current (all outputs)				−300	μA
I_{OL}	Low-level output current (all outputs)				2	mA
T_A	Operating free-air temperature		−55			°C
T_C	Operating case temperature				125	°C

electrical characteristics over specified temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS		MIN	TYP§	MAX	UNIT
V_{OH}	High-level output voltage		I_{OH} = MAX		2.4	3		V
			I_{OH} = 20 μA (see Note 1)		V_{CC}−0.4 V#			
V_{OL}	Low-level output voltage		I_{OL} = MAX			0.3	0.5	V
I_{OZ}	Off-state output current		V_{CC} = MAX	V_O = 2.4 V			20	μA
				V_O = 0.4 V			−20	
I_I	Input current		V_I = V_{SS}, V_{CC} = MAX				±50	μA
I_{CC}¶	Supply current	SMJ320C15	V_{CC} = 5.5 V, f = 20.5 MHz				55	mA
		SMJ320C15-25	V_{CC} = 5.5 V, f = 25.6 MHz				65	
C_i	Input capacitance	Data bus	f = 1 MHz, All other pins 0 V			25#		pF
		All others				15#		
C_o	Output capacitance	Data bus				25#		
		All others				10#		

§ All typical values except for I_{CC} are at V_{CC} = 5 V, T_A = 25°C.

¶ I_{CC} characteristics are inversely proportional to temperature; i.e., I_{CC} decreases approximately linearly with temperature.

Value derived from characterization data and is guaranteed to limit but not tested.

NOTE 1: This voltage specification is included for interface to HC logic. However, note that all other timing parameters defined in this data sheet are specified for TTL logic levels and will differ for HC logic levels.

CLOCK CHARACTERISTICS AND TIMING

The SMJ320C15 can use either its internal oscillator or an external frequency source for a clock.

internal clock option

The internal oscillator is enabled by connecting a crystal across X1 and X2/CLKIN (see Figure 1). The frequency of CLKOUT is one-fourth the crystal fundamental frequency. The crystal should be fundamental mode, and parallel resonant, with an effective series resistance of 30 ohms, a power dissipation of 1 mW, and be specified at a load capacitance of 20 pF.

PARAMETER		TEST CONDITIONS	MIN	NOM	MAX	UNIT
Crystal frequency, f_x	SMJ320C15	−55°C to 125°C with 8 MHz crystal	6.7†		20.5†	MHz
	SMJ320C15-25		6.7†		25.6†	
C1, C2				10		pF

†Value derived from characterization data and is guaranteed to limit but not tested.

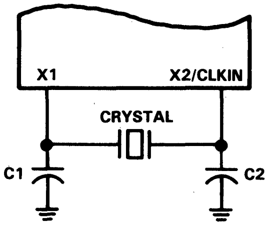


FIGURE 1. INTERNAL CLOCK OPTION

external clock option

An external frequency source can be used by injecting the frequency directly into X2/CLKIN with X1 left unconnected. The external frequency injected must conform to the specifications listed in the timinq requirements table.

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	SMJ320C15			SMJ320C15-25			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	
$t_{c(C)}$[‡]	CLKOUT cycle time	R_L = 825 Ω, C_L = 100 pF	195.12	200	600	156.25	160	600	ns
$t_{r(C)}$	CLKOUT rise time			10			10		ns
$t_{f(C)}$	CLKOUT fall time			8			8		ns
$t_{w(CL)}$	Pulse duration, CLKOUT low			92			72		ns
$t_{w(CH)}$	Pulse duration, CLKOUT high			90			70		ns
$t_{d(MCC)}$[†]	Delay time CLKIN↑ to CLKOUT↓		15		40	15		40	ns

[†] Value derived from characterization data and is guaranteed to limit but not tested.

[‡] $t_{c(C)}$ is the cycle time of CLKOUT i.e., $4t_{c(MC)}$ (4 times CLKIN cycle time if an external oscillator is used).

timing requirements over recommended operating conditions

		SMJ320C15			SMJ320C15-25			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
$t_{c(MC)}$	Master clock cycle time	48.78	50	150	39.06	40	150	ns
$t_{r(MC)}$	Rise time master clock input		5			5		ns
$t_{f(MC)}$	Fall time master clock input		5			5		ns
$t_{w(MCP)}$[†]	Pulse duration, master clock	$0.4t_{c(MC)}$		$0.6t_{c(MC)}$	$0.45t_{c(MC)}$		$0.55t_{c(MC)}$	ns
$t_{w(MCL)}$	Pulse duration, master clock low		20			15		ns
$t_{w(MCH)}$	Pulse duration, master clock high		20			15		ns

[†] Value derived from characterization data and is guaranteed to limit but not tested.

NOTE 2: CLKIN rise and fall times must be less than 10 ns.

MEMORY AND PERIPHERAL INTERFACE TIMING

switching characteristics over recommended operating conditions

PARAMETER		SMJ320C15			SMJ320C15-25			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
t_{d1}	Delay time, CLKOUT↓ to address bus valid (see Note 3)	10[§]		50	10[§]		40	ns
t_{d2}	Delay time, CLKOUT↓ to $\overline{\text{MEN}}$↓	$0.25t_{c(C)}-5$[§]		$0.25t_{c(C)}+15$	$0.25t_{c(C)}-5$[§]		$0.25t_{c(C)}+12$	ns
t_{d3}	Delay time, CLKOUT↓ to $\overline{\text{MEN}}$↑	−10[§]		15	−10[§]		12	ns
t_{d4}	Delay time, CLKOUT↓ to $\overline{\text{DEN}}$↓	$0.25t_{c(C)}-5$[§]		$0.25t_{c(C)}+15$	$0.25t_{c(C)}-5$[§]		$0.25t_{c(C)}+12$	ns
t_{d5}	Delay time, CLKOUT↓ to $\overline{\text{DEN}}$↑	−10[§]		15	−10[§]		12	ns
t_{d6}	Delay time, CLKOUT↓ to $\overline{\text{WE}}$↓	$0.50t_{c(C)}-5$[§]		$0.50t_{c(C)}+15$	$0.50t_{c(C)}-5$[§]		$0.50t_{c(C)}+12$	ns
t_{d7}	Delay time, CLKOUT↓ to $\overline{\text{WE}}$↑	−10[§]		15	−10[§]		12	ns
t_{d8}	Delay time, CLKOUT↓ to data bus OUT valid			$0.25t_{c(C)}+65$			$0.25t_{c(C)}+52$	ns
t_{d9}	Time after CLKOUT↓ that data bus starts to be driven	$0.25t_{c(C)}-5$[§]			$0.25t_{c(C)}-5$[§]			ns
t_{d10}	Time after CLKOUT↓ that data bus stops being driven			$0.25t_{c(C)}+40$[§]			$0.25t_{c(C)}+40$[§]	ns
t_v	Data bus OUT valid after CLKOUT↓	$0.25t_{c(C)}-10$			$0.25t_{c(C)}-10$			ns
$t_{h(A-WMD)}$	Address hold time after $\overline{\text{WE}}$↑, $\overline{\text{MEN}}$↑, or $\overline{\text{DEN}}$↑ (see Note 3)	0[§]			0[§]			ns
$t_{su(A-MD)}$	Address bus setup time prior to $\overline{\text{MEN}}$↓ or $\overline{\text{DEN}}$↓	$0.25t_{c(C)}-45$			$0.25t_{c(C)}-35$			ns

[§] Value derived from characterization data and is guaranteed to limit but not tested.

NOTE 3: Address bus will be valid upon $\overline{\text{WE}}$↑, $\overline{\text{DEN}}$↑ or $\overline{\text{MEN}}$↑.

timing requirements over recommended operating conditions

		TEST CONDITIONS	MIN	NOM	MAX	UNIT
$t_{su(D)}$	Setup time data bus valid prior to CLKOUT↓	R_L = 825 Ω, C_L = 100 pF, See Figure 2	50			ns
$t_{h(D)}$	Hold time data bus held valid after CLKOUT↓ (see Note 4)		0			ns

NOTE 4: Data may be removed from the data bus upon $\overline{\text{MEN}}$↑ or $\overline{\text{DEN}}$↑ preceding CLKOUT↓.

RESET ($\overline{\text{RS}}$) TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d11}	Delay time $\overline{\text{DEN}}$↑, $\overline{\text{WE}}$↑, and $\overline{\text{MEN}}$↑ from $\overline{\text{RS}}$	R_L = 825 Ω, C_L = 100 pF, See Figure 5			$\frac{1}{2}t_{c(C)}+50$†	ns
$t_{dis(R)}$	Data bus disable time after $\overline{\text{RS}}$				$\frac{1}{4}t_{c(C)}+50$†	ns

†Value derived from characterization data and is guaranteed to limit but not tested.

timing requirements over recommended operating conditions

		SMJ320C15			SMJ320C15-25			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
$t_{su(R)}$	Reset ($\overline{\text{RS}}$) setup time prior to CLKOUT (see Note 5)	50			40			ns
$t_{w(R)}$	$\overline{\text{RS}}$ pulse duration	$5t_{c(C)}$			$5t_{c(C)}$			ns

NOTE 5: $\overline{\text{RS}}$ can occur anytime during a clock cycle. Time given is minimum to ensure synchronous operation.

INTERRUPT ($\overline{\text{INT}}$) TIMING

timing requirements over recommended operating conditions

		SMJ320C15			SMJ320C15-25			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
$t_{f(INT)}$	Fall time $\overline{\text{INT}}$ (see Note 6)		10			10		ns
$t_{w(INT)}$	Pulse duration $\overline{\text{INT}}$	$t_{c(C)}$			$t_{c(C)}$			ns
$t_{su(INT)}$	Setup time $\overline{\text{INT}}$↓ before CLKOUT↓	50			40			ns

NOTE 6: $\overline{\text{INT}}$ fall time must be less than 15 ns.

$\overline{\text{BIO}}$ TIMING

timing requirements over recommended operating conditions

		SMJ320C15			SMJ320C15-25			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
$t_{f(IO)}$	Fall time $\overline{\text{BIO}}$ (see Note 7)		10			10		ns
$t_{w(IO)}$	Pulse duration $\overline{\text{BIO}}$	$t_{c(C)}$			$t_{c(C)}$			ns
$t_{su(IO)}$	Setup time $\overline{\text{BIO}}$↓ before CLKOUT↓	50			40			ns

NOTE 7: $\overline{\text{BIO}}$ fall time must be less than 15 ns.

PARAMETER MEASUREMENT INFORMATION

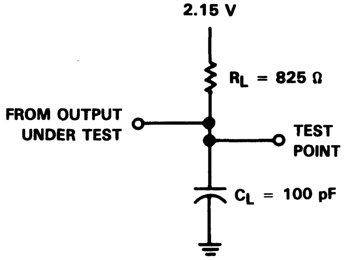
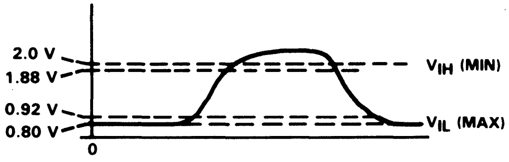
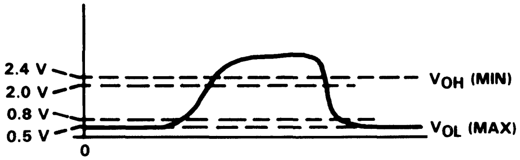


FIGURE 2. TEST LOAD CIRCUIT



(a) INPUT



(b) OUTPUTS

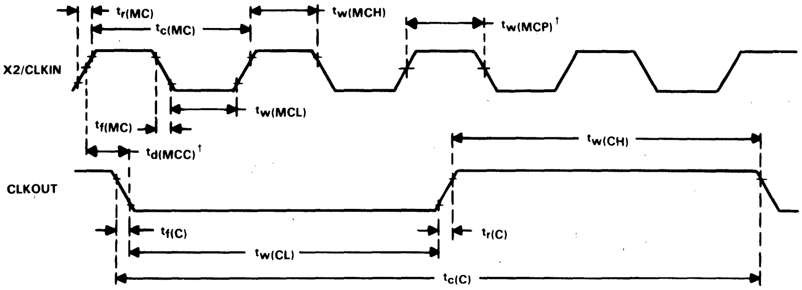
FIGURE 3. VOLTAGE REFERENCE LEVELS

TIMING DIAGRAMS

This section contains all the timing diagrams for the SMJ320 first-generation devices. Refer to the top corner for the specific device.

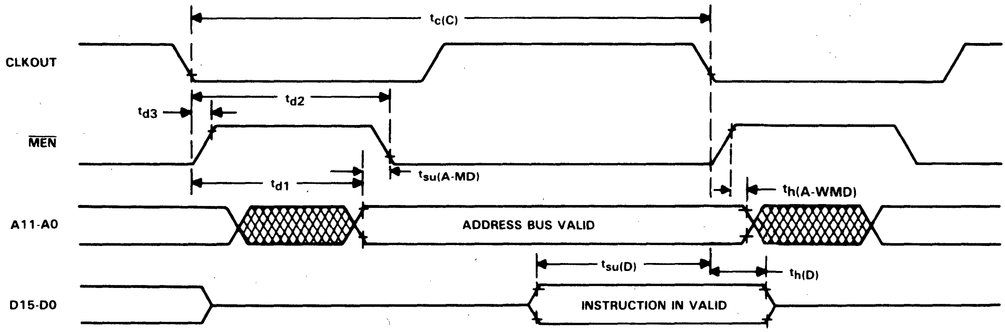
Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

clock timing

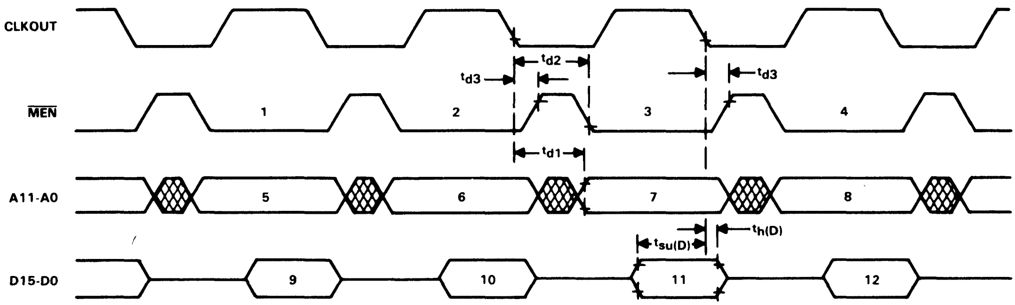


†$t_{d(MCC)}$ and $t_{w(MCP)}$ are referenced to an intermediate level of 1.5 volts on the CLKIN waveform.

memory read timing



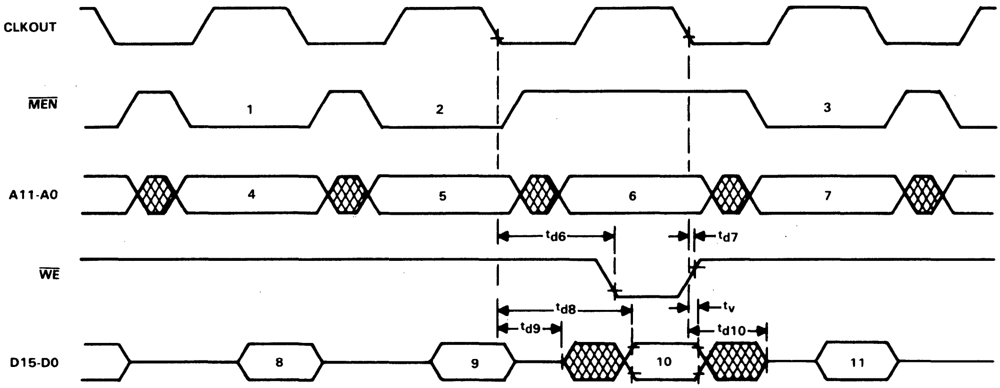
TBLR instruction timing



LEGEND:

1. TBLR INSTRUCTION PREFETCH
2. DUMMY PREFETCH
3. DATA FETCH
4. NEXT INSTRUCTION PREFETCH
5. ADDRESS BUS VALID
6. ADDRESS BUS VALID
7. ADDRESS BUS VALID
8. ADDRESS BUS VALID
9. INSTRUCTION VALID
10. INSTRUCTION VALID
11. DATA INPUT VALID
12. INSTRUCTION VALID

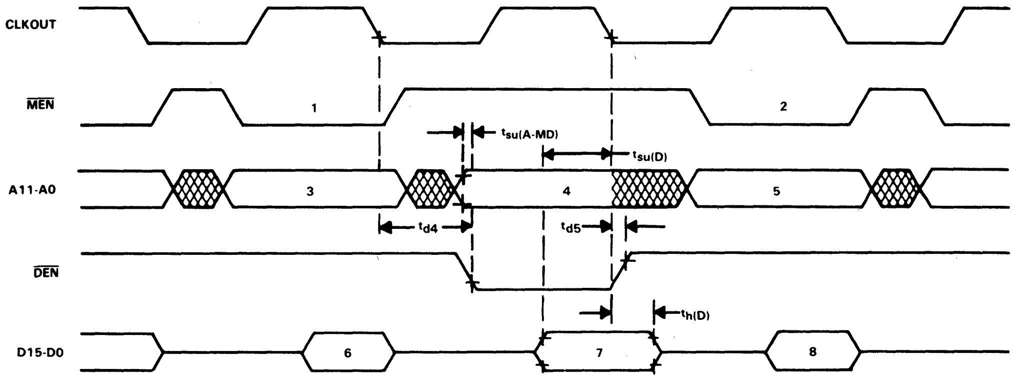
TBLW instruction timing



LEGEND:

1. TBLW INSTRUCTION PREFETCH
2. DUMMY PREFETCH
3. NEXT INSTRUCTION PREFETCH
4. ADDRESS BUS VALID
5. ADDRESS BUS VALID
6. ADDRESS BUS VALID
7. ADDRESS BUS VALID
8. INSTRUCTION VALID
9. INSTRUCTION VALID
10. DATA OUTPUT VALID
11. INSTRUCTION VALID

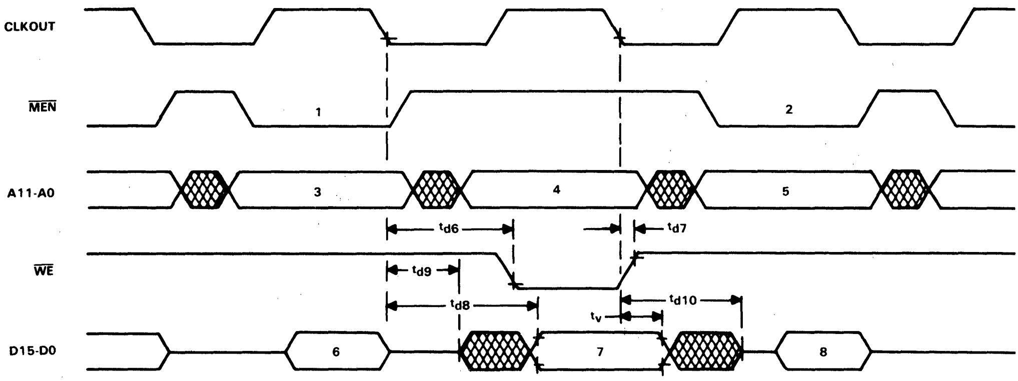
IN instruction timing



LEGEND:

1. IN INSTRUCTION PREFETCH
2. NEXT INSTRUCTION PREFETCH
3. ADDRESS BUS VALID
4. PERIPHERAL ADDRESS VALID
5. ADDRESS BUS VALID
6. INSTRUCTION VALID
7. DATA INPUT VALID
8. INSTRUCTION VALID

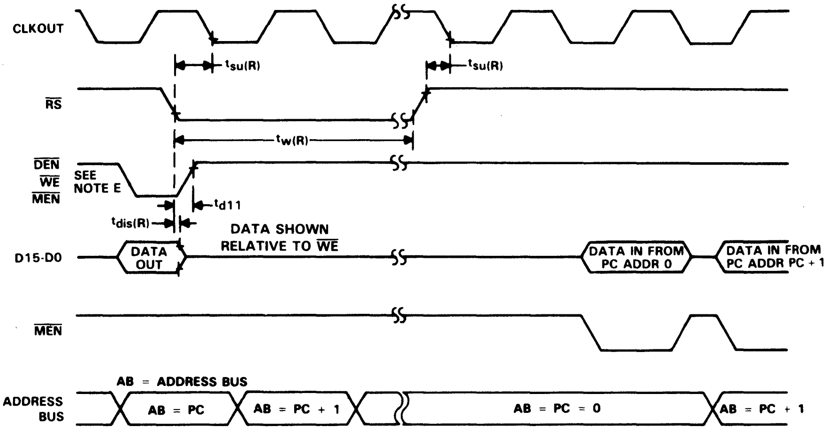
OUT instruction timing



LEGEND:

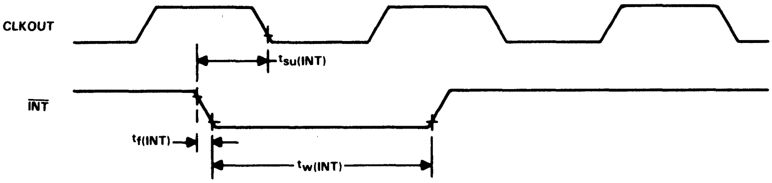
1. OUT INSTRUCTION PREFETCH
2. NEXT INSTRUCTION PREFETCH
3. ADDRESS BUS VALID
4. PERIPHERAL ADDRESS VALID
5. ADDRESS BUS VALID
6. INSTRUCTION VALID
7. DATA OUTPUT VALID
8. INSTRUCTION VALID

reset timing

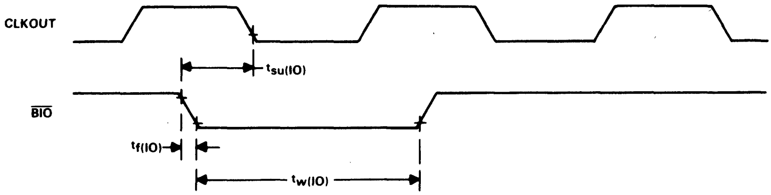


NOTES:
A. RS forces DEN, WE, and MEN high and places data bus D0 through D15 in a high-impedance state. AB outputs (and program counter) are synchronously cleared to zero after the next complete CLK cycle from RS↓.
B. RS must be maintained for a minimum of five clock cycles.
C. Resumption of normal program will commence after one complete CLK cycle from RS↑.
D. Due to the synchronizing action on RS, time to execute the function can vary dependent upon when RS↑ or RS↓ occur in the CLK cycle.
E. Diagram shown is for definition purpose only. DEN, WE, and MEN are mutually exclusive.
F. During a write cycle, RS may produce an invalid write address.

interrupt timing

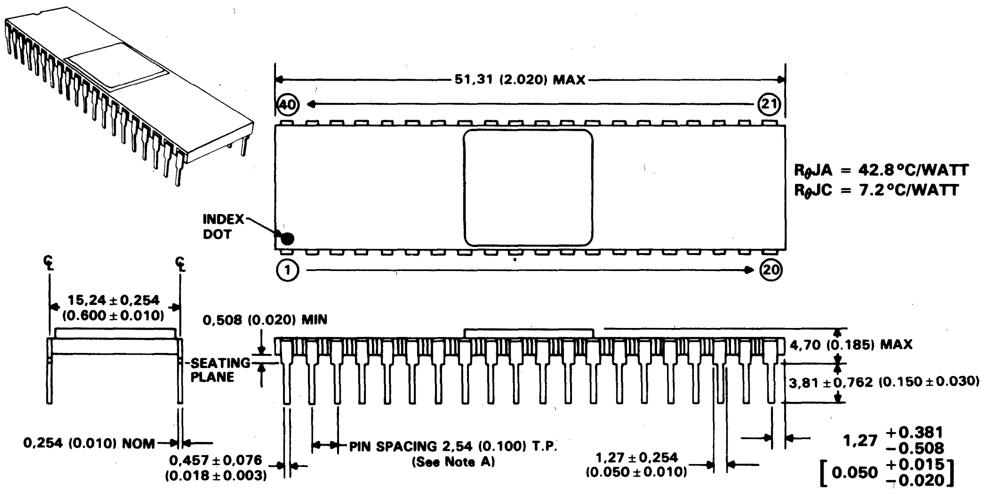


BIO timing



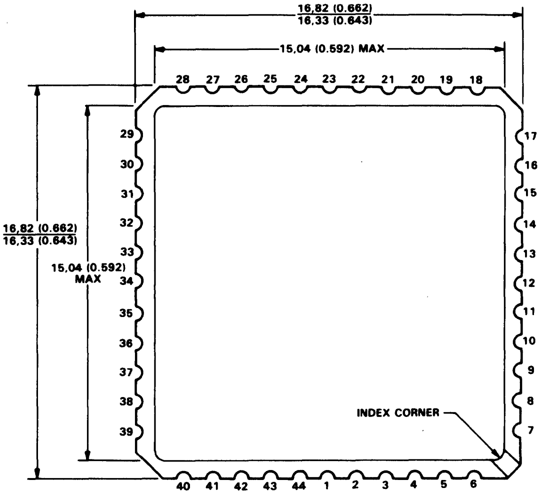
MECHANICAL DATA

40-pin JD ceramic dual-in-line package

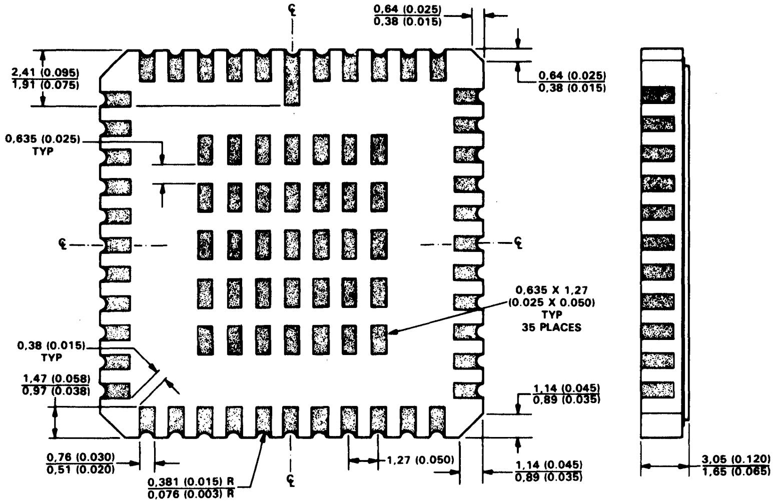


NOTE A: Each pin centerline is located within 0,254 (0.010) of its true longitudinal position.
ALL DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES

44-pad FD ceramic chip carrier package



$R_{\theta JA}$ = 44.7°C/WATT
$R_{\theta JC}$ = 13.3°C/WATT



The checkerboard pattern is aligned vertically and is symmetrical horizontally as shown.
ALL DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES
Texas Instruments reserves the right to make changes at any time in order to improve design and to supply the best product possible.

Appendix C

ROM Codes

Size of a printed circuit board must be considered in many DSP applications. To fully utilize the board space, Texas Instruments offers two options which will reduce the chip count and provide a single-chip solution to its customers. These options incorporate 4K words of on-chip program from either a mask programmable ROM or an EPROM. This allows the customer to use a code-customized processor for a specific application while taking advantage of the following:

- Greater memory expansion
- Lower system cost
- Less hardware and wiring
- Smaller PCB

If used otten, the routine or entire algorithm can be programmed into the on-chip ROM of a TMS320 DSP. TMS320 programs can also be expanded by using external memory; this reduces chip count and allows for a more flexible program memory. Multiple functions are easily implemented by a single device, thus enhancing system capabilities.

TMS320 Development Tools are used to develop, test, refine, and finalize the algorithms. The microcomputer/microprocessor (MC/$\overline{\text{MP}}$) mode is available on all ROM-coded TMS320 DSP devices when accessing either on-chip or off-chip memory is required. The microprocessor mode is used to develop, test, and refine a system application. In this mode of operation, the TMS320 acts as a standard microprocessor by using external program memory. When the algorithm has been finalized, the designer may submit the code to Texas Instruments for masking into the on-chip program ROM. At that time, the TMS320 becomes a microcomputer which executes customized programs out of the on-chip ROM. Should the code need changing or upgrading, the TMS320 may once again be use in the microprocessor mode. This shortens the field upgrade time and avoids the possibility of inventory obsolescence.

Figure C-1 illustrates the procedural flow for TMS320 masked parts. When ordering, there is a one-time/non-refundable charge for mask-tooling. A minimum production order per year is required for any masked-ROM device. ROM codes will be deleted from the TI system after one year from the last delivery.

A digital signal processor with the EPROM option is the solution for low-volume production orders. The EPROM option allows for form-factor emulation. Field upgrades and changes are possible with the EPROM option.

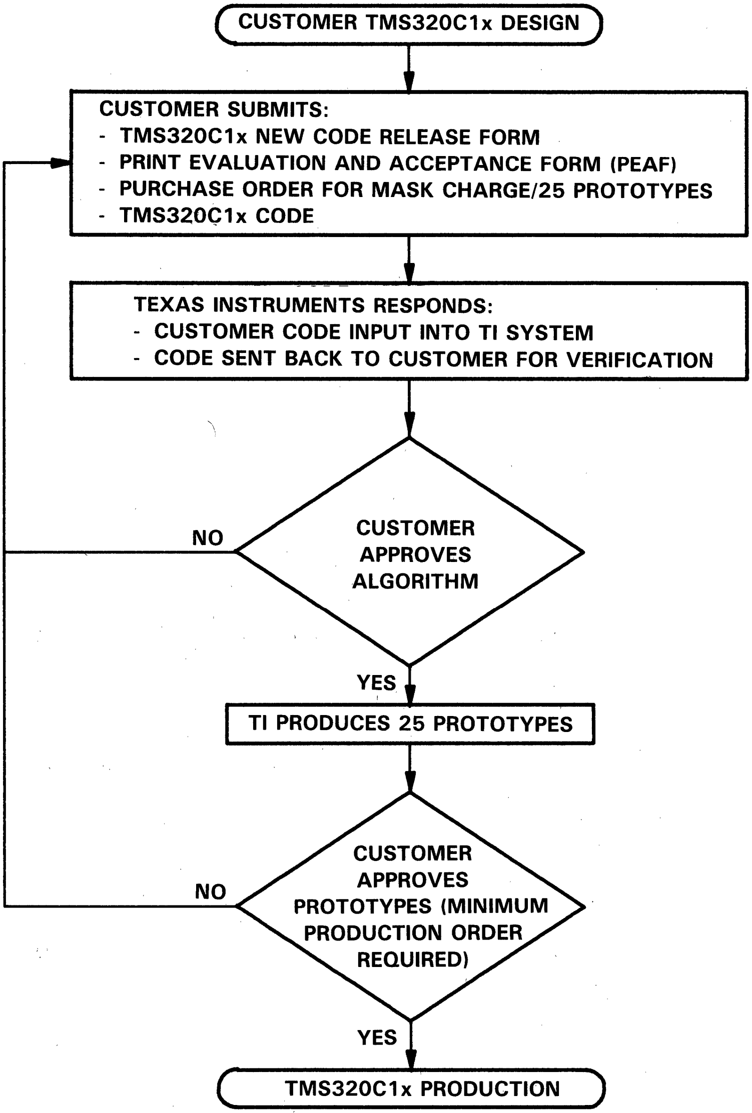

CUSTOMER TMS320C1x DESIGN
CUSTOMER SUBMITS:
- TMS320C1x NEW CODE RELEASE FORM
- PRINT EVALUATION AND ACCEPTANCE FORM (PEAF)
- PURCHASE ORDER FOR MASK CHARGE/25 PROTOTYPES
- TMS320C1x CODE
TEXAS INSTRUMENTS RESPONDS:
- CUSTOMER CODE INPUT INTO TI SYSTEM
- CODE SENT BACK TO CUSTOMER FOR VERIFICATION
NO
CUSTOMER APPROVES ALGORITHM
YES
TI PRODUCES 25 PROTOTYPES
NO
CUSTOMER APPROVES PROTOTYPES (MINIMUM PRODUCTION ORDER REQUIRED)
YES
TMS320C1x PRODUCTION

Figure C-1. TMS320 ROM Code Flowchart

A TMS320 ROM code may be submitted in one of the following formats (the preferred media is 5 1/4" floppies):

FLOPPY:	TI Cross-Assembler Format
EPROM (others):	TMS2764, TMS2508, TMS2516, TMS2532, TMS2564
PROM:	TBP28S166, TBP28S86
MODEM (BBS):	TI Cross-Assembler Format

When a code is submitted to Texas Instruments for masking, the code is reformatted to accommodate the TI mask generation system. System level verification by the customer is therefore necessary. Although the code has been reformatted, it is important that the changes remain transparent to the user and do not affect the execution of the algorithm. The formatting changes involve the removal of address re-location information (the code address begins at the base address of the ROM in the TMS320 device and progresses without gaps to the last address of the ROM on the TMS320 device) and the addition of data in the reserved locations of the ROM for device ROM test. Note that because these changes have been made, a 'checksum' comparison is not a valid means of verification.

With each masked device order, the customer must sign a disclaimer stating:

> "The units to be shipped against this order were assembled, for expediency purposes, on a prototype (i.e., non-production qualified) manufacturing line, the reliability of which is not fully characterized. Therefore, the anticipated inherent reliability of these prototype units cannot be expressly defined."

and a release stating:

> "Any masked ROM device may be resymbolized as TI standard product and resold as though it were an unprogrammed version of the device at the convenience of Texas Instruments."

Contact the nearest TI Field Sales Office for more information on procedures, leadtimes, and cost.

Appendix D

Quality and Reliability

The quality and reliability performance of Texas Instruments Microprocessor and Microcontroller Products, which includes the three generations of TMS320 digital signal processors, relies on feedback from:

- Our customers
- Our total manufacturing operation from front-end wafer fabrication to final shipping inspection
- Product quality and reliability monitoring.

Our customer's perception of quality must be the governing criterion for judging performance. This concept is the basis for Texas Instruments Corporate Quality Policy, which is as follows:

> "For every product or service we offer, we shall define the requirements that solve the customer's problems, and we shall conform to those requirements without exception."

Texas Instruments offers a leadership reliability qualification system, based on years of experience with leading-edge memory technology as well as years of research into customer requirements. Quality and reliability programs at TI are therefore based on customer input and internal information to achieve constant improvement in quality and reliability.

TI Qualification test updates are available upon request at no charge. TI will consider performing any additional reliability test(s), if requested. For more information on TI quality and reliability programs, contact the nearest TI Field Sales Office.

Note:

Texas Instruments reserves the right to make changes in MOS Semiconductor test limits, procedures, or processing without notice. Unless prior arrangements for notification have been made, TI advises all customers to reverify current test and manufacturing conditions prior to relying on published data.

D.1 Reliability Stress Tests

Accelerated stress tests are performed on new semiconductor products and process changes to ensure product reliability excellence. The typical test environments used to qualify new products or major changes in processing are:

- High-temperature operating life
- Storage life
- Temperature cycling
- Biased humidity
- Autoclave
- Electrostatic discharge
- Package integrity
- Electromigration
- Channel-hot electrons (performed on geometries less than 2.0 μm).

Typical events or changes that require internal requalification of product include:

- New die design, shrink, or layout
- Wafer process (baseline/control systems, flow, mask, chemicals, gases, dopants, passivation, or metal systems)
- Packaging assembly (baseline control systems or critical assembly equipment)
- Piece parts (such as lead frame, mold compound, mount material, bond wire, or lead finish)
- Manufacturing site.

TI reliability control systems extend beyond qualification. Total reliability controls and management include a product reliability monitor and final product release controls. MOS memories, utilizing high-density active elements, serve as leading indicators in wafer-process integrity at TI MOS fabrication sites, enhancing all MOS logic device yields and reliability. Thousands of MOS devices per month are randomly tested to ensure product reliability and excellence.

Table D-1 lists the microprocessor and microcontroller reliability tests, the duration of the test, and sample size. The following defines and describes those tests in the table.

AOQ (Average Outgoing Quality) Amount of defective product in a population, usually expressed in terms of parts per million (PPM).

FIT (Failure In Time) Estimated field failure rate in number of failures per billion power-on device hours; 1000 FITS equals 0.1 percent fail per 1000 device hours.

Operating lifetest

Device dynamically exercised at a high ambient temperature (usually 125°C) to simulate field usage that would expose the device to a much lower ambient temperature (such as 55°C). Using a derived high temperature, a 55°C ambient failure rate can be calculated.

High-temperature storage

Device exposed to 150°C unbiased condition. Bond integrity is stressed in this environment.

Biased humidity

Moisture and bias used to accelerate corrosion-type failures in plastic packages. Conditions include 85°C ambient temperature with 85-percent relative humidity (RH). Typical bias voltage is +5 V and ground on alternating pins.

Autoclave (pressure cooker)

Plastic-packaged devices exposed to moisture at 121°C using a pressure of one atmosphere above normal pressure. The pressure forces moisture permeation of the package and accelerates corrosion mechanisms (if present) on the device. External package contaminates can also be activated and caused to generate inter-pin current leakage paths.

Temperature cycle

Device exposed to severe temperature extremes in an alternating fashion (-65°C for 15 minutes and 150°C for 15 minutes per cycle) for at least 1000 cycles. Package strength, bond quality, and consistency of assembly process are stressed in this environment.

Thermal shock

Test similar to the temperature cycle test, but involving a liquid-to-liquid transfer, per MIL-STD-883C, Method 1011.

PIND

Particle Impact Noise Detection test. A non-destructive test to detect loose particles inside a device cavity.

Mechanical Sequence:

Fine and gross leak	Per MIL-STD-883C, Method 1014.5
Mechanical shock	Per MIL-STD-883C, Method 2002.3, 1500 g, 0.5 ms, Condition B
PIND (optional)	Per MIL-STD-883C, Method 2020.4
Vibration, variable frequency	Per MIL-STD-883C, Method 2007.1, 20 g, Condition A
Constant acceleration	Per MIL-STD-883C, Method 2001.2, 20 kg, Condition D, Y1 Plane min
Fine and gross leak	Per MIL-STD-883C, Method 1014.5
Electrical test	To data sheet limits

Thermal Sequence:

Fine and gross leak	Per MIL-STD-883C, Method 1014.5
Solder heat (optional)	Per MIL-STD-750C, Method 1014.5
Temperature cycle (10 cycles minimum)	Per MIL-STD-883C, Method 1010.5, -65 to +150°C, Condition C
Thermal shock (10 cycles minimum)	Per MIL-STD-883C, Method 1011.4, -55 to +125°C, Condition B
Moisture resistance	Per MIL-STD-883C, Method 1004.4
Fine and gross leak	Per MIL-STD-883C, Method 1014.5
Electrical test	To data sheet limits

Thermal/Mechanical Sequence:

Fine and gross leak	Per MIL-STD-883C, Method 1014.5
Temperature cycle (10 cycles minimum)	Per MIL-STD-883C, Method 1010.5, -65 to +150°C, Condition C
Constant acceleration	Per MIL-STD-883C, Method 2001.2, 30 kg, Y1 Plane
Fine and gross leak	Per MIL-STD-883C, Method 1014.5
Electrical test	To data sheet limits

Electrostatic discharge	Per MIL-STD-883C, Method 3015
Solderability	Per MIL-STD-883C, Method 2003.3
Solder heat	Per MIL-STD-750C, Method 2031, 10 sec
Salt atmosphere	Per MIL-STD-883C, Method 1009.4, Condition A, 24 hrs min
Lead pull	Per MIL-STD-883C, Method 2004.4, Condition A
Lead integrity	Per MIL-STD-883C, Method 2004.4, Condition B1
Electromigration	Accelerated stress testing of conductor patterns to ensure acceptable lifetime of power-on operation
Resistance to solvents	Per MIL-STD-883C, Method 2015.4

Table D-1. Microprocessor and Microcontroller Tests

TEST	DURATION	SAMPLE SIZE PLASTIC	SAMPLE SIZE CERAMIC
Operating life, 125°C, 5.0 V	1000 hrs	129	129
Operating life, 150°C, 5.0 V	1000 hrs	77*	77
Storage life, 150°C	1000 hrs	77	77
Biased 85°C/85 percent RH, 5.0 V	1000 hrs	129	-
Autoclave, 121°C, 1 ATM	240 hrs	77	-
Temperature cycle, -65 to 150°C	1000 cyc	129	129
Temperature cycle, 0 to 125°C	3000 cyc	129	129
Thermal shock, -65 to 150°C	200 cyc	129	129
Electrostatic discharge, ±2 kV		12	12
Latch-up (CMOS devices only)		5	5
Mechanical sequence		-	38
Thermal sequence		-	38
Thermal/mechanical sequence		-	38
PIND		-	45
Internal water vapor		-	3
Solderability		22	22
Solder heat		22	22
Resistance to solvents		15	15
Lead integrity		15	15
Lead pull		22	-
Lead finish adhesion		15	15
Salt atmosphere		15	15
Flammability (UL94-V0)		3	-
Thermal impedance		5	5

*If junction temperature does not exceed plasticity of package.

Table D-2 provides a list of the TMS320C1x devices, the approximate number of transistors, and the equivalent gates. The numbers have been determined from design verification runs.

Table D-2. TMS320C1x Transistors

DEVICE	# TRANSISTORS	# GATES
NMOS:		
TMS32010 (all speeds)	50K	17K
CMOS:		
TMS320C10 (all speeds)	58K	15K
TMS320C14 (all speeds)	122K	25K
TMS320E14 (all speeds)	125K	26K
TMS320C15 (all speeds)	110K	20K
TMS320E15 (all speeds)	113K	21K
TMS320C17 (all speeds)	115K	22K
TMS320E17 (all speeds)	118K	23K

Appendix E

Development Support/Part Order Information

This section provides development support information, device part numbers, and support tool ordering information for all TMS320C1x (first-generation TMS320) products. Extensive documentation, including application reports, user's guides, and textbooks, is available to support DSP design, research, and education. To order TMS320 literature, contact the TI Customer Response Center (CRC) hotline number via 1-800-232-3200. For more information about support products and documentation, refer to the *TMS320 Family Development Support Reference Guide* (literature number SPRU011A).

The nearest TI Field Sales Office can be contacted for support tool availability or further details (see list of offices and distributors at end of book). For technical support, contact the TMS320 DSP hotline via (713) 274-2320.

The major topics discussed in this section are listed below.

- Development Support (Section E.1 on page E-3)
  - TMS320C1x/TMS320C2x Assembly Language Tools
  - TMS320C1x Simulator
  - TMS320C1x Evaluation Module (EVM)
  - TMS320C1x Emulator (XDS/22)
  - TMS320C1x XDS/22 Upgrade Kit
  - TMS320 Third-party Support
  - TMS320 Analog Interface Board
  - TMS320E15 EPROM DSP Starter Kit
  - Digital Filter Design Package (DFDP)
  - DSP Software Library
  - TMS320 Bell 212A Modem Software
  - TMS320 DSP Hotline/Bulletin Board Service

- Part Order Information (Section E.2 on page E-12)
  - Device part numbers
  - Software and hardware support tools part numbers
  - Device and support tool prefix designators
  - Device and support tool nomenclature

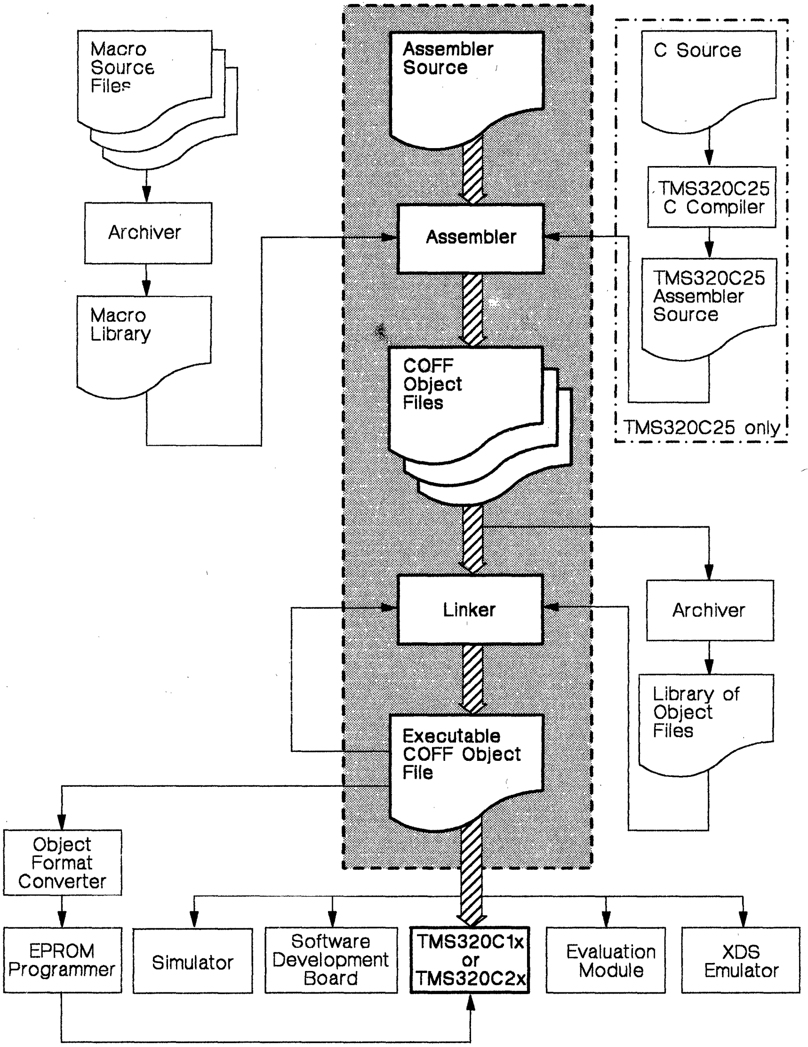


Figure E-2. TMS320C1x Development Tools

E.1 First-Generation TMS320 Development Support

Texas Instruments offers extensive development support and complete documentation with the first-generation TMS320 digital signal processors. Tools are provided to evaluate the performance of the processors, develop algorithm implementations, and fully integrate the design's software and hardware modules. Developmental operations are performed with the TMS320C1x/TMS320C2x Assembly Language Tools, Simulator, Evaluation Module (EVM), Emulator (XDS), and other support products.

A description and key features for each TMS320C1x development support tool is provided in the following subsections. For more information about support products, refer to the *TMS320 Family Development Support Reference Guide* (literature number SPRU011A). For ordering information, see Section E.2.

E.1.1 TMS320C1x/TMS320C2x Assembly Language Tools

The TMS320C1x/TMS320C2x Assembly language Tools generate the program code for the first- and second-generation TMS320 devices. This assembly language package consists of the following:

- An **Assembler** which translates assembly language source files into machine language object code in a common object file format (COFF).
- An **Archiver** which allows the programmer to collect a group of files into a single file or to produce a "library" of macros.
- A **Linker** which combines the object files into a single module for execution.
- A **Format Conversion Utility** which converts the files into a TI-tagged, Intel, or Tektronix object format.

Figure E-2 shows the developmental flowchart for the assembly language tools. The shaded area represents the basic routine for a software development. All devices which lie outside of this shaded portion are optional items.

The TMS320C1x/TMS320C2x Assembly Lanuguage Tools create and use the object files which are in the common object file format (COFF). This format is an improvement over those object codes which were developed by earlier macro assemblers. The COFF files provide more efficient programming for any TMS320C1x/E1x device since the programmer is allowed to divide and subdivide the program code into sections for modular manipulation/relocation.

> **Note:**
>
> The COFF files which are generated by the assembly language tools are not compatible with the TI-tagged, Intel, or Tektronix object files. The code conversion utility will convert COFF files into the standard format when using most EPROM programmers.

The assembly language tools are currently available for the IBM PC/MS-DOS and VAX/VMS operating systems.

E.1.2 TMS320C1x Simulator

The TMS320C1x CPU Simulator is a software program that simulates operation of the TMS320C1x CPU to allow program verification. The debug mode enables the user to monitor the state of the simulated TMS320C1x while the program is executing. The simulator uses the object code produced by the TMS320C1x Assembly Language Tool. During program execution, the internal registers and memory of the simulated device are modified as each instruction is interpreted by the host computer. Once program execution is suspended, the internal registers and both program and data memories can be inspected and/or modified. In addition, files can be associated with the I/O ports.

The following features highlight simulator capability for effective TMS320C1x software development:

- Program debug/verification
- Single-step option
- Trace/breakpoint capabilities
- Full access to simulated registers and memories
- I/O device simulation.

The simulator is currently available for the VAX/VMS and IBM PC/PC-DOS operating systems.

Note:

The TMS320C1x CPU Simulator only simulates the operation of the CPU, not the peripherals.

E.1.3 TMS320C1x Evaluation Module (EVM)

The TMS320C1x Evaluation Module (EVM) is a low-cost development board for TMS32010/C10/C15/E15 devices, used for full-speed in-circuit emulation and hardware debugging. (Note that the EVM does not support the TMS320C17/E17 devices.) It consists of a single board that enables a designer to evaluate certain characteristics of the processor to determine if it meets the requirements of an application.

The powerful firmware package of the TMS320C1x EVM contains a debug monitor, assembler/reverse assembler, and software communication via three EIA ports. The EVM can communicate to a host computer and several peripherals. The three EIA ports allow the EVM to communicate with a designer's terminal, a host computer, a printing device, or audio cassette. In addition, the EVM also supports an onboard PROM utility for programming TMS2764 EPROMs, used for mass program storage.

The EVM assembles source code created on a host computer or on the EVM's text editor, a line-numbered editor with character-editing capabilities. The EVM has a one-pass assembler, which resolves both forward and reverse labels and converts the incoming text into executable code. Object code produced by the EVM assembler is stored in memory. The reverse assembler converts object code back to assembly language mnemonics, and the patch assembler allows modification of the code.

Some key features of the TMS320C1x EVM are:

- On-board TMS32010
- 20-MHz operation
- Event counter for one breakpoint
- Text editor
- On-board EPROM programmer
- Audio cassette interface
- 4K words of on-board program RAM
- Target connector for full-speed in-circuit emulation from EVM memory
- Debug monitor including commands with full prompting
- Line-by-line assembler/reverse assembler
- Transparency mode for host CPU upload/download
- Eight instruction breakpoints available
- Single-step execution with software trace
- Standalone or host CPU configurable.

The TMS320C1x EVM functions in two modes: host computer mode or PC mode (single-user system). In the host computer mode, object and source code can be uploaded/downloaded between the host computer and EVM. In the PC mode, the EVM can support host uploads/downloads over a single port to allow a single-user system, such as an IBM PC, to function as both a terminal and a host (see Figure E-3). Commercially available terminal emulation software for the single-user system is required in this configuration.

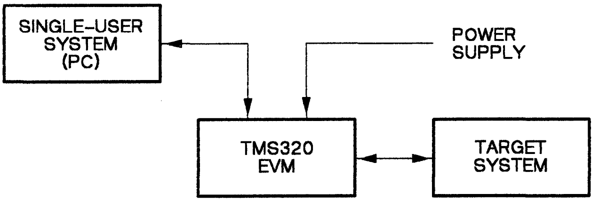


Figure E-3. TMS320C1x EVM/Single-User System

E.1.4 TMS320C1x Emulator (XDS)

The TMS320C1x Emulator (XDS/22) is a user-friendly system that has all the features necessary for realtime in-circuit emulation. This allows integration of hardware and software modules in the debug mode. By setting breakpoints based on internal conditions or external events, execution of the program can be suspended and control be given to the debug mode. In the debug mode, all registers and memory locations can be inspected and modified. Single-step execution is available. Full-trace capabilities at full speed and a reverse assembler that translates machine code back into assembly instructions also increase debugging productivity. Using a standard RS-232-C port, the object file is first produced by the TMS320C1x Assembly Language Tools, downloaded into the emulator, and finally controlled through a terminal.

The XDS/22 provides 4K x 16 words of high-speed static RAM (zero wait states) for program memory. It also has the capability of executing out of target memory to utilize the full TMS320C1x program/data address range. Foı multiprocessing configurations, up to nine emulators can be daisy-chained together.

The XDS/22 emulator is a completely self-contained system with power supply. With three RS-232-C ports, the XDS/22 Emulator can be interfaced to a terminal, host computer for source or object downloading/uploading capabilities, and printer or PROM programmer.

The TMS320C1x emulator supports in-circuit emulation on all speed versions of the TMS32010/C10/C15. Emulators are also available for supporting in-circuit emulations of the TMS320C14 and TMS320C17.

The key features of the TMS320C1x XDS/22 Emulator are as follows:

- Full-speed in-circuit emulation
- 4K words of program memory for user code
- Hardware breakpoint on program, data, or I/O conditions
- 2K words of full-speed hardware trace
- Use of target system crystal or internal crystal
- Up to ten software breakpoints
- Single-step option
- Assembler/reverse assembler
- Host-independent upload/download capabilities to/from program or data memory
- Ability to inspect and modify registers and program/data memory
- Multiprocessor system development.

Figure E-4 shows a block diagram of a typical system configuration using the TMS320C1x XDS/22 Emulator.

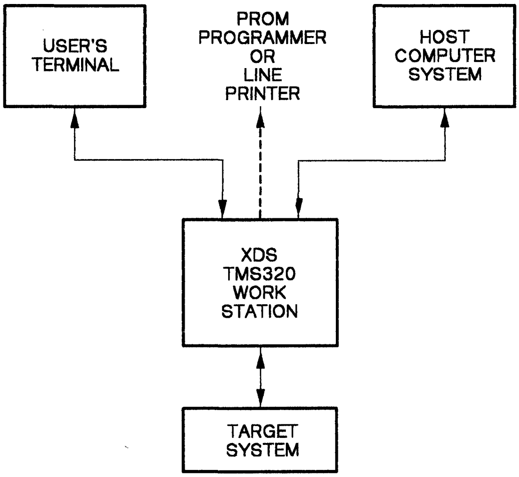


Figure E-4. TMS320C1x XDS/22 System Configuration

E.1.5 TMS320C1x XDS/22 Upgrade Kit

Texas Instruments offers a TMS320C1x XDS upgrade kit, which extends the functions and capabilities of existing development systems at a minimal cost to the customer. The upgrade kit will enable a TMS32010 XDS/22 to emulate operation of the TMS32010/C10/C15 devices. Note that early systems support TMS32010 and TMS320C10 performance. Upgrade kits allow upgrade only within a generation, not from a first- to a second-generation XDS.

E.1.6 TMS320 Third-party Support

The TMS320 family of digital signal processors is supported and serviced by many independent vendors and consultants, known as third parties. These products range from hardware to software, simulator to DPS utility package, or logic analyzer to emulator. The services range from simple speech-encoding or vector quantization to a more complex software/hardware design or system analysis.

The *TMS320 Family Development Support Reference Guide* (literature number SPRU011A) lists and describes a number of tools and services that augment the support that Texas Instruments provides; see Section 11. These publications furnish TI customers with additional information on supportive equipment and accessories.

E.1.7 TMS320 Analog Interface Board

Two TMS320 Analog Interface Boards (AIB1 and AIB2) are presently available for the first-generation TMS320 family. Both boards are capable of converting analog-to-digital/digital-to-analog signals. Either board can function as a preliminary target system with the TMS320C1x EVM, XDS, or another emulator. Figure E-5 shows the layout of a typical AIB system.

Each AIB board is an excellent educational tool which provides a simple, inexpensive method for learning the digital signal processing (DSP) techniques. And, either board allows testing of application programs with analog I/O by providing an interface to the TMS320C1x EVM or XDS/22.

Key features of the AIB1 are as follows:

- 12-bit analog-to-digital converter with sample and hold
- 12-bit digital-to-analog converter
- One 16-bit input port for additional A/D or user application
- One 16-bit output port for additional D/A or user application
- Two low-pass filters
- Audio amplifier
- TBLW (TABLE WRITE) decoder
- Extended I/O data memory
- Prototyping area for user application

Key features of the AIB2 are as follows:

- 16-bit analog-to-digital converter with sample and hold
- 16-bit digital-to-analog converter
- Supports TLC3204x Analog Interface chips and TCM2918 codec chips
- Stand-alone operation (dual 27xxx EPROM sockets and socket-type oscillator)
- On-board noise and function generator
- Sockets for TMS320C10/C15/C17/C25 devices
- Socket for applicable second-generation TMS320 members

The sample rate clock for each AIB is derived from an on-board oscillator and may be programmed to provide an periodic analog input, output, or both. There are two analog lowpass filters on the board, too. One filter minimizes the aliasing effects by limiting the band-width of the A/D input. The other filter smooths the output of the D/A. The frequency response of the filters is controlled by varying the external components in the filter stages. The cutoff of these filters is set to 4.7 kHz, but may be (plug) programmed. An audio amplifier that will drive an 8-ohm speaker is provided for applications with audio output. Sockets for 8K words of expansion memory are also provided. This memory is addressed through I/O and can support direct or autoincrement/decrement addressing. Up to 64K words of memory may be addressed through the memory expansion connector via this I/O interface.

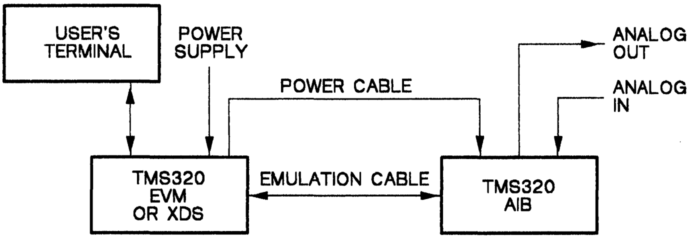


Figure E-5. TMS320 AIB System Configuration

E.1.8 TMS320E15 EPROM DSP Starter Kit

To assist with developing, debugging, and testing programs, Texas Instruments offers the TMS320E15 EPROM DSP Starter Kit. The kit includes the following:

- TMS320C1x Evaluation Module (EVM) to provide a standalone development system for the TMS32010/C10/C15/E15.
- Two TMS320E15JDL devices (TMS320EPROM/15 - EPROM DSP Twin-Pack), each of which provides an on-chip 256-word RAM and 4K-word program EPROM for realtime code development and modification. The device is object-code and pin compatible with the TMS32010/C10 and features EPROM code protection for copyright security.
- 40-pin to 28-pin conversion EPROM programmer adaptor socket (RTC/PGM320A-06) to facilitate the TMS320E15 with programming when using an EVM or a standard PROM programmer which is capable of programming the 28-pin 64K CMOS EPROMs.
- Documentation.

Contact the nearest TI Field Sales Office or distributor for availability or further information regarding the TMS320E15 EPROM DSP Starter Kit (part number RTC/EVM320E-15).

E.1.9 Digital Filter Design Package (DFDP)

Available from Atlanta Signal Processors, Inc. (ASPI), the The Digital Filter Design Package (DFDP) is a user-friendly, menu-driven software package. This package shortens the design time of various filter structures which use digital filters with floating-point accuracy or fixed-point economy. The package consists of four interactive filter design modules capable of performing the following functions:

1) Designing FIR filters (Kaiser window)
2) Designing FIR filters (Parks-McClellan)
3) Designing IIR filters (Butterworth, Chebychev I and II, and elliptic)
4) Generating TMS320C1x assembly code by converting the ASCII file which contains the filter coefficients into a fully commented assembly language code for TMS320C1x devices.

Cascade and parallel structures as well as higher-performance lattice, normalized lattice, and orthogonal forms are included in the modules.

The DFDP can design filters to meet any piecewise linear response specification, evaluate filter characteristics before and after coefficient quantization, and design special-purpose FIR filters, i.e., multi-band filters, differentiators, Hilbert transformers, and raised-cosine filters. The DFDP can also generate coefficients for filter implementations on any general-purpose processor or signal processing chip, as well as fully commented assembly language code for a variety of DSP chips. Magnitude, log magnitude, and impulse responses can be plotted for printer or screen display; in addition, the phase, group delay, and pole-zero map can be plotted for IIR filters. After the filter is designed, the user can generate code associated with the filter using the CGEN design module.

The DFDP runs on the IBM PS/2, IBM PC/XT/AT, and compatible systems. Operating systems must have 192K bytes of memory available. For more information, contact the nearest TI Field Sales Office. For details, contact Atlanta Signal Processors, Inc. via (404) 892-7265.

E.1.10 DSP Software Library

The Digital Signal Processing Software Library contains the major DSP routines (FFT, FIR/IIR filtering, and floating-point operations) and application algorithms (echo cancellation, ADPCM, and DTMF coding/decoding) as presented in the book, *Digital Signal Processing Applications with the TMS320 Family* (literature number SPRA012A). These routines and algorithms are written in TMS320C1x source code as well as TMS320C2x source code. In addition, macros for the TMS320C1x are included in this library.

The software package consists of four diskettes for use with the IBM PC/MS-DOS (version 1.1 or later) or a 1600 BPI magnetic tape for the VAX/VMS version. All the directories on the PC/MS-DOS version are contained on the magnetic tape for the VMS version. Each directory contains a README.LIS file briefly describing the contents of the files in the directory and the reference to the code. The book, *Digital Signal Processing Applications with the TMS320 Family* (literature number SPRA012A), serves as the major reference for theory and application of the algorithms; printed codes for the application reports are given in the appendices.

The library can also be ordered separately through TI (see Table E-2 for ordering information). All of the software in the library is copyrighted by Texas Instruments. The library is continually being updated; to obtain current information, contact TMS320 DSP Bulletin Board via (713) 274-2323.

E.1.11 TMS320 Bell 212A Modem Software

Texas Instruments is offering a software package containing source code and documentation for the design and implementation of a 1200-bps Bell 212A modem with the TMS320C17/E17 digital signal processor and the TMS7041 microcontroller.

The documentation included in the package consists of two reports. One report discusses in detail the theory behind the design of the modem, as well as the functions implemented. The second report describes the hardware, algorithms, and coding techniques used in the implementation of a Bell 212A modem demonstration unit. This implementation has been built and tested to verify its operation. After reading this report, the user should be able to design and build a similar unit as well as understand some tradeoffs involved in making custom modifications.

The source code for the TMS320 Bell 212A Modem Software package is provided on a 5 1/4" floppy for PC/MS-DOS or compatible operating systems. Contact the nearest TI Field Sales Office for further information.

E.1.12 TMS320 DSP Hotline/Bulletin Board Service

The TMS320 group at Texas Instruments provides a DSP Hotline to answer TMS320 technical questions, i.e., device problems, development tools, documentation, upgrades, and new TMS320 products. The hotline operates five days a week from 8:00 AM to 6:00 PM Central Time. The commercial telephone number is (713) 274-2320. To order literature, call the Customer Response Center (CRC) at 1-800-232-3200. Additionally, the TMS320 DSP maintains a facsimile (FAX) hotline which may be used for technical questions and other information; the FAX hotline number is (713) 274-2324. For details and availability of TMS320 devices or development tools, contact the nearest TI Field Sales Office.

The TMS320 DSP Bulletin Board Service is a telephone-line computer bulletin board that provides access to information pertaining to TMS320 devices. Specification updates for current or new TMS320 devices and development tools are communicated via the bulletin board as the information becomes available. The Bulletin Board Service can be accessed by dialing (713) 274-2323 with a 2400-, 1200-, or 300-bps modem.

The bulletin board contains TMS320 source code from the application reports included in the book, *Digital Signal Processing Applications with the TMS320 Family* (literature number SPRA012A). The bulletin board also provides new DSP applications software as it becomes available. See the *TMS320 Family Development Support Reference Guide* (literature number SPRU011A) for information on how to access the bulletin board.

E.2 Part Order Information

This section provides the device and support tool part numbers. Table E-1 lists the part numbers for all the first-generation members of the TMS320 family. Table E-2 gives ordering information for TMS320C1x hardware and software support tools. Table E-3 provides a list and description of the development tool connections to a target system. A discussion of the TMS320 family device and development support tool prefix and suffix designators is included to assist in understanding the TMS320 product numbering system.

Table E-1. TMS320C1x Digital Signal Processor Part Numbers

DEVICE NAME	TECHNOLOGY	OPERATING FREQUENCY	PACKAGE TYPE	TYPICAL POWER
TMS32010NL	2.4-μm NMOS	20 MHz†	Plastic 40-pin DIP	900 mW
TMS320C10FNL TMS320C10FNL25	2.0-μm CMOS 2.0-μm CMOS	20 MHz 25 MHz	Plastic 44-lead PLCC	165 mW 200 mW
TMS320C10NA	2.0-μm CMOS	20 MHz	Plastic 40-pin DIP	165 mW
TMS320C10NL TMS320C10NL14 TMS320C10NL25	2.0-μm CMOS 2.0-μm CMOS 2.0-μm CMOS	20 MHz† 14 MHz 25 MHz	Plastic 40-pin DIP	165 mW 140 mW 200 mW
TMS320C14FNL	1.6-μm CMOS	25 MHz‡	Plastic 68-lead PLCC	275 mW
TMS320E14FZL	1.6-μm CMOS	25 MHz‡	Plastic 68-lead CER-QUAD	325 mW
TMS320C15FNL TMS320C15FNL25	1.8-μm CMOS 1.8-μm CMOS	20 MHz 25 MHz	Plastic 44-lead PLCC	165 mW 200 mW
TMS320C15NL TMS320C15NL25	1.8-μm CMOS 1.8-μm CMOS	20 MHz‡ 25 MHz	Plastic 40-pin DIP	165 mW 200 mW
TMS320E15FZL TMS320E15FZL25	2.0-μm CMOS 2.0-μm CMOS	20 MHz 25 MHz	Ceramic 44-lead CER-QUAD	275 mW 325 mW
TMS320E15JDA	2.0-μm CMOS	20 MHz	Ceramic 44-pin DIP	275 mW
TMS320E15JDL TMS320E15JDL25	2.0-μm CMOS 2.0-μm CMOS	20 MHz 25 MHz	Ceramic 40-pin DIP	275 mW 325 mW
TMS320C17FNL	1.8-μm CMOS	20 MHz	Plastic 44-lead PLCC	250 mW
TMS320C17NL	1.8-μm CMOS	20 MHz‡	Plastic 40-pin DIP	250 mW
TMS320E17FZL	2.0-μm CMOS	20 MHz	Ceramic 44-lead CER-QUAD	275 mW
TMS320E17JDA	2.0-μm CMOS	20 MHz	Ceramic 40-pin DIP	275 mW
TMS320E17JDL	2.0-μm CMOS	20 MHz	Ceramic 40-pin DIP	275 mW

†Military version available.

‡Military versions planned; contact TI Field Sales Office for availability.

Table E-2. TMS320C1x Support Tool Part Numbers

TOOL DESCRIPTION	OPERATING SYSTEM	PART NUMBER
SOFTWARE		
Macro Assembler/Linker	PC/MS-DOS VAX/VMS VAX ULTRIX SUN-3 UNIX	TMDS3242850-02 TMDS3242250-08 TMDS3242260-08 TMDS3242550-08
Simulator	PC/MS-DOS VAX/VMS	TMDS3240811-02 TMDS3240211-08
Digital Filter Design Package	IBM PC-DOS	DFDP/IBM002
DSP Software Library	PC/MS-DOS VAX/VMS	TMDC3240812-12 TMDC3240212-18
TMS320 Bell 212A Modem Software	PC/MS-DOS	TMDX3240813-12
Data Encryption Standard Software	PC/MS-DOS	TMDX3240814-12
HARDWARE		
Evaluation Module (EVM)		RTC/EVM320A-03
XDS/22 Emulator TMS320C10/C15 TMS320C14 TMS320C17		 TMDS3262211 TMDX3262214 TMDX3262217
XDS/22 Upgrade Kit TMS32010 to TMS320C10/C15 TMS320C10/C15 to TMS320C14 TMS320C10/C15 to TMS320C17		 TMDS3282215 TMDX3285010 and TMDX3285018 TMDX3285014 and TMDX3285018
Analog Interface Board 1 (AIB1) Analog Interface Board 2 (AIB2)		RTC/EVM320C-06 RTC/AIB320A-06
EPROM DSP Starter Kit (TMS320E15)		RTC/EVM320E-15
EPROM Programmer Adaptor Socket 40- to 28-pin (TMS320E15/E17) 44- to 28-pin (TMS320E15/E17) 68- to 28-pin (TMS320E14) Additional Target Connector 44-lead PLCC (TMS320C10)		 RTC/PGM320A-06 RTC/PGM320C-06 TMDX3270110 TMDX3288810

Table E-3. Development Tool Connections to a Target System

TOOL	TARGET CONN.	INCL.	OPT.	PART NUMBER
TMS320C10 XDS/22	40-pin DIP 44-lead PLCC	X	 X	 TMDS3288810
TMS320C10 XDS/22 (Upgrade Kit)	40-pin DIP 44-lead PLCC	X	 X	 TMDS3288810
TMS320C14 XDS/22	68-lead PLCC	X		TMDX3262214
TMS320C17 XDS/22	40-pin DIP 44-lead PLCC	X	 X	 TMDS3288810
TMS32010 EVM	40-pin DIP	X		RTC/EVM320A-03

E.2.1 Device and Development Support Tool Prefix Designators

To assist the user in understanding the stages in the product development cycle, Texas Instruments assigns prefix designators to the part numbers of all TMS320 devices and support tools. Each TMS320 member will have one of three prefix designators: TMX, TMP, and TMS. TI recommends two of three possible prefix designators for its support tools: TMDX and TMDS. These prefixes represent one of the evolutionary stages of product development from engineering prototypes (TMX/TMDX) through fully qualified production devices/tools (TMS/TMDS). This development flow is defined below.

Device Development Evolutionary Flow:

TMX Experimental device that is not necessarily representative of the final device's electrical specifications.

TMP Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification.

TMS Fully qualified production device.

Support Tool Development Evolutionary Flow:

TMDX Development support product that has not yet completed Texas Instruments internal qualification testing.

TMDS Fully qualified development support product.

TMX and TMP devices and TMDX development support tools are shipped against the following disclaimer:

> "Developmental product is intended for internal evaluation purposes."

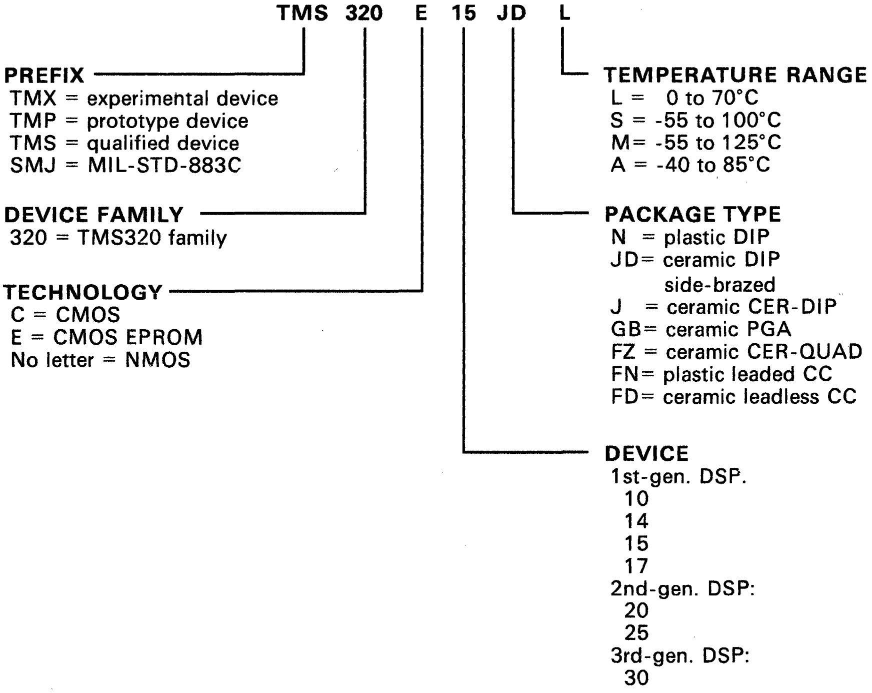
TMS devices and TMDS development support tools have been fully characterized and the quality and reliability of the device has been fully demonstrated. Texas Instruments standard warranty applies.

Note:

Predictions show that prototype devices (TMX or TMP) will have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system since their expected end-use failure rate is still undefined. Only qualified production devices are to be used.

E.2.2 Device and Development Support Tool Nomenclature

In addition to the prefix, the device nomenclature includes a suffix that follows the device family name. This suffix indicates the package type (e.g., N, FN, or GB) and temperature range (e.g., L). Figure E-6 provides a legend for reading the complete device name for any TMS320 family member.



TMS 320 E 15 JD L

PREFIX
TMX = experimental device
TMP = prototype device
TMS = qualified device
SMJ = MIL-STD-883C

DEVICE FAMILY
320 = TMS320 family

TECHNOLOGY
C = CMOS
E = CMOS EPROM
No letter = NMOS

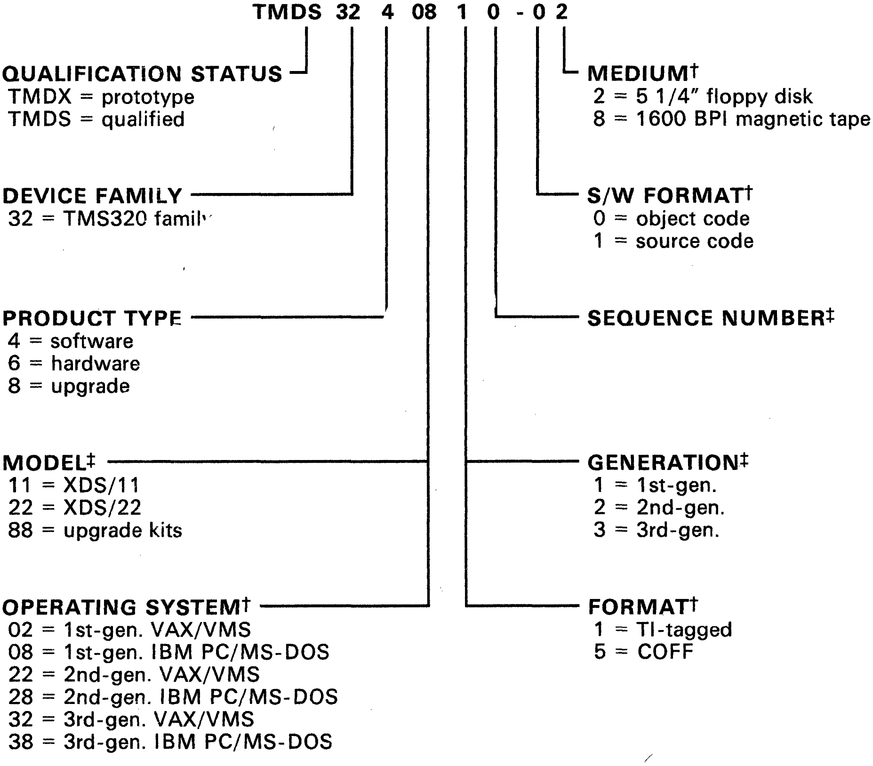
TEMPERATURE RANGE
L = 0 to 70°C
S = -55 to 100°C
M= -55 to 125°C
A = -40 to 85°C

PACKAGE TYPE
N = plastic DIP
JD= ceramic DIP side-brazed
J = ceramic CER-DIP
GB= ceramic PGA
FZ = ceramic CER-QUAD
FN= plastic leaded CC
FD= ceramic leadless CC

DEVICE
1st-gen. DSP.
10
14
15
17
2nd-gen. DSP:
20
25
3rd-gen. DSP:
30

Figure E-6. TMS320 Device Nomenclature

Figure E-7 provides a legend for reading the part number for any TMS320 hardware or software development tool.



TMDS 32 4 08 1 0 - 0 2

QUALIFICATION STATUS
TMDX = prototype
TMDS = qualified

DEVICE FAMILY
32 = TMS320 family

PRODUCT TYPE
4 = software
6 = hardware
8 = upgrade

MODEL‡
11 = XDS/11
22 = XDS/22
88 = upgrade kits

OPERATING SYSTEM†
02 = 1st-gen. VAX/VMS
08 = 1st-gen. IBM PC/MS-DOS
22 = 2nd-gen. VAX/VMS
28 = 2nd-gen. IBM PC/MS-DOS
32 = 3rd-gen. VAX/VMS
38 = 3rd-gen. IBM PC/MS-DOS

MEDIUM†
2 = 5 1/4" floppy disk
8 = 1600 BPI magnetic tape

S/W FORMAT†
0 = object code
1 = source code

SEQUENCE NUMBER‡

GENERATION‡
1 = 1st-gen.
2 = 2nd-gen.
3 = 3rd-gen.

FORMAT†
1 = TI-tagged
5 = COFF

† Software only.
‡ Hardware only.

Figure E-7. TMS320 Development Tool Nomenclature

Appendix F

Memories, Analog Converters, Sockets, and Crystals

This appendix provides product information regarding memories, analog converters, and sockets, which are manufactured by Texas Instruments and compatible with the TMS320C1x. Information is also given regarding crystal frequencies, specifications, and vendors.

The contents of the major areas in this appendix are listed below.

- TI Memories and Analog Converters (Section F.1 on page F-2)
  - EPROM memories
  - Codecs and filters
  - Analog interface circuits
  - A/D and D/A converters.
- TI Sockets for DIP and PLCC Packages (Section F.2 on page F-139)
  - Production sockets
  - Burn-in/test sockets.
- Crystals (Section F.3 on page F-144)
  - Commonly used crystal frequencies
  - Crystal specification requirements
  - Vendors of suitable crystals.

F.1 TI Memories and Analog Converters

This section provides pages of product information taken from data sheets for EPROM memories, codecs, analog interface circuits, and D/A and D/A converters.

All of these devices can be interfaced with TMS320C1x processors (see Section 6 for hardware interface designs). Refer to *Digital Signal Processing Applications with the TMS320 Family* (literature number SPRA012A) for additional information on interfaces using memories and analog conversion devices.

The following paragraphs give the name of each device and where the data sheet for that device is located in order to obtain further specification information if desired.

Data sheets for EPROM memories are located in the *MOS Memory Data Book* (SMYD006). The name of the device and the page number in the book on which the device is introduced are listed.

TMS27C64	(page 6-55)
TMS27C128	(page 6-79)
TMS27C256	(page 6-91)
TMX27C512	(page 6-105)

Another EPROM memory, TMS27C291/292, is described in a data sheet (SMLS291A).

The TCM29C13/14/16/17 codecs and filters are described in the data sheet beginning on page 2-111 of the *Telecommunications Circuits Data Book* (SCT001). An analog interface for the DSP using a codec and filter is provided by the TCM29C18/19 (data sheet number SCT021).

The data sheet for the TLC32040 analog interface circuit is provided in the *Interface Circuits Data Book* (SLYD002); see page 2-271.

In the same book, data sheets for A/D and D/A converters can be found. The name of the device and the introductory pages are as follows:

TLC0820	(page 2-113)
TLC1205/1225	(page 2-181)
TLC7524	(page 2-243)

TMS27C64 65,536-BIT UV ERASABLE PROGRAMMABLE READ-ONLY MEMORY
TMS27PC64 65,536-BIT PROGRAMMABLE READ-ONLY MEMORY

NOVEMBER 1985—REVISED APRIL 1988

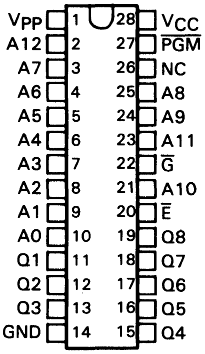
This Data Sheet is Applicable to All TMS27C64s and TMS27PC64s Symbolized with Code "A" as Described on Page 12.

- **Organization . . . 8K × 8**
- **Single 5-V Power Supply**
- **Pin Compatible with Existing 64K MOS ROMs, PROMs, and EPROMs**
- **All Inputs/Outputs Fully TTL Compatible**
- **Max Access/Min Cycle Times**

V_{CC} ±5%	V_{CC} ±10%	
'27C64-100		100 ns
'27C/PC64-120	'27C/PC64-12	120 ns
'27C/PC64-1	'27C/PC64-15	150 ns
'27C/PC64-2	'27C/PC64-20	200 ns
'27C/PC64	'27C/PC64-25	250 ns

- **Power Saving CMOS Technology**
- **Very High-Speed SNAP! Pulse Programming or Fast Programming Algorithms**
- **3-State Output Buffers**
- **400 mV Guaranteed DC Noise Immunity with Standard TTL Loads**
- **Latchup Immunity of 250 mA on All Input and Output Lines**
- **Low Power Dissipation (V_{CC} = 5.25 V)**
  — Active . . . 158 mW Worst Case
  — Standby . . . 1.4 mW Worst Case (CMOS Input Levels)
- **PEP4 Version Available with 168 Hour Burn-In, and also Extended Guaranteed Operating Temperature Ranges**

J & N PACKAGE
(TOP VIEW)



PIN NOMENCLATURE	
A0-A12	Address Inputs
$\overline{E}$	Chip Enable Power Down
$\overline{G}$	Output Enable
GND	Ground
NC	No Connection
NU	Make No External Connection
$\overline{PGM}$	Program
Q1-Q8	Outputs
V_{CC}	5-V Power Supply
V_{PP}	12-13 V Programming Power Supply

description

The TMS27C64 series are 65,536-bit, ultraviolet-light erasable, electrically programmable read-only memories.

The TMS27PC64 series are 65,536-bit, one-time, electrically programmable read-only memories.

These devices are fabricated using power saving CMOS technology for high-speed and simple interface with MOS and bipolar circuits. All inputs (including program data inputs) can be driven by Series 74 TTL circuits without the use of external pull-up resistors. Each output can drive one Series 74 TTL circuit without external resistors.

The data outputs are three-state for connecting multiple devices to a common bus. The TMS27C64 and the TMS27PC64 are pin compatible with 28-pin 64K MOS ROMs, PROMs, and EPROMs.

TEXAS INSTRUMENTS
POST OFFICE BOX 1443 • HOUSTON, TEXAS 77001

TMS27C128 131,072-BIT UV ERASABLE PROGRAMMABLE READ-ONLY MEMORY
TMS27PC128 131,072-BIT PROGRAMMABLE READ-ONLY MEMORY

OCTOBER 1984 — REVISED FEBRUARY 1989

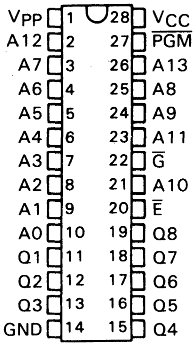
This Data Sheet is Applicable to All TMS27C128s and TMS27PC128s Symbolized with Code "A" as Described on Page 11.

- **Organization . . . 16K × 8**
- **Single 5-V Power Supply**
- **Pin Compatible with Existing 128K MOS ROMs, PROMs, and EPROMs**
- **All Inputs/Outputs Fully TTL Compatible**
- **Max Access/Min Cycle Times**

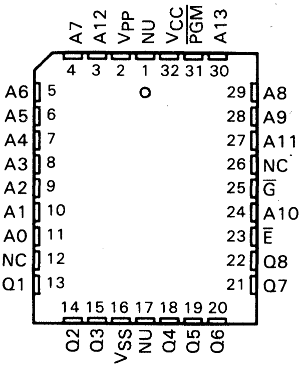
V_{CC} ±5%	V_{CC} ±10%	
'27C128-100		100 ns
'27C128-120	'27C128-12	120 ns
'27C/PC128-1	'27C/PC128-15	150 ns
'27C/PC128-2	'27C/PC128-20	200 ns
'27C/PC128	'27C/PC128-25	250 ns

- **Power Saving CMOS Technology**
- **Very High-Speed SNAP! Pulse Programming or Fast Programming Algorithms**
- **3-State Output Buffers**
- **400 mV Guaranteed DC Noise Immunity with Standard TTL Loads**
- **Latchup Immunity of 250 mA on All Input and Output Lines**
- **Low Power Dissipation (V_{CC} = 5.25 V)**
  — Active . . . 158 mW Worst Case
  — Standby . . . 1.4 mW Worst Case (CMOS Input Levels)
- **PEP4 Version Available with 168 Hour Burn-in, and also Guaranteed Operating Temperature Ranges**
- **128K EPROM Available with MIL-STD-883C Class B High Reliability Processing (SMJ27C128)**

J AND N PACKAGES
(TOP VIEW)



FM PACKAGE
(TOP VIEW)



PIN NOMENCLATURE	
A0-A13	Address Inputs
E̅	Chip Enable/Power Down
G̅	Output Enable
GND	Ground
NC	No Connection
NU	Make No External Connection
PGM̅	Program
Q1-Q8	Outputs
V_{CC}	5-V Power Supply
Vpp	12-13 V Programming Power Supply

description

The TMS27C128 series are 131,072-bit, ultraviolet-light erasable, electrically programmable read-only memories.

The TMS27PC128 series are 131, 072-bit, one-time, electrically programmable read-only memories.

These devices are fabricated using power-saving CMOS technology for high speed and simple interface with MOS and bipolar circuits. All inputs

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS INSTRUMENTS
POST OFFICE BOX 1443 • HOUSTON, TEXAS 77001

TMS27C256 262,144-BIT UV ERASABLE PROGRAMMABLE READ-ONLY MEMORY
TMS27PC256 262,144-BIT PROGRAMMABLE READ-ONLY MEMORY

SEPTEMBER 1984 — REVISED FEBRUARY 1989

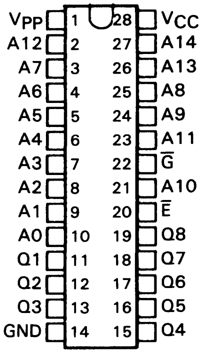
This Data Sheet is Applicable to All TMS27C256s and TMS27PC256s Symbolized with Code "A" as Described on Page 12.

- **Organization . . . 32K × 8**
- **Single 5-V Power Supply**
- **Pin Compatible with Existing 256K MOS ROMs, PROMS, and EPROMs**
- **All Inputs/Outputs Fully TTL Compatible**
- **Max Access/Min Cycle Times**

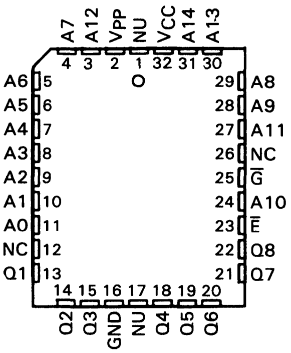
V_{CC} ±5%	V_{CC} ±10%	
'27C256-120	'27C256-12	120 ns
'27C/PC256-150	'27C/PC256-15	150 ns
'27C/PC256-1	'27C/PC256-17	170 ns
'27C/PC256-2	'27C/PC256-20	200 ns
'27C/PC256	'27C/PC256-25	250 ns

- **Power Saving CMOS Technology**
- **Very High Speed SNAP! Pulse Programming or Fast Programming Algorithms**
- **3-State Output Buffers**
- **400 mV Guaranteed DC Noise Immunity with Standard TTL Loads**
- **Latchup Immunity of 250 mA on All Input and Output Lines**
- **Low Power Dissipation (V_{CC} = 5.25 V)**
  — Active . . . 158 mW Worst Case
  — Standby . . . 1.4 mW Worst Case
  (CMOS-Input Levels)
- **PEP4 Version Available with 168 Hour Burn-in, and also Guaranteed Operating Temperature Ranges**
- **256K EPROM Available with MIL-STD-883C Class B High Reliability Processing (SMJ27C256)**

J AND N PACKAGES
(TOP VIEW)



FM PACKAGE
(TOP VIEW)



PIN NOMENCLATURE	
A0-A14	Address Inputs
E̅	Chip Enable/Power Down
G̅	Output Enable
GND	Ground
NC	No Connection
NU	Make No External Connecction
Q1-Q8	Outputs
V_{CC}	5-V Power Supply
V_{PP}	12-13 V Programming Power Supply

description

The TMS27C256 series are 262,144-bit, ultraviolet-light erasable, electrically programmable read-only memories.

The TMS27PC256 series are 262,144-bit, one-time, electrically programmable read-only memories.

These devices are fabricated using power saving CMOS technology for high speed and simple interface with MOS and bipolar circuits. All inputs (including program data inputs) can

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TMS27C512 524,288-BIT ERASABLE PROGRAMMABLE READ-ONLY MEMORY
TMS27PC512 524,288-BIT PROGRAMMABLE READ-ONLY MEMORY

NOVEMBER 1985 — REVISED FEBRUARY 1989

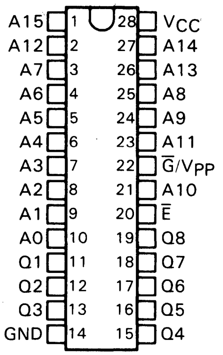
This Data Sheet is Applicable to All TMS27C512s and TMS27PC512s Symbolized with Code "A" as Described on Page 12.

- **Organization . . . 64K × 8**
- **Single 5-V Power Supply**
- **Pin Compatible with Existing 512K MOS ROMs, PROMs, and EPROMs**
- **All Inputs/Outputs Fully TTL Compatible**
- **Max Access/Min Cycle Time**

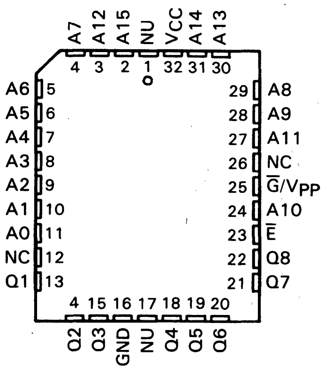
V_{CC} ±5%	V_{CC} ±10%	
'27C/PC512-150	'27C/PC512-15	150 ns
'27C/PC512-1	'27C/PC512-17	170 ns
'27C/PC512-2	'27C/PC512-20	200 ns
'27C/PC512	'27C/PC512-25	250 ns
'27C/PC512-3	'27C/PC512-30	300 ns

- **Power Saving CMOS Technology**
- **Very High-Speed SNAP! Pulse Programming or Fast Programming Algorithms**
- **3-State Output Buffers**
- **400 mV Guaranteed DC Noise Immunity with Standard TTL Loads**
- **Latchup Immunity of 250 mA on All Input and Output Lines**
- **Low Power Dissipation (V_{CC} = 5.25 V)**
  - **Active . . . 158 mW Worst Case**
  - **Standby . . . 1.4 mW Worst Case (CMOS Input Levels)**
- **PEP4 Version Available with 168 Hour Burn-in, and also Guaranteed Operating Temperature Ranges**
- **512K EPROM Available with MIL-STD-883C Class B High Reliability Processing (SMJ27C512)**

J AND N PACKAGE
(TOP VIEW)



FM PACKAGE
(TOP VIEW)



PIN NOMENCLATURE	
A0-A15	Address Inputs
E̅	Chip Enable/Power Down
G̅/V_{PP}	12-13 V Programming Power Supply
GND	Ground
NC	No Connection
NU	Make No External Connection
Q1-Q8	Outputs
V_{CC}	5-V Power Supply

description

The TMS27C512 series are 524,288-bit, ultraviolet-light erasable, electrically programmable read-only memories.

The TMS27PC512 series are 524, 288-bit, one-time, electrically programmable read-only memories.

These devices are fabricated using power-saving CMOS technology for high speed and simple interface with MOS and bipolar circuits. All inputs (including program data inputs) can be driven by Series 74 TTL circuits without the use of external pull-up resistors. Each output can drive one Series 74 TTL circuit without external resistors.

TMS27C291, TMS27C292 16,384-BIT UV ERASABLE PROGRAMMABLE READ-ONLY MEMORIES
TMS27PC291 16,384-BIT PROGRAMMABLE READ-ONLY MEMORY

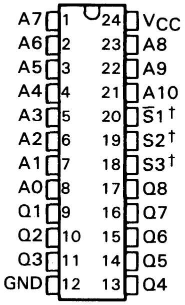
SEPTEMBER 1986—REVISED APRIL 1988

- **Organization . . . 2K × 8**
- **Single 5-V Power Supply**
- **Pin Compatible with Existing 2K × 8 Bipolar/High-Speed CMOS EPROMs and PROMs**
- **All Inputs/Outputs TTL Compatible**
- **High Speed**
- **Max Access/Min Cycle Time**

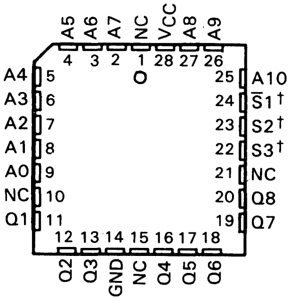
	V_{CC} ± 5%	
'27C/PC291-3	**'27C292-3**	**35 ns**
'27C/PC291	**'27C292**	**45 ns**
'27C/PC291-5	**'27C292-5**	**50 ns**
	V_{CC} ± 10%	
'27C/PC291-35	**'27C292-35**	**35 ns**
'27C/PC291-45	**'27C292-45**	**45 ns**
'27C/PC291-50	**'27C292-50**	**50 ns**

- **Low-Power CMOS Technology**
- **3-State Output Buffers**
- **Low Power Dissipation (V_{CC} = 5.25 V) — Active . . . 394 mW Max**
- **Erasable**
- **100% Pretestable**

J AND N PACKAGE
(TOP VIEW)



FN PACKAGE
(TOP VIEW)



†These pins have different pin assignments and functions in the program mode (see page 3).

description

The TMS27C291 and TMS27C292 series are 16,384-bit, ultraviolet-light erasable, electrically programmable read-only memories. The TMS27PC291 series are 16,384-bit, one-time, electrically programmable read-only memories. These devices are fabricated using CMOS technology for high speed and simple interface with MOS and bipolar circuits. All inputs (including program data inputs) can be driven by Series 74 TTL circuits without the use of external resistors. Each output can drive eight Series 74 TTL circuits without external resistors. The data outputs are three-state for connecting multiple devices to a common bus. The J and N dual-in-line packages are pin compatible with existing 24-pin bipolar PROMs and high speed EPROMs.

READ MODE

PIN NOMENCLATURE	
A0-A10	Address Inputs
GND	Ground
NC	No Connection
Q1-Q8	Outputs
S̄1, S2, S3	Chip Selects
V_{CC}	5-V Power Supply

The TMS27C291 and TMS27C292 are offered in dual-in-line ceramic packages (J suffix). The TMS27C291 ceramic package is designed for insertion in mounting-hole rows on 7,62-mm (300-mil) centers. The TMS27C292 ceramic package is designed for insertion in mounting-hole rows on 15,24-mm (600-mil) centers.

ADVANCE INFORMATION

ADVANCE INFORMATION concerns new products in the sampling or preproduction phase of development. Characteristic data and other specifications are subject to change without notice.

The TMS27PC291 PROM is offered in dual-in-line plastic package (N suffix) designed for insertion in mounting-hole rows on 7,62-mm (300-mil) centers. This version of the device is still in development, and the ADVANCE INFORMATION notices in this data sheet pertain to the N package devices. The TMS27PC291 PROM is also offered in a 28-lead plastic-leaded chip carrier (FN suffix) for surface mounting applications on solder lands on 1,27-mm (50-mil) centers.

All devices are guaranteed for operation from 0°C to 70°C.

operation

There are eight modes of operation for the TMS27C291, TMS27C292 and the TMS27PC291 as listed in the following table. The read mode requires a single 5-V supply. All inputs are TTL or CMOS levels except for V_{PP} during programming (13.5 V).

FUNCTION	MODE										
	Read	Output Disable#	Output Disable#	Output Disable#	Program Verify	Program Inhibit	Fast Program	Blank Check Ones	Blank Check Zeros	Signature	
$\overline{S1}$/V_{PP}†	V_{IL}	V_{IH}	X‡	X	V_{PP}	V_{PP}	V_{PP}	$V_{IL(P)}$¶	$V_{IL(P)}$	V_{IL}	
S2/$\overline{VFY}$†	V_{IH}	X	V_{IL}	X	$V_{IL(P)}$	$V_{IH(P)}$	$V_{IH(P)}$	$V_{IL(P)}$	$V_{IH(P)}$	V_{IH}	
S3/$\overline{PGM}$†	V_{IH}	X	X	V_{IL}	$V_{IH(P)}$	$V_{IH(P)}$	$V_{IL(P)}$	V_H§	V_H	V_H	
V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	
A9	X	X	X	X	X	X	X	X	X	V_{PP}	V_{PP}
A0	X	X	X	X	X	X	X	X	X	V_{IL}	V_{IH}
Q1-Q8	D_{OUT}	HI-Z	HI-Z	HI-Z	D_{OUT}	HI-Z	D_{IN}	Ones	Zeros	CODE	
										MFG	DEV
										97	02

†Pin assignment for program mode.
‡X can be V_{IL} or V_{IH}.
§V_H = 12 V ± 0.5 V.
¶(P) = Programming mode.
#Output can be disabled using any of these three methods.

read/output disable

When the outputs of two or more of these devices are connected in parallel on the same bus, the output of any particular device in the circuit can be read with no interference from competing outputs of the other devices. To read the output of a '27C291, '27PC291, or '27C292, a low-level signal is applied to $\overline{S1}$ and a high-level signal is applied to S2 and S3. Any other combination of logic states on these three inputs will disable the outputs. Output data is accessed at pins Q1 through Q8.

latchup immunity

Latchup immunity is a minimum of 250 mA on all inputs and outputs. This feature provides latchup immunity beyond any potential transients at the P.C. board level when the devices are interfaced to industry-standard TTL or MOS logic devices. The input/output layout approach controls latchup without compromising performance or packing density.

TCM129C13, TCM129C14, TCM129C16, TCM129C17, TCM29C13, TCM29C14, TCM29C16, TCM29C17 COMBINED SINGLE-CHIP PCM CODEC AND FILTER

D2765, APRIL 1986 – REVISED JUNE 1988

- Replaces Use of TCM2910A in Tandem with TCM2912C
- Reliable Silicon-Gate CMOS Technology
- Low Power Consumption:
  Operating Mode . . . 80 mW Typical
  Power-Down Mode . . . 5 mW Typical
- Excellent Power Supply Rejection Ratio Over Frequency Range of 0 to 50 kHz
- No External Components Needed for Sample, Hold, and Auto-Zero Functions
- Precision Internal Voltage References
- Direct Replacement for Intel 2913, 2914, 2916, and 2917
- TCM29C13N-3 is Primarily Used for Low-Cost DSP Applications with TMS320CXX

FEATURE TABLE

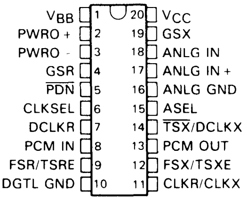
FEATURE	129C13 29C13	129C14 29C14	129C16 29C16	129C17 29C17
Number of Pins:				
24		X		
20	X			
16			X	X
μ-law/A-law Coding:				
μ-law	X	X	X	
A-law	X	X		X
Data Timing Rates:				
Variable Mode				
64 kHz to 2.048 MHz	X	X	X	X
Fixed Mode				
1.536 MHz	X	X		
1.544 MHz	X	X		
2.048 MHz	X	X	X	X
Loopback Test Capability		X		
8th-Bit Signaling				

description

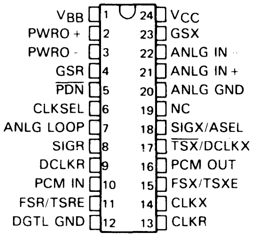
The TCM129C13, TCM129C14, TCM129C16, TCM129C17, TCM29C13, TCM29C14, TCM29C16, and TCM29C17 are single-chip pulse-code-modulated encoders and decoders (PCM codecs) and PCM line filters. These devices provide all the functions required to interface a full-duplex (4-wire) voice telephone circuit with a time-division-multiplexed (TDM) system. These devices are intended to replace the TCM2910A in tandem with the TCM2912C. Primary applications of the devices include:

- Line Interface for Digital Transmission and Switching of T1 Carrier, PABX, and Central Office Telephone Systems
- Subscriber Line Concentrators
- Digital Encryption Systems
- Digital Voice Band Data Storage Systems
- Digital Signal Processing

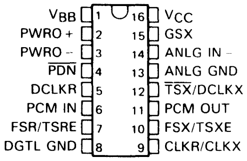
TCM129C13 . . . DW, DY, J, OR N PACKAGE
TCM29C13 . . . DW, DY, J, OR N PACKAGE
TCM29C13N-3 . . . N PACKAGE
(TOP VIEW)



TCM129C14 . . . DW OR JW PACKAGE
TCM29C14 . . . DW OR JW PACKAGE
(TOP VIEW)



TCM129C16, TCM129C17 . . . J OR N PACKAGE
TCM29C16, TCM29C17 . . . J OR N PACKAGE
(TOP VIEW)



Caution. These devices have limited built-in gate protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS INSTRUMENTS
POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

description (continued)

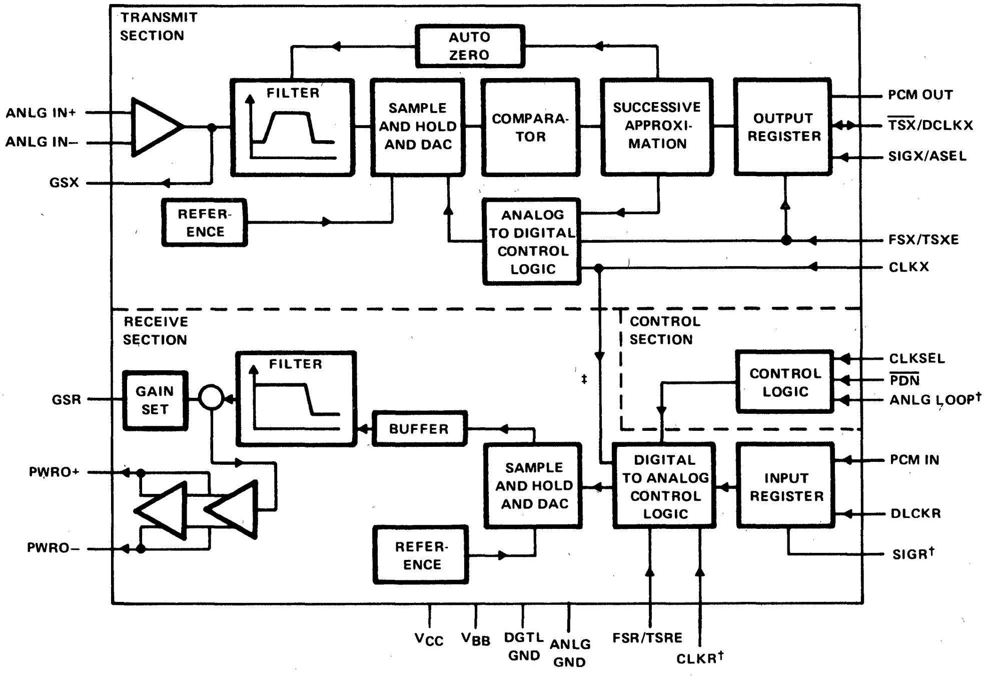
These devices are designed to perform the transmit encoding (A/D conversion) and receive decoding (D/A conversion) as well as the transmit and receive filtering functions in a pulse-code-modulated system. They are intended to be used at the analog termination of a PCM line or trunk.

The TCM129C13, TCM129C14, TCM129C16, TCM129C17, TCM29C13, TCM29C14, TCM29C16, and TCM29C17 provide the bandpass filtering of the analog signals prior to encoding and after decoding. These combination devices perform the encoding and decoding of voice and call progress tones as well as the signaling and supervision information.

The TCM29C13N-3 is the same as the TCM29C13N except for certain parameters as indicated in the specification section.

The TCM129C13, TCM129C14, TCM129C16, and TCM129C17 are characterized for operation from −40°C to 85°C. The TCM29C13, TCM29C14, TCM29C16, and TCM29C17 are characterized for operation from 0°C to 70°C.

functional block diagram



†TCM129C14 and TCM29C14 only
‡TCM129C13, TCM129C16, TCM129C17, TCM29C13, TCM29C16, and TCM29C17 only.

PIN			NAME	DESCRIPTION
TCM129C13 TCM29C13	TCM129C14 TCM29C14	TCM129C16 TCM129C17 TCM29C16 TCM29C17		
1	1	1	V_{BB}	Most negative supply voltage; input is −5 V ±5%.
2	2	2	PWRO +	Noninverting output of power amplifier. Can drive transformer hybrids or high-impedance loads directly in either a differential or a single-ended configuration.
3	3	3	PWRO −	Inverting output of power amplifier; functionally identical with and complementary to PWRO +.
4	4		GSR	Input to the gain-setting network on the output power amplifier. Transmission level can be adjusted over a 12-dB range depending upon the voltage at GSR.
5	5	4	$\overline{\text{PDN}}$	Power-down select. The device is inactive with a TTL low-level input to this pin and active with a TTL high-level input to the pin.
6	6		CLKSEL	Clock frequency selection. Input must be connected to V_{BB}, V_{CC}, or ground to reflect the master clock frequency. When tied to V_{BB}, CLK is 2.048 MHz. When tied to ground, CLK is 1.544 MHz. When tied to V_{CC}, CLK is 1.536 MHz.
	7		ANLG LOOP	Provides loopback test capability. When this input is high, PWRO + is internally connected to ANLG IN.
	8		SIGR	Signaling bit output, receive channel; in a fixed-data-rate mode, outputs the logical state of the 8th bit (LSB) of the PCM word in the most recent signaling frame.
7	9	5	DCLKR	Selects fixed or variable data-rate operation. When this pin is connected to V_{BB}, the device operates in the fixed-data-rate mode. When DCLKR is not connected to V_{BB}, the device operates in the variable-data-rate mode, and DCLKR becomes the receive data clock, which operates at frequencies from 64 kHz to 2.048 MHz
8	10	6	PCM IN	Receive PCM input. PCM data is clocked in on this pin on eight consecutive negative transitions of the receive data clock, which is CLKR in fixed-data-rate timing and DCLKR in variable-data-rate timing.
9	11	7	FSR/TSRE	Frame synchronization clock input/time slot enable for receive channel. In the fixed-data-rate mode, FSR distinguishes between signaling and non-signaling frames by a double- or single-length pulse, respectively. In the variable-data-rate mode, this signal must remain high for the duration of the timeslot. The receive channel enters the standby state when FSR is TTL low for 300 ms.
10	12	8	DGTL GND	Digital ground for all internal logic circuits. Not internally connected to ANLG GND.
11	13	9	CLKR	Receive master clock and data clock for the fixed-data-rate mode. Receive master clock only for variable-data-rate mode. CLKR and CLKX are internally connected together for TCM129C13, TCM129C16, TCM129C17, TCM29C13, TCM29C16, and TCM29C17.

PIN			NAME	DESCRIPTION
TCM129C13 TCM29C13	TCM129C14 TCM29C14	TCM129C16 TCM129C17 TCM29C16 TCM29C17		
11	14	9	CLKX	Transmit master clock and data clock for the fixed-data-rate mode. Transmit master clock only for variable data rate mode. CLKR and CLKX are internally connected for the TCM129C13, TCM129C16, TCM129C17, TCM29C13, TCM29C16, and TCM29C17.
12	15	10	FSX/TSXE	Frame synchronization clock input/time-slot enable for transmit channel. Operates independently of, but in an analagous manner to, FSR/TSRE. The transmit channel enters the standby state when FSX is low for 300 ms.
13	16	11	PCM OUT	Transmit PCM output. PCM data is clocked out on this output on eight consecutive positive transitions of the transmit data clock, which is CLKX in fixed-data-rate timing and DCLKX in variable-data-rate timing.
14	17	12	$\overline{\text{TSX}}$/DCLKX	Transmit channel time slot strobe (output) or data clock (input) for the transmit channel. In the fixed-data-rate mode, this pin is an open-drain output to be used as an enable signal for a three-state buffer. In the variable-data rate mode, DCLKX becomes the transmit data clock, which operates at TTL levels from 64 kHz to 2.048 MHz.
15	18		SIGX/ASEL	Used to select between A-law and μ-law operation. When connected to V_{BB}, A-law is selected. When connected to V_{CC} or ground, u-law is selected. When not connected to V_{BB}, it is a TTL-level input that is transmitted as the eighth bit (LSB) of the PCM word during signaling frames on the PCM OUT pin (TCM129C14 and TCM29C14 only). SIGX/ASEL is internally connected to provide μ-law operation for TCM129C16 and TCM29C16 and A-law operation for TCM129C17 and TCM29C17.
16	20	13	ANLG GND	Analog ground return for all internal voice circuits. Not internally connected to DGTL GND.
17	21		ANLG IN+	Noninverting analog input to uncommitted transmit operational amplifier. Internally connected to ANLG GND on TCM129C16, TCM29C16, TCM129C17, and TCM29C17.
18	22	14	ANLG IN−	Inverting analog input to uncommitted transmit operational amplifier.
19	23	15	GSX	Output terminal of internal uncommitted operational amplifier. Internally, this is the voice signal input to the transmit filter.
20	24	16	V_{CC}	Most positive supply voltage, input is 5 V ± 5%.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	−0.3 V to 15 V
Output voltage, V_O	−0.3 V to 15 V
Input voltage, V_I	−0.3 V to 15 V
Digital ground voltage	−0.3 V to 15 V
Continuous total dissipation at (or below) 25°C free-air temperature	1375 mW
Operating free-air temperature range: TCM129C__	−40°C to 85°C
TCM29C__	0°C to 70°C
Storage temperature range	−65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds: DW, DY, or N package	260°C
Lead temperature 1,6 mm (1/16 inch) from case for 60 seconds: J or JW package	300°C

NOTES: 1. Voltage values for maximum ratings are with respect to V_{BB}.

recommended operating conditions (see Note 2)

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage (see Note 3)		4.75	5	5.25	V
V_{BB}	Supply voltage		−4.75	−5	−5.25	V
	DGTL GND voltage with respect to ANLG GND			0		V
V_{IH}	High-level input voltage, all inputs except CLKSEL		2.2			V
V_{IL}	Low-level input voltage, all inputs except CLKSEL				0.8	V
	Clock select input voltage	For 2.048 MHz	V_{BB}		$V_{BB}+0.5$	V
		For 1.544 MHz	0		0.5	
		For 1.536 MHz	$V_{CC}-0.5$		V_{CC}	
R_L	Load resistance	At GSX	10			kΩ
		At PWRO+ and/or PWRO−	300			Ω
C_L	Load capacitance	At GSX			50	pF
		AT PWRO+ and/or PWRO−			100	
T_A	Operating free-air temperature	TCM129C__	−40		85	°C
		TCM29C__	0		70	

NOTES: 2. To avoid any possible damage and reliability problems to these CMOS devices when applying power, the following sequence should be followed:
(1) Connect ground
(2) Connect the most negative voltage
(3) Connect the most positive voltage
(4) Connect the input signals
When powering down the device, follow the above steps in reverse order. If the above procedure cannot be followed, connect a diode between V_{BB} and digital ground, cathode to DGND, anode to V_{BB}.
3. Voltages at analog inputs and outputs, V_{CC}, and V_{BB} terminals are with respect to the ANLG GND terminal. All other voltages are referenced to the DGTL GND terminal unless otherwise noted.

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature

supply current, f_{DCLK} = 2.048 MHz, outputs not loaded

PARAMETER‡		TEST CONDITIONS	TCM129C__ __		TCM29C__ __		UNIT
			TYP†	MAX	TYP†	MAX	
I_{CC} Supply current from V_{CC}	Operating		8	13	7	9	mA
	Standby	FSX or FSR at V_{IL} after 300 ms	0.7	1.5	0.5	1	
	Power-down	$\overline{PDN}$ V_{IL} after 10 µs	0.4	1	0.3	0.8	
I_{BB} Supply current from V_{BB}	Operating		−8	−13	−7	−9	mA
	Standby	FSX or FSR at V_{IL} after 300 ms	−0.7	−1.5	−0.5	−1	
	Power-down	$\overline{PDN}$ V_{IL} after 10 µs	−0.4	−1	−0.3	−0.8	
Power dissipation	Operating		80	130	70	90	mW
	Standby	FSX or FSR at V_{IL} after 300 ms	7	15	5	10	
	Power down	$\overline{PDN}$ V_{IL} after 10 µs	4	10	3	8	

digital interface

PARAMETER		TEST CONDITONS	TCM129C__ __			TCM29C__ __			UNIT
			MIN	TYP†	MAX	MIN	TYP†	MAX	
V_{OH} High-level output voltage	PCM out	I_{OH} = −9.6 mA	2.4			2.4			V
	SIGR	I_{OH} = −1.2 mA	2.4			2.4			
V_{OL} Low-level output voltage at PCM out, TSX, SIGR		I_{OL} = 3.2 mA			0.5			0.4	V
I_{IH} High-level input current, any digital input		V_I = 2.2 V to V_{CC}			12			10	µA
I_{IL} Low-level input current, any digial input		V_I = 0 to 0.8 V			12			10	µA
C_i Input capacitance				5	10		5	10	pF
C_o Output capacitance				5			5		pF

transmit amplifier input

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
Input current at ANLG IN+, ANLG IN−	V_I = −2.17 V to 2.17 V			±100	nA
Input offset voltage at ANLG IN+, ANLG IN−	V_I = −2.17 V to 2.17 V			±25	mV
Common-mode rejection at ANLG IN+, ANLG IN−	V_I = −2.17 V to 2.17 V	55			dB
Open-loop voltage amplification at GSX		5000			
Open-loop unity-gain bandwidth at GSX			1		MHz
Input resistance at ANLG IN+, ANLG IN−		10			MΩ

receive filter output

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
Output offset voltage PWRO+, PWRO− (single-ended)	Relative to ANLG GND		80		mV
Output resistance at PWRO+, PWRO−			1		Ω

†All typical values are at V_{BB} = −5 V, V_{CC} = 5 V, and T_A = 25°C.

gain and dynamic range, V_{CC} = 5 V, V_{BB} = −5 V, T_A = 25°C (unless otherwise noted) (see Notes 4, 5, and 6)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
Encoder milliwatt response (transmit gain tolerance)		Signal input = 1.064 V rms for μ-law	Standard version		±0.04	±0.2	dBm0
		Signal input = 1.068 V rms for A-law	TCM29C13N-3		±0.2	±0.5	
Encoder milliwatt response (nominal supplies and temperature)		T_A = 0°C to 70°C, Supplies = ±5%				±0.08	dB
Digital milliwatt response (receive tolerance gain) relative to zero-transmission level point		Signal input per CCITT G.711, Output signal = 1 kHz	Standard version		±0.04	±0.2	dBm0
			TCM29C13N-3		±0.2	±0.5	
Digital milliwatt response variation with temperature and supplies		T_A = 0°C to 70°C, Supplies = ±5%				±0.08	dB
Zero-transmission-level point, transmit channel (0 dBm0)	μ-law	R_L = 600 Ω			2.76		dBm
	A-law				2.79		
	μ-law	R_L = 900 Ω			1.00		
	A-law				1.03		
Zero-transmission-level point, receive channel (0 dBm0)	μ-law	R_L = 600 Ω			5.76		dBm
	A-law				5.79		
	μ-law	R_L = 900 Ω			4.00		
	A-law				4.03		

NOTES: 4. Unless otherwise noted, the analog input is a 0-dBm0, 1020-Hz sine wave, where 0 dBm0 is defined as the zero-reference point of the channel under test. This corresponds to an analog signal input of 1.064 V rms, or an output of 1.503 V rms.

5. The input amplifier is set for unity gain, noninverting. The digital input is a PCM bit stream generated by passing a 0-dBm0, 1020-Hz sine wave through an ideal encoder.

6. Receive output is measured single-ended in the maximum-gain configuration. To set the output amplifier for maximum gain, GSR is connected to PWRO− and the output is taken at PWRO+. All output levels are (sin x)/x corrected.

gain tracking over recommended ranges of supply voltage and operating free-air temperature, reference level = −10 dBm0

PARAMETER	TEST CONDITIONS	MIN	MAX	UNIT
Transmit gain tracking error, sinusoidal input	3 to −40 dBm0		±0.25	dB
	−40 to −50 dBm0		±0.5	
	−50 to −55 dBm0		±1.2	
Receive gain tracking error, sinusoidal input	3 to −40 dBm0		±0.25	dB
	−40 to −50 dBm0		±0.5	
	−50 to −55 dBm0		±1.2	

noise over recommended ranges of supply voltage and operating free-air temperature

PARAMETER	TEST CONDITIONS	MIN	MAX	UNIT
Transmit noise, C-message weighted	ANLG IN+ = ANLG GND, ANLG IN− = GSX		15	dBrnC0
Transmit noise, C-message weighted with eighth-bit signaling (TCM129C14 and TCM29C14 only)	ANLG IN+ = ANLG GND, ANLG IN− = GSX, 6th frame signaling		18	dBrnC0
Transmit noise, psophometrically weighted	ANLG IN+ = ANLG GND, ANLG IN− = GSX		−75	dBm0p
Receive noise, C-message weighted quiet code	PCM IN = 11111111 (μ-law) PCM IN = 10101010 (A-law) measured at PWRO+		11	dBrnC0
Receive noise, C-message weighted sign bit toggled	Input to PCM IN is zero code with sign bit toggled at 1-kHz rate		12	dBmC0
Receive noise, psophometrically weighted	PCM = lowest positive decode level		−79	dBm0p

power supply rejection and crosstalk attenuation over recommended ranges of supply voltage and operating free-air temperature

PARAMETER		TEST CONDITIONS	MIN	TYP[†]	MAX	UNIT
V_{CC} supply voltage rejection ratio, transmit channel	f = 0 to 30 kHz	Idle channel, supply signal = 200 mV p-p, f measured at PCM OUT		−30		dB
	f = 30 to 50 kHz			−45		
V_{BB} supply voltage rejection ratio, transmit channel	f = 0 to 30 kHz	Idle channel, supply signal = 200 mV p-p, f measured at PCM OUT		−30		dB
	f = 30 to 50 kHz			−55		
V_{CC} supply voltage rejection ratio, receive channel (single-ended)	f = 0 to 30 kHz	Idle channel, supply signal = 200 mV p-p, narrow-band, f measured at PWRO+		−20		dB
	f = 30 to 50 kHz			−45		
V_{BB} supply voltage rejection ratio,	f = 0 to 30 kHz	Idle channel, supply signal = 200 mV p-p, narrow-band, f measured at PWRO+		−20		dB
receive channel (single-ended)	f = 30 to 50 kHz			−45		
Crosstalk attenuation, transmit-to-receive (single-ended)		ANLG IN+ = 0 dBm0, f = 1.02 kHz, unity gain, PCM IN = lowest decode level, measured at PWRO+	71			dB
Crosstalk attenuation, receive-to-transmit (single-ended)		PCM IN = 0 dBm0, f = 1.02 kHz, Measured at PCM OUT	71			dB

[†] All typical values are at $V_{BB} = -5$ V, $V_{CC} = 5$ V, and $T_A = 25\,°C$.

distortion over recommended ranges of supply voltage and operating free-air temperature

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
Transmit signal to distortion ratio, sinusoidal input (CCITT G.712 — Method 2)	ANLG IN+ = 0 to −30 dBm0	36			dF
	ANLG IN+ = −30 to −40 dBm0	30			
	ANLG IN+ = −40 to −45 dBm0	25			
Receive signal to distortion ratio, sinusoidal input (CCITT G.712 — Method 2)	ANLG IN+ = 0 to −30 dBm0	36			dB
	ANLG IN+ = −30 to −40 dBm0	30			
	ANLG IN+ = −40 to −45 dBm0	25			
Transmit single-frequency distortion products	AT&T Advisory #64 (3.8), Input signal = 0 dBm0			−46	dBm0
Receive single-frequency distortion products	AT&T Advisory #64 (3.8), Input signal = 0 dBm0			−46	dBm0
Intermodulation distortion, end-to-end	CCITT G.712 (7.1)			−35	dBm0
	CCITT G.712 (7.2)			−49	
Spurious out-of-band signals, end-to-end	CCITT G.712 (6.1)			−25	dBm0
	CCITT G.712 (9)			−40	
Transmit absolute delay time to PCM OUT	Fixed data rate, f_{CLKX} = 2.048 MHz, Input to ANLG IN+ 1.02 kHz at 0 dBm0		245		µs
Transmit differential envelope delay time relative to transmit absolute delay time	f = 500 Hz to 600 Hz		170		µs
	f = 600 Hz to 1000 Hz		95		
	f = 1000 Hz to 2600 Hz		45		
	f = 2600 Hz to 2800 Hz		105		
Receive absolute delay time to PWRO+	Fixed data rate, f_{CLKR} = 2.048 MHz, Digital input is DMW codes		190		µs
Receive differential envelope delay time relative to transmit absolute delay time	f = 500 Hz to 600 Hz		45		µs
	f = 600 Hz to 1000 Hz		35		
	f = 1000 Hz to 2600 Hz		85		
	f = 2600 Hz to 2800 Hz		110		

† All typical values are at V_{BB} = −5 V, V_{CC} = 5 V, and T_A = 25°C.

transmit filter transfer over recommended ranges of supply voltage and operating free-air temperature (see Figure 1)

PARAMETER	TEST CONDITIONS		MIN	MAX	UNIT
Gain relative to gain at 1.02 kHz	Input amplifier set for unity gain, Noninverting maximum gain output, Input signal at ANLG IN+ is 0 dBm0	16.67 Hz		−30	dB
		50 Hz		−25	
		60 Hz		−23	
		200 Hz	−1.8	−0.125	
		300 Hz to 3 kHz	−0.15	0.15	
		3.3 kHz	−0.35	0.03	
		3.4 kHz	−1	−0.1	
		4 kHz		−14	
		4.6 kHz and above		−32	
		3.4 kHz (TCM29C13N-3 only)	−1.4	−0.1	

receive filter transfer over recommended ranges of supply voltage and operating free-air temperature (see Figure 2)

PARAMETER	TEST CONDITIONS		MIN	MAX	UNIT
Gain relative to gain at 1.02 kHz	Input signal at PCM IN is 0 dBm0	Below 200 Hz		0.15	dB
		200 Hz	−0.5	0.15	
		300 Hz to 3 kHz	−0.15	0.15	
		3.3 kHz	−0.35	0.03	
		3.4 kHz	−1	−0.1	
		4 kHz		−14	
		4.6 kHz and above		−30	
		3.4 kHz (TCM29C13N-3 only)	−1.4	−0.1	

clock timing requirements over recommended ranges of supply voltage and operating free-air temperature (see timing diagrams)

PARAMETER		MIN	TYP†	MAX	UNIT
$t_{c(CLK)}$	Clock period for CLKX, CLKR (2.048-MHz systems)	488			ns
t_r, t_f	Rise and fall times for CLKX and CLKR	5		30	ns
$t_{w(CLK)}$	Pulse duration for CLKX and CLKR (see Note 7)	220			ns
$t_{w(DCLK)}$	Pulse duration for DCLK (f_{DCLK} = 64 Hz to 2.048 MHz) (see Note 7)	220			ns
	Clock duty cycle [$t_{w(CLK)}/t_{c(CLK)}$] for CLKX and CLKR	45	50	55	%

† All typical values are at V_{BB} = −5 V, V_{CC} = 5 V, and T_A = 25°C.

transmit timing requirements over recommended ranges of supply voltage and operating free-air temperature, fixed-data-rate mode (see timing diagrams)

PARAMETER		MIN	MAX	UNIT
$t_{d(FSX)}$	Frame sync delay time	100	$t_{c(CLK)} - 100$	ns
$t_{su(SIGX)}$	Setup time before Bit 7 falling edge (TCM129C14 and TCM29C14 only)	0		ns
$t_{h(SIGX)}$	Hold time after Bit 8 falling edge (TCM129C14 and TCM29C14 only)	0		ns

propagation delay times over recommended ranges of operating conditions, fixed-data-rate mode (see timing diagrams)

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{pd1}	From rising edge of transmit clock to Bit 1 data valid at PCM OUT (data enable time on time slot entry) (see Note 8)	C_L = 0 to 100 pF	0	145	ns
t_{pd2}	From rising edge of transmit clock Bit n to Bit n + 1 data valid at PCM OUT (data valid time)	C_L = 0 to 100 pF	0	145	ns
t_{pd3}	From falling edge of transmit clock Bit 8 to Bit 8 Hi-Z at PCM OUT (data float time on time slot exit) (see Note 8)	C_L = 0	60	215	ns
t_{pd4}	From rising edge of transmit clock Bit 1 to TSX active (low) (time slot enable time)	C_L = 0 to 100 pF	0	145	ns
t_{pd5}	From falling edge of transmit clock Bit 8 to TSX inactive (high) (timeslot disable time) (see Note 8)	C_L = 0	60	190	ns
t_{pd6}	From rising edge of channel time slot to SIGR update (TCM129C14 and TCM29C14 only)		0	2	μs

NOTES: 7. FSX CLK must be phase locked with the CLKX, FSR CLK must be phase locked with CLKR.
8. Timing parameters t_{pd1}, t_{pd3}, and t_{pd5} are referenced to the high-impedance state.

receive timing requirements over recommended ranges of supply voltage and operating free-air temperature, fixed-data-rate mode (see timing diagrams)

PARAMETER		MIN	MAX	UNIT
$t_{d(FSR)}$	Frame sync delay time	100	$t_{c(CLK)} - 100$	ns
$t_{su(PCM\ IN)}$	Setup time before Bit 7 falling edge (TCM129C14 and TCM29C14 only)	10		ns
$t_{h(PCM\ IN)}$	Hold time after Bit 8 falling edge (TCM129C14 and TCM29C14 only)	60		ns

transmit timing requirements over recommended ranges of supply voltage and operating free-air temperature, variable-data-rate mode (see timing diagrams)

PARAMETER		MIN	MAX	UNIT
$t_{d(TSDX)}$	Timeslot delay time from DCLKX (see Note 9)	140	$t_{d(DCLKX)} - 140$	ns
$t_{d(FSX)}$	Frame sync delay time	100	$t_{c(CLK)} - 100$	ns
$t_{c(DCLKX)}$	Clock period for DCLKX	488	15620	kHz

propagation delay times over recommended ranges of operating conditions, variable-data-rate mode (see Note 10 and timing diagrams)

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{pd7}	Data delay time from DCLKX	C_L = 0 to 100 pF	0	100	ns
t_{pd8}	Data delay from timeslot enable to PCM OUT	C_L = 0 to 100 pF	0	50	ns
t_{pd9}	Data delay from time slot disable to PCM OUT	C_L = 0 to 100 pF	0	80	ns
t_{pd10}	Data delay time from FSX	$t_{d(TSDX)}$ = 80 ns	0	140	ns

receive timing requirements over recommended ranges of supply voltage and operating free-air temperature, variable-data-rate mode (see timing diagrams)

PARAMETER		MIN	MAX	UNIT
$t_{d(TSDR)}$	Timeslot delay time from DCLKR (see Note 11)	140	$t_{d(DCLKR)} - 140$	ns
$t_{d(FSR)}$	Frame sync delay time	100	$t_{c(CLK)} - 100$	ns
$t_{su(PCM\ IN)}$	Setup time before Bit 7 falling edge	10		ns
$t_{h(PCM\ IN)}$	Hold time after Bit 8 falling edge	60		ns
$t_{c(DCLKR)}$	Data clock frequency	488	15620	ns
$t_{(SER)}$	Timeslot end receive time	0		ns

64-kilobit operation timing requirements over recommended ranges of supply voltage and operating free-air temperature, variable-data-rate mode

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{FSLX}	Transmit frame sync minimum down time	FSX = TTL high for remainder of frame	488		ns
t_{FSLR}	Receive frame sync minimum down time	FSR = TTL high for remainder of frame	1952		ns
t_{DCLK}	Pulse duration, data clock			10	µs

NOTES: 9. t_{FSLX} minimum requirement overrides the $t_{d(TSDX)}$ maximum requirement for 64-kHz operation.
10. Timing parameters t_{pd8} and t_{pd9} are referenced to a high-impedance state.
11. t_{FSLR} minimum requirement overrides the $t_{d(TSDR)}$ maximum requirement for 64-kHz operation.

CLK, CLKR, and CLKX Selection Requirements for DSP Based Applications

1) It should be noted that the CLKX, CLKR, CLK must be selected as follows:

CLKSEL PIN	CLK, CLKR, CLKX (BETWEEN 1.0 MHz to 3.0 MHz)	DEVICE TYPE
−5 V†	= (256) × (Frame Sync Frequency)	TCM129C13/14/16/17
		TCM29C13/14/16/17
0 V	= (193) × (Frame Sync Frequency)	TCM129C13/14
		TCM29C13/14
+5 V	= (192) × (Frame Sync Frequency)	TCM129C13/14
		TCM29C13/14

E.G.: For Frame Sync Frequency = 9.6 kHz

CLKSEL PIN	CLK, CLKR, CLKX (BETWEEN 1.0 MHz to 3.0 MHz)	DEVICE TYPE
−5 V†	=2.4576 MHz	TCM129C13/14/16/17
		TCM29C13/14/16/17
0 V	= 1.8528 MHz	TCM129C13/14
		TCM29C13/14
+5 V	= 1.8432 MHz	TCM129C13/14
		TCM29C13/14

†CLKSEL is internally set to −5 V for TCM129C16/17 and TCM29C16/17.

2) Corner frequency at 8 kHz Frame Sync Frequency = 3kHz

Therefore, the corner frequency = (3/8) × (Frame Sync Frequency). (For nonstandard frame sync.)

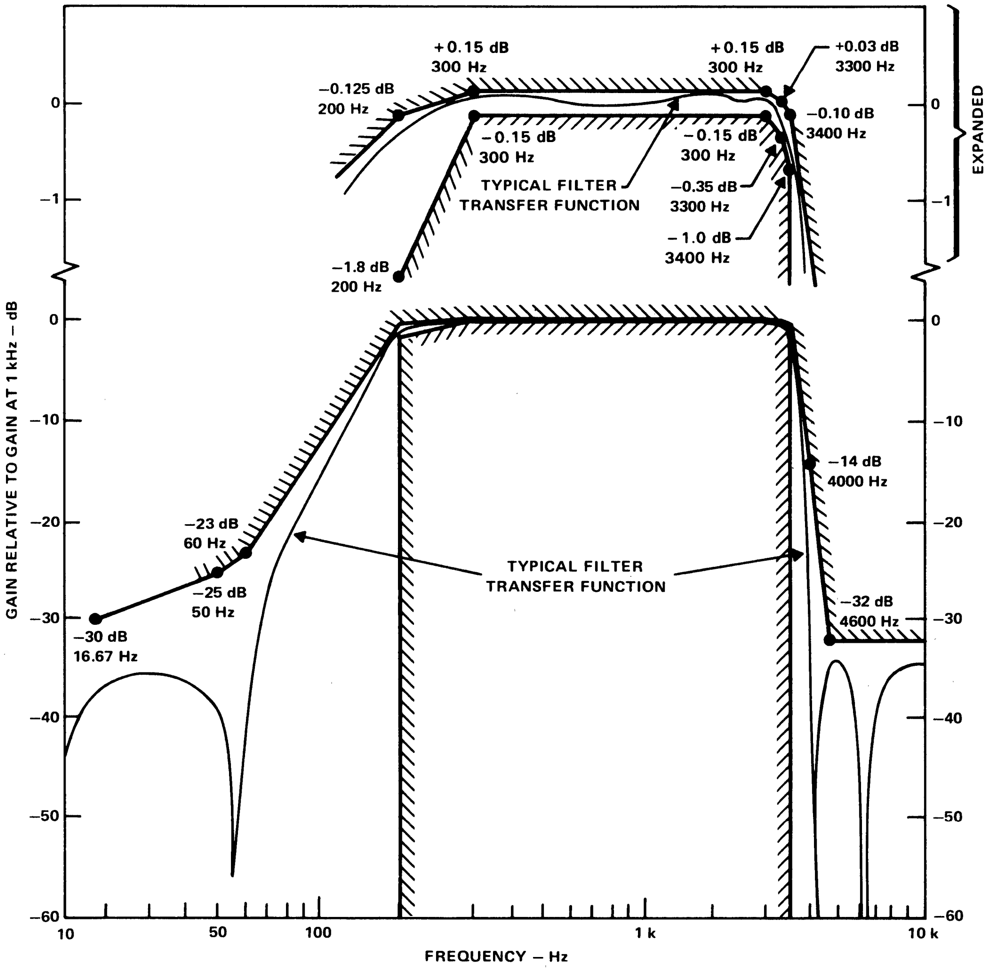
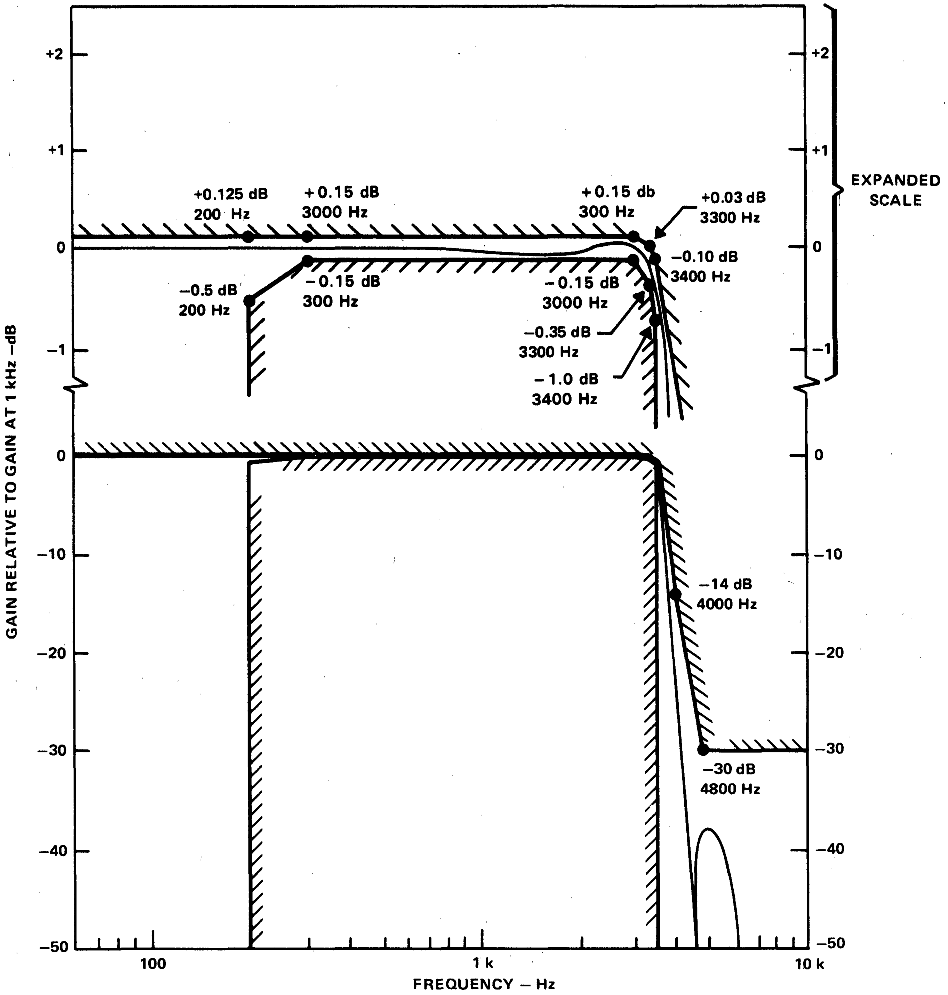
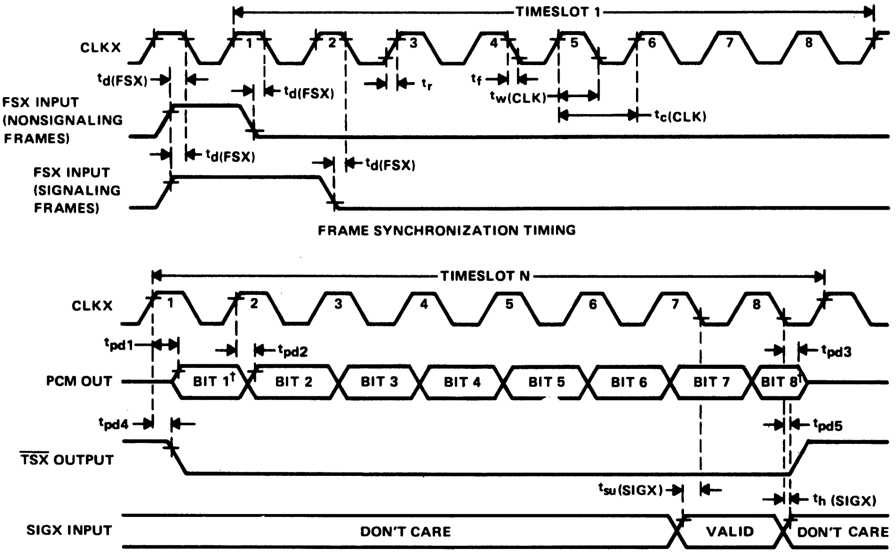


FIGURE 1. TRANSFER CHARACTERISTICS OF THE TRANSMIT FILTER



NOTE: This is a typical transfer function of the receive filter component.

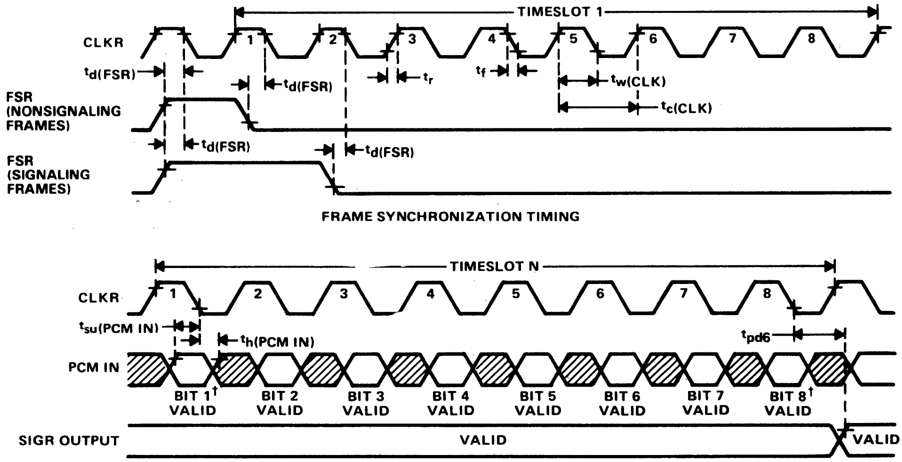
FIGURE 2. TRANSFER CHARACTERISTIC OF THE RECEIVE FILTER



FRAME SYNCHRONIZATION TIMING

OUTPUT TIMING

FIGURE 3. TRANSMIT TIMING (FIXED-DATA-RATE)



FRAME SYNCHRONIZATION TIMING

INPUT TIMING

FIGURE 4. RECEIVE TIMING (FIXED-DATA-RATE)

NOTE: Inputs and driven from 0.45 V to 2.4 V. Time intervals are referenced to 2 V if the high level is indicated and 0.8 V if the low level is indicated.

[†] Bit 1 = MSB = SIGN BIT and is clocked in first on the PCM-IN pin or clocked out first on the PCM-OUT pin. BIT 8 = LSB = LEAST SIGNIFICANT BIT and is clocked in last on the PCM-IN pin or is clocked out last on the PCM-OUT pin.

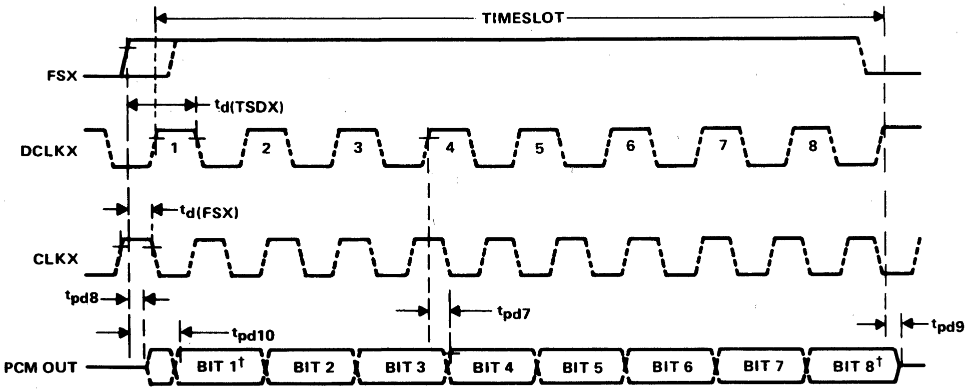
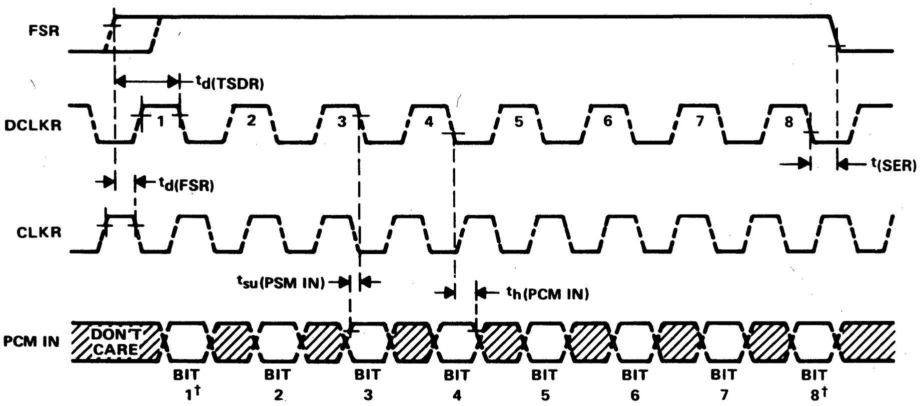


FIGURE 5. TRANSMIT TIMING (VARIABLE-DATA-RATE)



NOTE: All timing parameters referenced to V_{IH} and V_{IL} except t_{pd8} and t_{pd9}, which reference a high-impedance state.

FIGURE 6. RECEIVE TIMING (VARIABLE-DATA-RATE)

NOTE: All timing parameters, referenced to V_{IH} and V_{IL} except t_{pd8} and t_{pd9}, which reference a high-impedance state.

†Bit 1 = MSB = SIGN BIT and is clocked in first on the PCM-IN pin or clocked out first on the PCM-OUT pin. BIT 8 = LSB = LEAST SIGNIFICANT BIT and is clocked in last on the PCM-IN pin or is clocked out last on the PCM-OUT pin.

GENERAL OPERATION

system reliability features

The TCM129C13, TCM129C14, TCM129C16, TCM129C17, TCM29C13, TCM29C14, TCM29C16, and TCM29C17 are powered up in four steps:
V_{CC} and V_{BB} supply voltages are applied.
All clocks are connected.
TTL high is applied to $\overline{\text{PDN}}$.
FSX and/or FSR synchronization pulses are applied.

On the transmit channel, digital outputs PCM OUT and $\overline{\text{TSX}}$ are held in high-impedance state for approximately four frames (500 μs) after power up or application of V_{BB} or V_{CC}. After this delay, PCM OUT, $\overline{\text{TSX}}$, and signaling are functional and will occur in the proper timeslot. The analog circuits on the transmit side require approximately 60 ms to reach their equilibrium value due to the autozero circuit settling time. Thus valid digital information, such as for on/off hook detection, is available almost immediately, while analog information is available after some delay.

On the receive channel, the digital output SIGR is also held low for a maximum of four frames after power up or application of V_{BB} or V_{CC}. SIGR will remain low until it is updated by a signalling frame.

To further enhance system reliability, PCM OUT and $\overline{\text{TSX}}$ will be placed in a high-impedance state approximately 20 μs after an interruption of CLKX. SIGR will be held low approximately 20 μs after an interruption of CLKR. These interruptions could possible occur with some kind of fault condition.

power-down and standby operations

To minimize power consumption, a power-down mode and three standby modes are provided.

For power down, an external low signal is applied to the $\overline{\text{PDN}}$ pin. It is not sufficient to remove the high voltage to $\overline{\text{PDN}}$. In the absence of a signal, the $\overline{\text{PDN}}$ pin floats to high and the device remains active. In the power-down mode, the average power consumption is reduced to an average of 5 mW.

The standby modes give the user the option of putting the entire device on standby, putting only the transmit channel on standby, or putting only the receive channel on standby. To place the entire device on standby, both FSX and FSR are held at low. For transmit-only operation, FSX is high and FSR is held low. For receive-only operation, FSR is high and FSX is held low. See Table 1 for power down and standby procedures.

TABLE 1. POWER DOWN AND STANDBY PROCEDURES

DEVICE STATUS	PROCEDURE	TYPICAL POWER CONSUMPTION	DIGITAL OUTPUT STATUS
Power down	$\overline{\text{PDN}}$ low	3 mW	$\overline{\text{TSX}}$ and PCM OUT are in a high-impedance state; SIGR goes to low within 10 μs.
Entire device on standby	FSX and FSR are low	3 mW	$\overline{\text{TSX}}$ and PCM OUT are in a high-impedance state; SIGR goes to low within 300 ms.
Only transmit on standby	FSX is low FSR is high	40 mW	$\overline{\text{TSX}}$ and PCM OUT are placed in a high-impedance state within 300 ms.
Only receive on standby	FSR is low FSX is high	30 mW	SIGR is placed in a high-impedance state within 300 ms.

fixed-data-rate timing (see Figure 7)

Fixed-data-rate timing is selected by connecting DCLKR to V_{BB}. It uses master clocks CLKX and CLKR, frame synchronizer clocks FSX and FSR, and output $\overline{TSX}$. FSX and FSR are 8-kHz inputs that set the sampling frequency and distinguish between signaling and nonsignaling frames by their pulse durations. A frame synchronization pulse one master clock period long designates a nonsignaling frame, while a double-length sync pulse enables the signaling function (TCM129C14 and TCM29C14 only). Data is transmitted on the PCU OUT pin on the first eight positive transitions of CLKX following the rising edge of FSX. Data is received on the PCM IN pin on the first eight falling edges of CLKR following FSX. A digital-to-analog (D/A) conversion is performed on the received digital word and the resulting analog sample is held on an internal sample-and-hold capacitor until transferred to the receive filter.

The clock selection pin (CLKSEL) is used to select the frequency of CLKX and CLKR (TCM129C13, TCM129C14, TCM29C13, and TCM29C14 only). The TCM129C13, TCM129C14, TCM29C13, and TCM29C14 fixed-data-rate mode can operate with frequencies of 1.536 MHz, 1.544 MHz, or 2.048 MHz. The TCM129C16, TCM129C17, TCM29C16, and TCM29C17 fixed data rate mode operates at 2.048 MHz only.

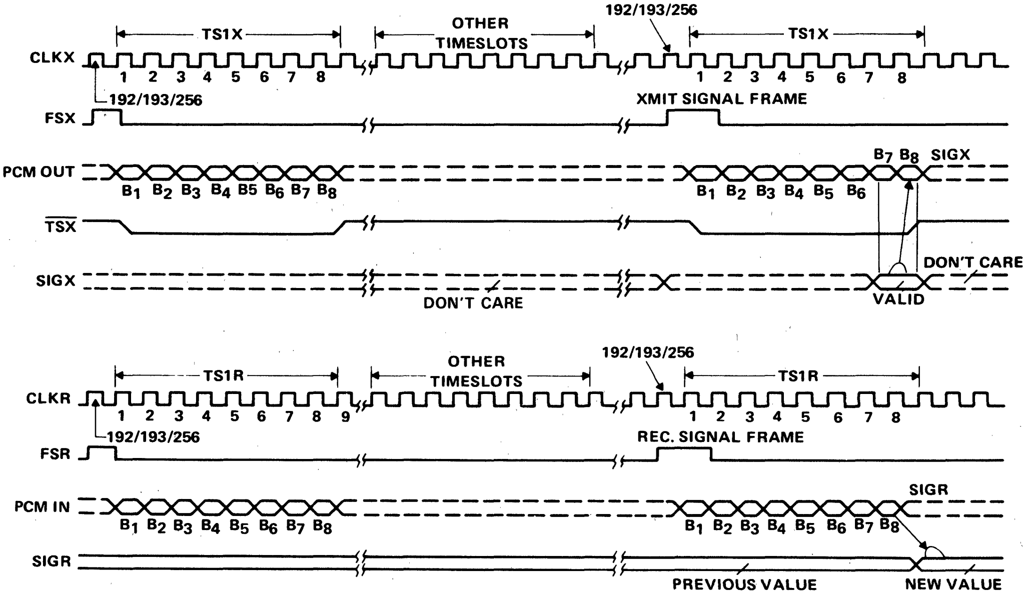


FIGURE 7. SIGNALING TIMING (FIXED-DATA-RATE ONLY)

variable data rate timing

Variable-data-rate timing is selected by connecting DCLKR to the bit clock for the receive PCM highway rather than to V_{BB}. It uses master clocks CLKX and CLKR, bit clocks DCLKX and DCLKR, and frame synchronization clocks FSX and FSR.

Variable-data-rate timing allows for a flexible data frequency. The frequency of the bit clocks can be varied from 64 kHz to 2.048 MHz. The bit clocks can be asynchronous in the TCM129C14 and TCM29C14, but must be synchronous in the TCM129C13, TCM129C16, TCM129C17, TCM29C13, TCM29C16, and TCM29C17. Master clocks in types TCM129C13, TCM129C14, TCM29C13, and TCM29C14 are restricted to frequencies of operation of 1.536 MHz, 1.544 MHz, or 2.048 MHz as in the fixed-data-rate timing mode. The master clock for the TCM129C16, TCM129C17, TCM29C16, and TCM29C17 is restricted to 2.048 MHz.

While FSX/TSXE input is high, PCM data is transmitted from PCM OUT onto the highway on the next eight consecutive positive transitions of DCLKX. Similarly, while the FSR/TSRE input is high, the PCM word is received from the highway by PCM IN on the next eight consecutive negative transitions of DCLKR.

The transmitted PCM word will be repeated in all remaining timeslots in the 125 μs frame as long as DCLKX is pulsed and FSX is held high. This feature, which allows the PCM word to be transmitted to the PCM highway more than once per frame, if desired, is available only with variable-data-rate timing. Signaling is allowed only in the fixed-data-rate mode because the variable-data-rate mode provides no means with which to specify a signaling frame.

signaling

The TCM29C14 (only) provides 8th-bit signaling in the fixed-data-rate timing mode. Transmit and receive signaling frames are independent of each other and are selected by a double-width frame sync pulse on the appropriate channel. During a transmit signaling frame, the signal present on SIGX is substituted for the least significant bit (LSB) of the encoded PCM word. In a receive signaling frame, the codec will decode the seven most significant bits in accordance with CCITT G.733 recommendations, and output the logical state of the LSB on the SIGR pin until it is updated in the next signaling frame. Timing relationships for signaling operations are shown n Figure 9. The signaling path is used to transmit digital signaling information such as ring control, rotary dial pulses, and off-hook and disconnect supervision. The voice path is used to transmit prerecorded messages as well as the call progress tones; dial tone, ring-back tone, busy tone, and re-order tone.

asynchronous operation

The TCM129C14 and TCM29C14 can be operated with asynchronous clocks in either the fixed- or variable-data-rate modes. In order to avoid crosstalk problems associated with special interrupt circuits, the design of the TCM129C13, TCM129C14, TCM29C13, and TCM29C14 includes separate digital-to-analog converters and voltage references on the transmit and receive sides to allow completely independent operation of the two channels.

In either timing mode, the master clock, data clock, and timeslot strobe must be synchronized at the beginning of each frame. Specifically, in the variable-data-rate mode the rising edge of CLKX must occur within $t_{d(FSX)}$ ns before the rise of FSX, while the leading edge of DCLKX must occur within t_{TSDX} ns of the rise of FSX. CLKX and DCLKX are synchronized once per frame but may be of different frequencies. The receive channel operates in a similar manner and is completely independent of the transmit timing (see variable data rate timing diagrams). This approach requires the provision of two separate master clocks but avoids the use of a synchronizer, which can cause intermittent data conversion errors.

analog loopback

A distinctive feature of the TCM129C14 and TCM29C14 is their analog loopback capability. With this feature, the user can test the line circuit remotely by comparing the signals sent into the receive channel (PCM IN) with those generated on the transmit channel (PCM OUT). The test is accomplished by sending a control signal that internally connects the analog input and output ports. WhenANLG LOOP is TTL high, the receive output (PWRO +) is internally connected to ANLG IN +, GSR is internally connected to PWRO –, and ANLG IN – is internally connected to GSX (see Figure 8).

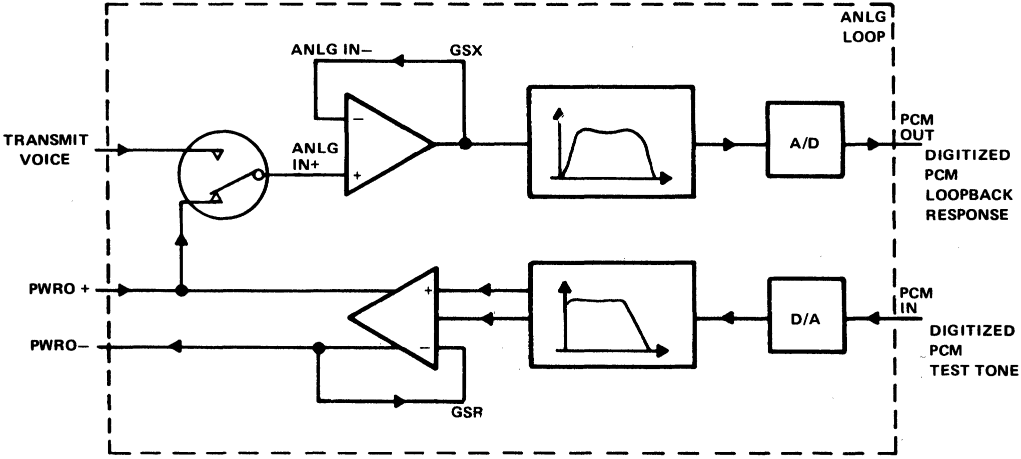


FIGURE 8. TCM129C14 AND TCM29C14 ANALOG LOOPBACK CONFIGURATION

Due to the difference in the transmit and receive transmission levels, a 0 dBmO code into PCM IN will emerge from PCM OUT as a 3-dBmO code, an implicit gain of 3 dB. Because of this, the maximum signal that can be tested by analog loopback is 0 dBmO.

precision voltage references

No external components are required with the devices to provide the voltage references. Voltage references that determine the gain and dynamic range characteristics of the device are generated internally. A difference in subsurface charge density between two suitably implanted MOS devices is used to derive a temperature- and bias-stable reference voltage. These references are calibrated during the manufacturing process.Separate references are supplied to the transmit and receive sections, and each is calibrated independently. Each reference value is then further trimmed in the gain setting operational amplifiers to a final precision value. Manufacturing tolerances can be achieved of typically ±0.04 dB in absolute gain for each half channel, providing the user a significant margin to compensate for error in other board components.

conversion laws

The TCM129C13, TCM129C14, TCM29C13, and TCM29C14 provide pin-selectable μ-law operation as specified by CCITT G.711 recommendation. A-law operation is selected when the ASEL pin is connected to V_{BB}. Signaling is not allowed during A-law operation. The TCM129C16 and TCM29C16 are μ-law only. The TCM129C17 and TCM29C17 are A-law only.

The μ-law operation is effectively selected by not selecting A-law operation. If the ASEL pin is connected to v_{CC} or GND, the device is in μ-law operation. If μ-law operation is selected, SIGX is a TTL-level input that can be used in the fixed data rate timing mode to modify the LSB of the PCM output is signaling frames.

transmit operation

transmit filter

The input section provides gain adjustment in the passband by means of an on-chip uncommitted operational amplifier. the load impedance to ground (ANLG GND) at the amplifier output (GSX) must be greater than 10 kΩ in parallel with less than 50 pF. The input signal on the ANLG IN + pin can be either ac or dc coupled. The input operational amplifier can also be used in the inverting mode or differential amplifier mode.

A low-pass antialiasing section is included on the device. This section provides 35-dB attenuation at the sampling frequency. No external components are required to provide the necessary antialiasing function for the switched capacitor section of the transmit filter.

The passband section provides flatness and stopband attenuation that fulfills the AT&T D3/D4 channel bank transmission specification and CCITT recommendation G.712. The device specifications meet or exceed digital class 5 central office switching systems requirements.

A high-pass section configuration was chosen to reject low-frequency noise from 50- and 60-Hz power lines, 17-Hz European electric railroads, ringing frequencies and their harmonics, and other low-frequency noise. Even with the high rejection at these frequencies, the sharpness of the band edge gives low attenuation at 200 Hz. This feature allows the use of low-cost transformer hybrids without external components.

encoding

The encoder internally samples the output of the transmit filter and holds each sample on an internal sample and hold capacitor. The encoder performs an analog-to-digital conversion on a switched capacitor array. Digital data representing the sample is transmitted on the first eight data clocks bits of the next frame.

The autozero circuit corrects for dc offset on the input signal to the encoder. The autozero circuit uses the sign bit averaging technique. The sign bit from the encoder output is long-term averaged and subtracted from the input to the encoder. All dc offset is removed from the encoder input waveform.

receive operation

decoding

The serial PCM word is received at the PCM IN pin on the first ight data clock bits of the frame. Digital-to-analog conversion is performed and the corresponding analog sample is held on an internal sample-and-hold capacitor. This sample is transferred to the receive filter.

receive filter

The receive section of the filter provides passband flatness and stopband rejection that fulfills both the AT&T D3/D4 specification and CCITT recommendation G.712. The filter contains the required compensation for the (sin x)/x response of such decoders.

receive output power amplifiers

A balanced output amplifier is provided to allow maximum flexibility in output configuration. Either of the two outputs can be used single-ended (i.e., referenced to ANLG GND) to drive single-ended loads. Alternatively, the differential output will directly drive a bridged load. The output stage is capable of driving loads as low as 300 ohms single-ended to a level of 12 dBm or 600 ohms differentially to a level of 15 dBm.

The receive channel transmission level may be adjusted between specified limits by manipulation of the GSR input. GSR is internally connected to an analog gain-setting network. When GSR is connected to PWRO −, the receive level is at maximum. When GSR is connected to PWRO +, the level is minimum. The output transmission level is adjusted between 0 and − 12 dB as GSR is adjusted (with an adjustable resistor) between PWRO + and PWRO −.

Transmission levels are specified relative to the receive channel output under digital milliwatt conditions (i.e., when the digital input at PCM IN is the eight-code sequence specified in CCITT recommendation G.711).

TYPICAL APPLICATION DATA

output gain set design considerations (see Figure 9)

PWRO + and PWRO − are low-impedance complementary outputs. The voltages at the nodes are:

V_{O+} at PWRO +
V_{O-} at PWRO −
$V_{OD} = V_{O+} - V_{O-}$ (total differential response)

R1 and R2 are a gain-setting resistor network with the center tap connected to the GSR input.

A value greater than 10 kΩ and less than 100 kΩ for R1 + R2 is recommended because of the following:

The parallel combination of R1 + R2 and R_L sets the total loading.
The total capacitance at the GSR input and the parallel combination of R1 and R2 define a time constant that has to be minimized to avoid inaccuracies.

V_A represents the maximum available digital milliwatt output response ($V_A = 3.06$ V rms).

$$V_{OD} = A \cdot V_A$$

$$\text{Where } A = \frac{1 + (R1/R2)}{4 + (R1/R2)}$$

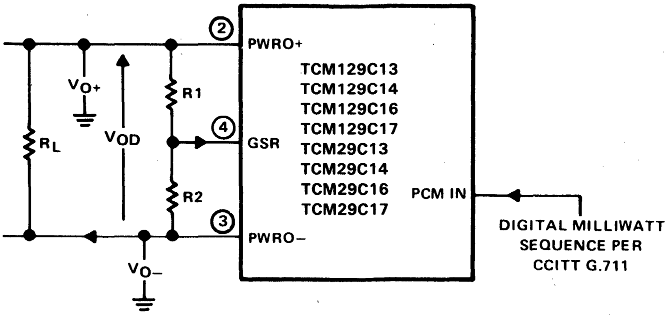


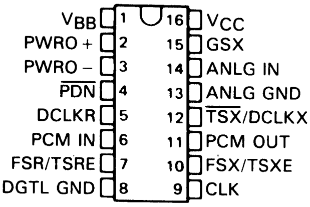
FIGURE 9. GAIN-SETTING CONFIGURATION

TCM129C18, TCM129C19, TCM29C18, TCM29C19
ANALOG INTERFACE FOR DSP

D3036, AUGUST 1987—REVISED JUNE 1988

- **Reliable Silicon-Gate CMOS Technology**
- **Low Power Consumption**
  Operating Mode . . . 80 mW
  Power-Down Mode . . . 5 mW
  μ-Law Coding
- **Excellent Power Supply Rejection Ratio Over Frequency Range of 0 to 50 kHz**
- **No External Components Needed for Sample, Hold, and Auto-Zero Functions**
- **Precision Internal Voltage References**
- **Single Chip Contains A/D, D/A, and Associated Filters**

N DUAL-IN-LINE PACKAGE
(TOP VIEW)



FEATURE TABLE

16 Pins
μ-Law Coding
Variable Mode:
64 kHz to 2.048 MHz
Fixed Mode:
2.048 MHz (TCM129C18, TCM29C18),
1.536 MHz (TCM129C19, TCM29C19)
8-Bit Resolution
12-Bit Dynamic Range

description

The TCM129C18, TCM129C19, TCM29C18, and TCM29C19 are low-cost single-chip pulse-code-modulated encoders and decoders (PCM codecs) and PCM line filters. These devices incorporate both the A/D and D/A functions, an anti-aliasing filter (A/D), and a smoothing filter (D/A). These devices are ideal for use with the TMS320 family members, particularly those featuring a serial port such as the TMS32020, TMS32011, and TMS320C25.

Primary applications of these devices include:

Digital Encryption Systems
Digital Voice-Band Data Storage Systems
Digital Signal Processing

These devices are designed to perform encoding of analog input signals (A/D conversion) and decoding of digital PCM signals (D/A conversion). They are useful for implementation in the analog interface of a digital-signal processing system. Both devices also provide band-pass filtering of the analog signals prior to encoding and smoothing after decoding.

The analog input is encoded into an 8-bit digital representation by use of the μ-law encoding scheme (CCITT G.711) which equates to 12 bits of resolution for low amplitude signals. Similarly, the decoding section converts 8-bit PCM data into an analog signal with 12 bits of dynamic range. The filter characteristics (bandpass) for the encoder and decoder are determined by a single clock input (CLK). The filter roll-off (-3 dB) is derived by:

$f_{CO} = k \cdot f_{CLK}/256$ for the TCM129C18 and TCM29C18 or $f_{CO} = k \cdot f_{CLK}/192$ for the TCM129C19 and TCM29C19

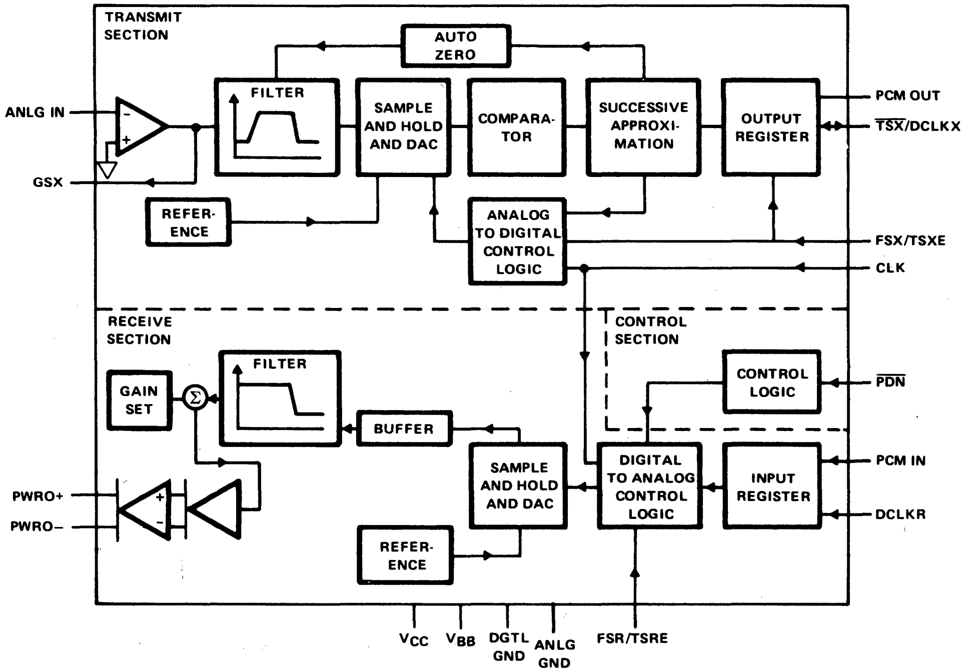
where k has a value of 0.44 for the high-frequency roll-off point, and a value of 0.019 for the low-frequency roll-off point.

description (continued)

The sampling rate of the ADC is determined by the Frame Sync Clock, FSX; the sampling rate of the DAC is determined by the Frame Sync Clock, FSR. Once a conversion is initiated by FSX or FSR, data is clocked in or out on the next consecutive eight clock pulses in the fixed data rate mode. Likewise, data may also be transferred on the next eight consecutive clock pulses of the data clocks, DCLKX and DCLKR, in the variable data rate mode. In the variable data rate mode, DCLKX and DCLKR are independent, but must be in the range from $f_{CLK}/32$ to f_{CLK}.

The TCM129C18 and TCM129C19 are characterized for operation over the temperature range of −40°C to 85°C. The TCM29C18 and TCM29C19 are characterized for operation over the temperature range of 0°C to 70°C.

functional block diagram



NAME	PIN	DESCRIPTION
ANLG IN	14	Inverting analog input to uncommitted transmit operational amplifier
ANLG GND	13	Analog ground return for all voice circuits. Not internally connected to digital ground.
CLK	9	Master clock and data clock for the fixed data rate mode. Master (filter) clock only for variable data-rate mode. This clock is used for both the transmit and receive sections.
DCLKR	5	When this pin is connected to V_{BB}, the device operates in the fixed-data-rate mode. When DCLKR is not connected to V_{BB}, the device operates in the variable-data-rate mode and DCLKR becomes the receive data clock, which operates at frequencies from 64 kHz to 2.048 MHz.
DGTL GND	8	Digital ground for all internal logic circuits. Not internally connected to analog ground.
FSR/TSRE	7	Frame sync clock input/time-slot enable for the receive channel. In the variable-data-rate-mode, this signal must remain high for the duration of the time-slot. The receive channel enters the standby state when FSR is TTL low for 30 ms.
FSX/TSXE	10	Frame synchronization clock input/time-slot enable for transmit channel. Operates independently of, but in an analogous manner to FSR/TSRE. The transmit channel enters the standby state when FSX is low for 300 ms.
GSX	15	Output terminal of internal uncommitted operational amplifier. Internally, this is the voice signal input to the transmit filter.
PCM IN	6	Receive PCM input. PCM data is clocked in on this pin on eight consecutive negative transitions of the receive data clock, which is CLKR in fixed-data-rate timing and DCLKR in variable-data-rate timing.
PCM OUT	11	Transmit PCM output. PCM data is clocked out of this output on eight consecutive positive transitions of the transmit data clock, which is CLKX in fixed-data-rate timing and DCLKX in variable-data-rate timing.
$\overline{\text{PDN}}$	4	Power-Down Select. On the TCM129C18 and the TCM29C18, the device is inactive with a TTL low-level input and active with a TTL high-level input to the pin. On the TCM129C19 and the TCM29C19, this pin must be connected to a TTL high level.
PWRO+	2	Noninverting output of power amplifier can drive transformer hybrids or high-impedance loads directly in either a differential or a single-ended configuration.
PWRO−	3	Inverting output of power amplifier, functionally identical to PWRO+
$\overline{\text{TSX}}$/DCLKX	12	Transmit channel time slot strobe (output) or data clock (input). In the fixed-data-rate mode, this is an open-drain output to be used as an enable signal for a three-state-buffer. In the variable-data-rate mode, DCLKX becomes the transmit data clock, which operates at TTL levels from 64 kHz to 2.048 MHz.
V_{BB}	1	Negative supply voltage, −5 V ±5%.
V_{CC}	16	Positive supply voltage, 5 V ±5%.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	−0.3 to 15 V
Output voltage, V_O	−0.3 to 15 V
Input voltage, digital inputs, V_I	−0.3 to 15 V
Digital ground voltage	−0.3 to 15 V
Operating free-air temperature range	−10°C to 80°C
Storage temperature range	−65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C

NOTE 1: Voltage values for maximum ratings are with respect to V_{BB}.

recommended operating conditions (see Note 2)

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage (see Note 3)		4.75	5	5.25	V
V_{BB}	Supply voltage		−4.75	−5	−5.25	V
	DGTL GND voltage with respect to ANLG GND			0		V
V_{IH}	High-level input voltage, all inputs except ANLG IN		2.2			V
V_{IL}	Low-level input voltage, all inputs except ANLG IN				0.8	V
V_{IPP}	Peak-to-peak analog input voltage				4.2	V
R_L	Load resistance	GSX	10			kΩ
		PWRO+ and/or PWRO−	300			Ω
C_L	Load capacitance	GSX			50	pF
		PWRO+ and/or PWRO−			100	
T_A	Operating free-air temperature	TCM129C18 or TCM129C19	−40		85	°C
		TCM29C18 or TCM29C19	0		70	

NOTES: 2. To avoid any possible damage and reliability problems to these CMOS devices when applying power, the following sequence should be followed:
(1) Connect ground
(2) Connect the most negative voltage
(3) Connect the most positive voltage
(4) Connect the input signals.
When powering down the device, follow the above steps in reverse order. If the above procedure cannot be followed, connect a diode between V_{BB} and DGTL GND, cathode to DGTL GND, anode to V_{BB}.

3. Voltages at analog inputs and outputs, V_{CC} and V_{BB} terminals are with respect to the ANLG GND terminal. All other voltages are referenced to the DGTL GND terminal unless otherwise noted.
4. Analog input signals that exceed 4.2 V peak-to-peak may contribute to clipping and preclude correct A/D conversion. The digital code representing values higher than 4.200 V is 10000000. For values more negative than 4.200 V, the code is 0000000.

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature

supply current, fdclk = 2.048 MHz, outputs not loaded

PARAMETER			TEST CONDITIONS	TCM129CXX		TCM29CXX		UNIT
				MIN	MAX	MIN	MAX	
I_{CC}	Supply current from V_{CC}	operating			14		10	mA
		standby	FSX or FSR at V_{IL} after 300 ms		1.5		1.2	
		power down	$\overline{PDN}$ at V_{IL} after 10 μs		1.2		1	
I_{BB}	Supply current from V_{BB}	operating			−14		−10	mA
		standby	FSX or FSR at V_{IL} after 300 ms		−1.5		−1.2	
		power down	$\overline{PDN}$ at V_{IL} after 10 μs		−1.2		−1	

digital interface

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage, PCM OUT	I_{OH} = −9.6 mA	2.4			V
		I_{OH} = −0.1 mA	3.5			
V_{OL}	Low-level output voltage, $\overline{TSX}$	I_{OL} = 3.2 mA			0.5	V
I_{IH}	High-level input current, any digital input	V_I = 2.2 V to V_{CC}			12	μA
I_{IL}	Low-level input current, any digital input	V_I = 0 to 0.8 V			12	μA
C_i	Input capacitance			5	10	pF
C_o	Output capacitance			5	10	pF

†All typical values are at V_{BB} = −5 V, V_{CC} = 5 V, and T_A = 25°C.

transmit side (A/D) characteristics

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
Input offset current at ANLG IN	V_I = −2.17 V to 2.17 V		1		pA
Input offset voltage at ANLG IN	V_I = −2.17 V to 2.17 V			±25	mV
Input bias current	V_I = −2.17 V to 2.17 V			±100	nA
Open-loop voltage amplification at GSX		5000			
Unity-gain bandwidth at GSX			1		MHz
Input resistance at ANLG IN		10			Ω
Gain tracking error with sinusoidal input (see Notes 5, 6, and 7)	3 dBm0 to −40 dBm0, REF level = −10 dBm0			±0.5	dB
	−40 dBm0 to −50 dBm0, REF level = −10 dBm0			±2.5	
Transmit gain tolerance	V_i = 1.06 V, f = 1.02 kHz	0.95		1.19	Vrms
Noise	Ref max output level: 200 Hz to 3 kHz			−70	dB
Supply voltage rejection ratio, V_{CC} or V_{BB}	f = 0 to 30 kHz, (measured at PCM OUT) idle channel, Supply signal = 200 mV P-P	−20			dB
Crosstalk attenuation, transmit-to-receive (single-ended)	ANLG IN = 0 dBm, f = 1 kHz unity gain, PCM IN = lowest decode level, measured at PWRO+	62			dB
Signal-to-distortion ratio, with sinusoidal input (see Note 8)	ANLG IN = 0 to −30 dBm0	33			dB
	ANLG IN = −30 to −40 dBm0	27			
	ANLG IN = −40 to −45 dBm0	22			
Absolute delay time to PCM OUT	Fixed data rate, FCLKX = 2.048 MHz, input to ANLG IN = 1 kHz at 0 dB		245		µs

receive side (D/A) characteristics (see Note 9)

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
Output offset voltage PWRO+ and PWRO− (single-ended)	Relative to ANLG GND			±200	mV
Output resistance at PWRO+ and PWRO−			1	2	Ω
Gain tracking error with sinusoidal input (see Notes 5, 6, and 7)	3 dBm0 to −40 dBm0, REF level = −10 dBm0			±0.5	dB
	−40 dBm0 to −50 dBm0, REF level = −10 dBm0			±2.5	
Receive gain tolerance	V_i = 1.06 V, f = 1.02 kHz	1.34		1.69	Vrms
Noise	Ref max output level: 200 Hz to 3 kHz			−70	dB
Supply voltage rejection ratio, V_{CC} or V_{BB} (single-ended)	f = 0 to 30 kHz, idle channel, Supply signal = 200 mV P-P, narrow band, frequency at PWRO+	−20			dB
Crosstalk attenuation, receive-to-transmit (single-ended)	PCM IN = 0 dB, Frequency = 1 kHz at PCM OUT	60			dB
Signal-to-distortion ratio, sinusoidal input (see Note 8)	ANLG IN = 0 dBm0 to −30 dBm0	33			dB
	ANLG IN = −30 dBm0 to −40 dBm0	27			
	ANLG IN = −40 dBm0 to −45 dBm0	22			
Absolute delay time to PWRO+	Fixed data rate, FCLKX = 2.048 MHz		190		µs

†All typical values are at V_{BB} = −5 V, V_{CC} = 5 V, and T_A = 25°C.

NOTES:
5. Unless otherwise noted, the analog input is a 0-dBm0, 1020-Hz sine wave, where 0 dBm0 is defined as the zero-reference point of the channel under test. This corresponds to an analog signal input of 1.064 V rms, or an output of 1.503 V rms.
6. The input amplifier is set for unity gain. The digital input is a PCM bit stream generated by passing a 0-dBm0, 1020-Hz sine wave through an ideal encoder.
7. The TCM129C18, TCM129C19, TCM29C18, and TCM29C19 are internally connected to set PWRO+ and PWRO− to 0dBm. All output levels are (sin x)/x corrected.
8. CCITT G.712 — Method 2.
9. The receive side (D/A) characteristics are referenced to a 600-Ω termination.

propagation delay times over recommended ranges of operating conditions, fixed-data-rate mode (see timing diagrams)

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{pd1}	From rising edge of transmit clock to bit 1 data valid at PCM OUT (data enable time on time slot entry)	C_L = 0 to 100 pF	0	145	ns
t_{pd2}	From rising edge of transmit clock bit n to bit n+1 data valid at PCM OUT (data valid time)	C_L = 0 to 100 pF	0	145	ns
t_{pd3}	From falling edge of transmit clock bit 8 to bit 8 Hi-Z at PCM OUT (data float time on time slot exit)	C_L = 0	60	215	ns
t_{pd4}	From rising edge of transmit clock bit 1 to TSX active (low) (time slot enable time)	C_L = 0 to 100 pF	0	145	ns
t_{pd5}	From falling edge of transmit clock bit 8 to TSX inactive (high) (timeslot disable time)	C_L = 0	60	190	ns

propagation delay times over recommended ranges of operating conditions, variable-data-rate mode

PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{pd6}	From DCLKX	C_L = 0 to 100 pF	0	100	ns
t_{pd7}	From time slot enable to PCM OUT	C_L = 0 to 100 pF	0	50	ns
t_{pd8}	From time slot disable to PCM OUT	C_L = 0 to 100 pF	0	80	ns
t_{pd9}	From FSX	$t_{d(TSDX)}$ = 140 ns	0	140	ns

clock timing requirements over recommended ranges of supply voltage and operating free-air temperature (see timing diagrams)

PARAMETER		MIN	TYP†	MAX	UNIT
$t_{c(CLK)}$	Clock period for CLK, (2.048-MHz systems)	488			ns
t_r, t_f	Rise and fall times for CLK	5		30	ns
$t_{w(CLK)}$	Pulse duration for CLK	220			ns
$t_{w(DCLK)}$	Pulse duration for DCLK (f_{DCLK} = 64 Hz to 2.048 MHz)	220			ns
	Clock duty cycle [$t_{w(CLK)}/t_{c(CLK)}$] for CLK	45	50	55	%

transmit timing requirements over recommended ranges of supply voltage and operating free-air temperature, fixed-data-rate mode (see timing diagrams)

PARAMETER		MIN	MAX	UNIT
$t_{d(FSX)}$	Frame sync delay time	100	$t_{c(CLK)} - 100$	ns

receive timing requirements over recommended ranges of supply voltage and operating free-air temperature, fixed-data-rate mode (see timing diagrams)

PARAMETER		MIN	MAX	UNIT
$t_{d(FSR)}$	Frame sync delay time	100	$t_{c(CLK)} - 100$	ns

transmit timing requirements over recommended ranges of supply voltage and operating free-air temperature, variable-data-rate mode

PARAMETER		MIN	MAX	UNIT
$t_{d(TSDX)}$	Delay time, timeslot from DCLKX (see Note 10)	140	$t_{w(DCLKX)} - 140$	ns
$t_{d(FSX)}$	Delay time, frame sync	100	$t_{c(CLK)} - 100$	ns
$t_{w(DCLKX)}$	Pulse duration, DCLKX	488	15620	ns

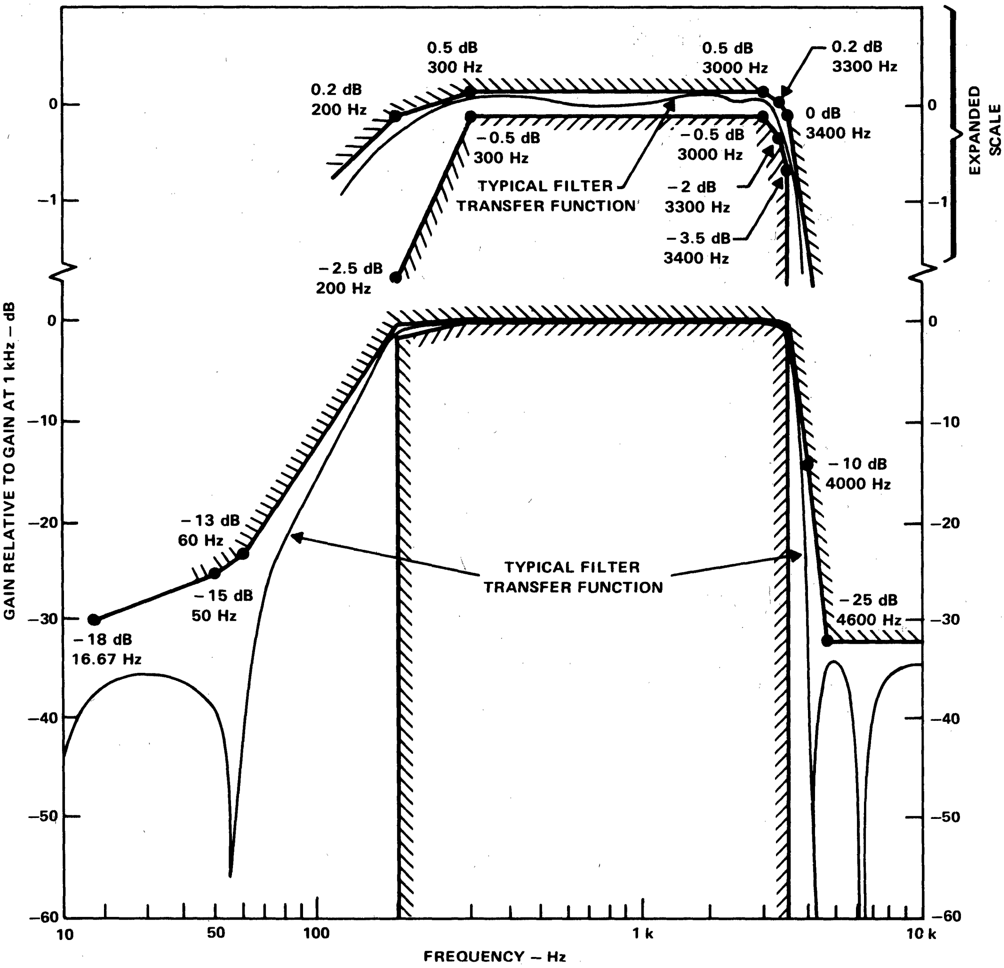
receive timing requirements over recommended ranges of supply voltage and operating free-air temperature, variable-data-rate mode

PARAMETER		MIN	MAX	UNIT
$t_{d(TSDR)}$	Delay time, timeslot from DCLKR (see Note 11)	140	$t_{w(DCLKR)} - 140$	ns
$t_{d(FSR)}$	Delay time, frame sync $T_{C(CLK)}$	100	$t_{c(CLK)} - 100$	ns
$t_{su(PCM\ IN)}$	Setup time, before bit 7 falling edge	10		ns
$t_{h(PCM\ IN)}$	Hold time after bit 8 falling edge	60		ns
$t_{w(DCLKR)}$	Pulse duration, DCLKR	488	15620	ns
$t_{(SER)}$	Time slot end receive time	0		ns

64-kbit operation timing requirements over recommended ranges of supply voltage and operating free-air temperature, variable-data-rate mode

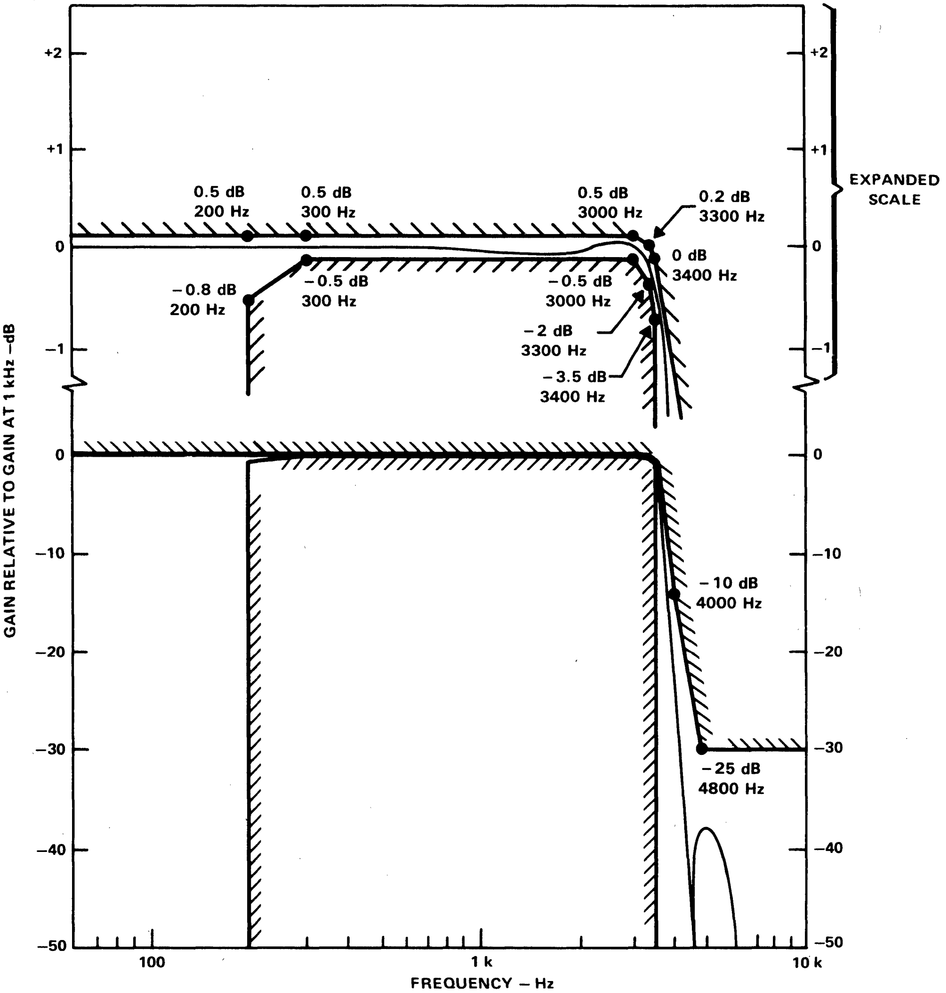
PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
t_{FSLX}	Transmit frame sync minimum down time	FSX = TTL high for remainder of frame	488		ns
t_{FSLR}	Receive frame sync minimum down time	FSR = TTL high for remainder of frame	1952		ns
t_{wCLK}	Pulse duration, data clock			10	μs

NOTES: 10. t_{FSLX} min requirement overrides the $t_{d(TSCDX)}$ max requirement for 64-kHz operation.
11. t_{FSLR} min requirement overrides the $t_{c(TSDR)}$ max requirement for 64-kHz operation.



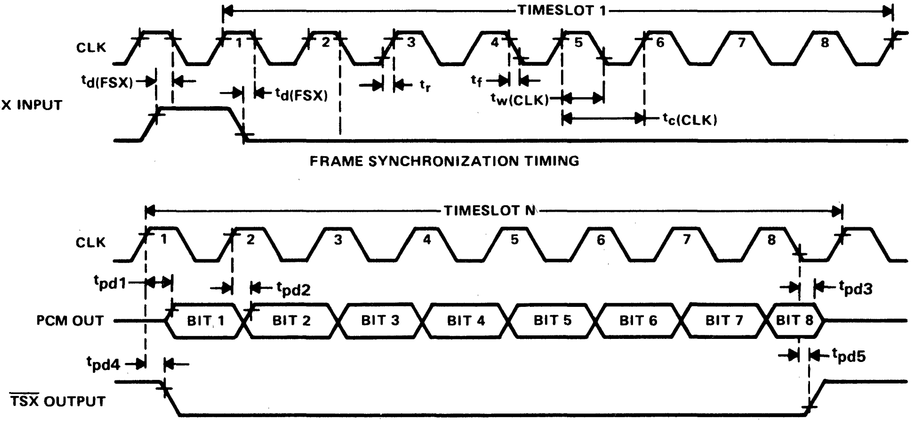
NOTE: This is a typical transfer function of the receiver filter component.

FIGURE 1. TRANSFER CHARACTERISTICS OF THE TRANSMIT FILTER



NOTE: This is a typical transfer function of the receiver filter component.

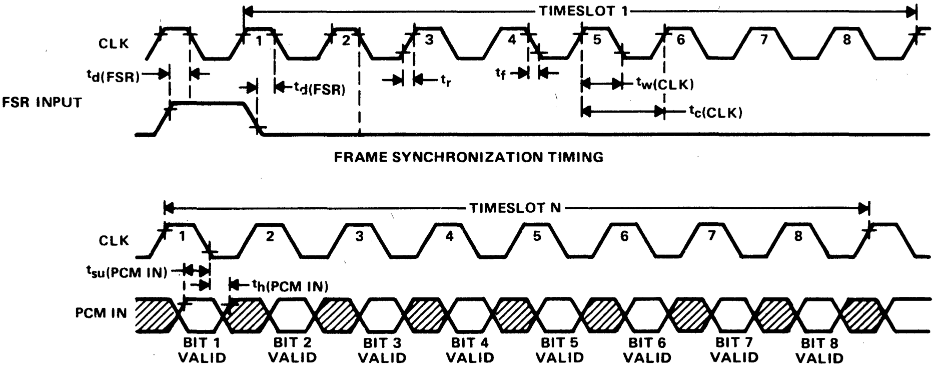
FIGURE 2. TRANSFER CHARACTERISTIC OF THE RECEIVE FILTER



FRAME SYNCHRONIZATION TIMING

OUTPUT TIMING

FIGURE 3. TRANSMIT TIMING (FIXED-DATA-RATE)



FRAME SYNCHRONIZATION TIMING

INPUT TIMING

FIGURE 4. RECEIVE TIMING (FIXED-DATA-RATE)

NOTES: A. Inputs are driven from 0.45 V to 2.4 V. Time intervals are referenced to 2 V if the high level is indicated and 0.8 V if the low level is indicated.

B. Bit 1 is the most significant bit (MSB) and is clocked in first on the PCM IN input or is clocked out first on the PCM OUT output. Bit 8 is the least significant bit (LSB) and is clocked in last on the PCM IN input or is clocked out last on the PCM OUT output.

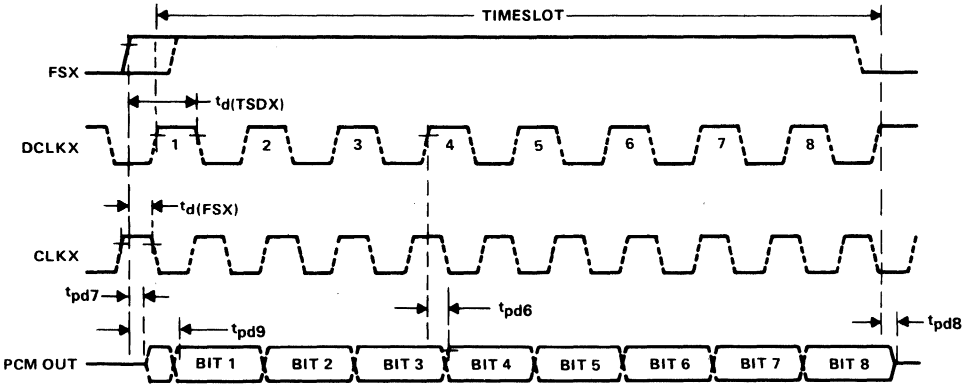
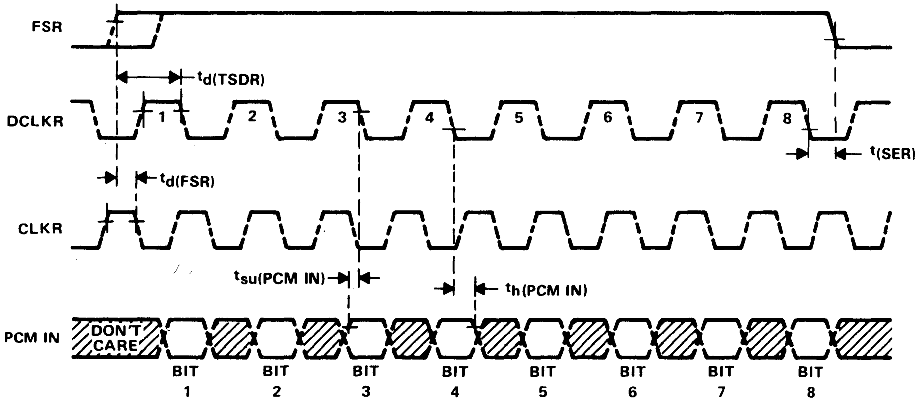


FIGURE 5. TRANSMIT TIMING (VARIABLE-DATA-RATE)



NOTE: All timing parameters referenced to V_{IH} and V_{IL} except t_{pd7} and t_{pd8}, which reference a high-impedance state.

FIGURE 6. RECEIVE TIMING (VARIABLE-DATA-RATE)

GENERAL OPERATION

system reliability features

The TCM129C18, TCM129C19, TCM29C18, and TCM29C19 are powered up in four steps:
V_{CC} and V_{BB} supply voltages are applied.
All clocks are connected.
TTL high is applied to $\overline{PDN}$.
FSX and/or FSR synchronization pulses are applied.

On the transmit channel, digital outputs PCM OUT and $\overline{TSX}$ are held in high-impedance state for approximately four frames (500 μs) after power up or application of V_{BB} or V_{CC}. After this delay, PCM OUT, $\overline{TSX}$, and signaling are functional and will occur in the proper timeslot. The analog circuits on the transmit side require approximately 60 ms to reach their equilibrium value due to the autozero circuit settling time. Thus valid digital information, such as for on/off hook detection, is available almost immediately, while analog information is available after some delay.

To further enhance system reliability, PCM OUT and $\overline{TSX}$ will be placed in a high-impedance state approximately 20 μs after an interruption of CLKX. These interruptions could possibly occur with some kind of fault condition.

power-down and standby operations

To minimize power consumption, a power-down mode and three standby modes are provided.

For power down, an external TTL low signal is applied to the $\overline{PDN}$ pin. It is not sufficient to remove the TTL high voltage to $\overline{PDN}$. In the absence of a signal, the $\overline{PDN}$ pin floats to TTL high and the device remains active. In the power-down mode, the average power consumption is reduced to an average of 5 mW.

The standby modes give the user the option of putting the entire device on standby, putting only the transmit channel on standby, or putting only the receive channel on standby. To place the entire device on standby, both FSX and FSR are held at TTL low. For transmit-only operation, FSX is high and FSR is held low. For receive-only operation, FSR is high and FSX is held low. See Table 1 for power down and standby procedures.

TABLE 1. POWER DOWN AND STANDBY PROCEDURES

DEVICE STATUS	PROCEDURE	TYPICAL POWER CONSUMPTION	DIGITAL OUTPUT STATUS
Power down	$\overline{PDN}$ = TTL low	5 mW	$\overline{TSX}$ and PCM OUT are in a high-impedance state
Entire device on standby	FSX and FSR are TTL low	12 mW	$\overline{TSX}$ and PCM OUT are in a high-impedance state
Only transmit on standby	FSX is TTL low FSR is TTL high	70 mW	$\overline{TSX}$ and PCM OUT are placed in a high-impedance state within 300 ms.
Only receive on standby	FSR is TTL low FSX is TTL high	110 mW	

fixed-data-rate timing (see Figure 3 and 4)

Fixed-data-rate timing is selected by connecting DCLKR to V_{BB}. It uses master clock CLK, frame synchronizer clocks FSX and FSR, and output $\overline{TSX}$. FSX and FSR are 8-kHz inputs that set the sampling frequency. Data is transmitted on the PCM OUT pin on the first eight positive transitions of CLK following the rising edge of FSX. Data is received on the PCM IN pin on the first eight falling edges of CLK following FSX. A digital-to-analog (D/A) conversion is performed on the received digital word and the resulting analog sample is held on an internal sample-and-hold capacitor until transferred to the receive filter.

The TCM129C18 and TCM29C18 operate at 2.048 MHz only. The TCM129C19 and TCM29C19 operate at 1.536 MHz only.

variable data rate timing

Variable-data-rate timing is selected by connecting DCLKR to the bit clock for the receive PCM highway rather than to V_{BB}. It uses master clock CLK, bit clocks DCLKX and DCLKR, and frame synchronization clocks FSX and FSR.

Variable-data-rate timing allows for a flexible data frequency. The frequency of the bit clocks can be varied from 64 kHz to 2.048 MHz. The bit clocks must be synchronous; however, the master clock is restricted to 2.048 MHz.

While FSX/TSXE input is high, PCM data is transmitted from PCM OUT onto the highway on the next eight consecutive positive transitions of DCLKX. Similarly, while the FSR/TSRE input is high, the PCM word is received from the highway by PCM IN on the next eight consecutive negative transitions of DCLKR.

The transmitted PCM word will be repeated in all remaining timeslots in the 125 μs frame as long as DCLKX is pulsed and FSX is held high. This feature, which allows the PCM word to be transmitted to the PCM highway more than once per frame, if desired, is available only with variable-data-rate timing.

asynchronous operation

In either timing mode, the master clock, data clock, and timeslot strobe must be synchronized at the beginning of each frame. Specifically, in the variable-data-rate mode the rising edge of CLK must occur within $t_{d(FSX)}$ ns before the rise of FSX, while the leading edge of DCLKX must occur within t_{TSDX} ns of the rise of FSX. CLK and DCLKX are synchronized once per frame but may be of different frequencies. The receive channel operates in a similar manner and is completely independent of the transmit timing (see variable data rate timing diagrams).

transmit operation

transmit filter

The input section provides gain adjustment in the passband by means of an on-chip uncommitted operational amplifier. The load impedance to ground (ANLG GND) at the amplifier output (GSX) must be greater than 10 kΩ in parallel with less than 50 pF. The input signal on the ANLG IN pin can be either ac or dc coupled.

A low-pass antialiasing section is included on the device. This section provides 35-dB attenuation at the sampling frequency. No external components are required to provide the necessary antialiasing function for the switched capacitor section of the transmit filter.

The passband section provides flatness and stopband attenuation that fulfills the AT&T D3/D4 channel bank transmission specification and CCITT recommendation G.712. The device specifications meet or exceed digital class 5 central office switching systems requirements.

A high-pass section configuration was chosen to reject low-frequency noise from 50- and 60-Hz power lines, 17-Hz European electric railroads, ringing frequencies and their harmonics, and other low-frequency noise. Even with the high rejection at these frequencies, the sharpness of the band edge gives low attenuation at 200 Hz. This feature allows the use of low-cost transformer hybrids without external components.

encoding

The encoder internally samples the output of the transmit filter and holds each sample on an internal sample and hold capacitor. The encoder performs an analog-to-digital conversion on a switched capacitor array. Digital data representing the sample is transmitted on the first eight data clock bits of the next frame.

The autozero circuit corrects for dc offset on the input signal to the encoder. The autozero circuit uses the sign bit averaging technique. The sign bit from the encoder output is long-term averaged and subtracted from the input to the encoder. All dc offset is removed from the encoder input waveform.

receive operation

decoding

The serial PCM word is received at the PCM IN pin on the first eight data clock bits of the frame. Digital-to-analog conversion is performed and the corresponding analog sample is held on an internal sample-and-hold capacitor. This sample is transferred to the receive filter.

receive filter

The receive section of the filter provides passband flatness and stopband rejection that fulfills both the AT&T D3/D4 specification and CCITT recommendation G.712. The filter contains the required compensation for the (sin x)/x response of such decoders.

receive output power amplifiers

A balanced output amplifier is provided to allow maximum flexibility in output configuration. Either of the two outputs can be used single-ended (i.e., referenced to ANLG GND) to drive single-ended loads. Alternatively, the differential output will directly drive a bridged load. The output stage is capable of driving loads as low as 300 ohms single-ended to a level of 12 dBm or 600 ohms differentially to a level of 15 dBm.

Transmission levels are specified relative to the receive channel output under digital milliwatt conditions (i.e., when the digital input at PCM IN is the eight-code sequence specified in CCITT recommendation G.711).

output gain

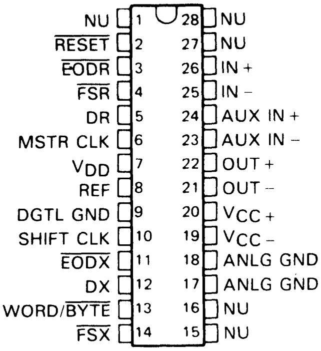
The devices are internally connected to set the PWRO+ and PWRO− to 0 dBm.

TLC32040I, TLC32040C, TLC32041I, TLC32041C, TLC32042I, TLC32042C ANALOG INTERFACE CIRCUITS

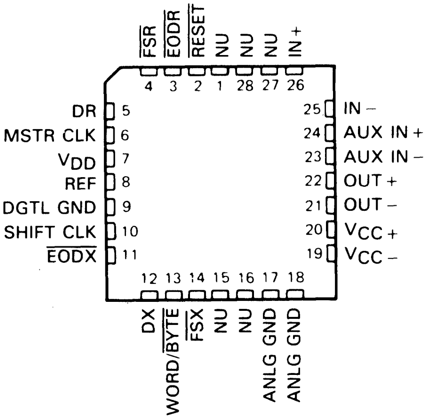
D2964, SEPTEMBER 1987—REVISED SEPTEMBER 1988

- Advanced LinCMOS™ Silicon-Gate Process Technology
- 14-Bit Dynamic Range ADC and DAC
- Variable ADC and DAC Sampling Rate Up to 19,200 Samples per Second
- Switched-Capacitor Antialiasing Input Filter and Output-Reconstruction Filter
- Serial Port for Direct Interface to TMS32011, TMS320C17, TMS32020, and TMS320C25 Digital Processors
- Synchronous or Asynchronous ADC and DAC Conversion Rates with Programmable Incremental ADC and DAC Conversion Timing Adjustments
- Serial Port Interface to SN74299 Serial-to-Parallel Shift Register for Parallel Interface to TMS32010, TMS320C15, or Other Digital Processors

N PACKAGE
(TOP VIEW)



FN PACKAGE
(TOP VIEW)



NU—Nonusable; no external connection should be made to these pins.

PART NUMBER	DESCRIPTION
TLC32040	Analog Interface Circuit with internal reference. Also a plug-in replacement for TLC32041.
TLC32041	Analog Interface Circuit without internal reference.
TLC32042	Identical to TLC32040, but has a slightly wider bandpass filter bandwidth

description

The TLC32040, TLC32041, and TLC32042 are complete analog-to-digital and digital-to-analog input/output systems, each on a single monolithic CMOS chip. This device integrates a bandpass switched-capacitor antialiasing input filter, a 14-bit-resolution A/D converter, four microprocessor-compatible serial port modes, a 14-bit-resolution D/A converter, and a low-pass switched-capacitor output-reconstruction filter. The device offers numerous combinations of Master Clock input frequencies and conversion/sampling rates, which can be changed via digital processor control.

Typical applications for this IC include modems (7.2-, 8-, 9.6-, 14.4-, and 19.2-kHz sampling rate), analog interface for digital signal processors (DSPs), speech recognition/storage systems, industrial process control, biomedical instrumentation, acoustical signal processing, spectral analysis, data acquisition, and instrumentation recorders. Four serial modes, which allow direct interface to the TMS32011, TMS320C17, TMS32020, and TMS320C25 digital signal processors, are provided. Also, when the transmit and receive sections of the Analog Interface Circuit (AIC) are operating synchronously, it will interface to two SN74299

Advanced LinCMOS™ is a trademark of Texas Instruments Incorporated.

description (continued)

serial-to-parallel shift registers. These serial-to-parallel shift registers can then interface in parallel to the TMS32010, TMS320C15, other digital signal processors, or external FIFO circuitry. Output data pulses are emitted to inform the processor that data transmission is complete or to allow the DSP to differentiate between two transmitted bytes. A flexible control scheme is provided so that the functions of the IC can be selected and adjusted coincidentally with signal processing via software control.

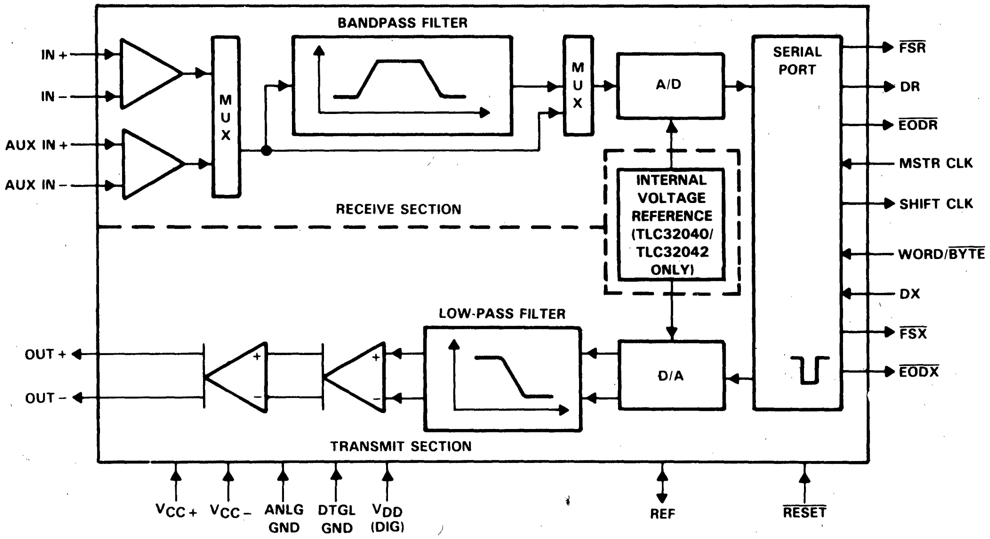
The antialiasing input filter comprises seventh-order and fourth-order CC-type (Chebyshev/elliptic transitional) low-pass and high-pass filters, respectively, and a fourth-order equalizer. The input filter is implemented in switched-capacitor technology and is preceded by a continuous time filter to eliminate any possibility of aliasing caused by sampled data filtering. When no filtering is desired, the entire composite filter can be switched out of the signal path. A selectable, auxiliary, differential analog input is provided for applications where more than one analog input is required.

The A/D and D/A converters each have 14 bits of resolution. The A/D and D/A architectures ensure no missing codes and monotonic operation. An internal voltage reference is provided on the TLC32040 and TLC32042 to ease the design task and to provide complete control over the performance of the IC. The internal voltage reference is brought out to a pin and is available to the designer. Separate analog and digital voltage supplies and grounds are provided to minimize noise and ensure a wide dynamic range. Also, the analog circuit path contains only differential circuitry to keep noise to an absolute minimum. The only exception is the DAC sample-and-hold, which utilizes pseudo-differential circuitry.

The output-reconstruction filter is a seventh-order CC-type (Chebyshev/elliptic transitional low-pass filter with a fourth-order equalizer) and is implemented in switched-capacitor technology. This filter is followed by a continuous-time filter to eliminate images of the digitally encoded signal.

The TLC32040I, TLC32041I, and TLC32042I are characterized for operation from −40 °C to 85 °C, and the TLC32040C, TLC32041C, and TLC32042C are characterized for operation from 0 °C to 70 °C.

functional block diagram



PRINCIPLES OF OPERATION

analog input

Two sets of analog inputs are provided. Normally, the IN + and IN − input set is used; however, the auxiliary input set, AUX IN + and AUX IN −, can be used if a second input is required. Each input set can be operated in either differential or single-ended modes, since sufficient common-mode range and rejection are provided. The gain for the IN +, IN −, AUX IN +, and AUX IN − inputs can be programmed to be either 1, 2, or 4 (see Table 2). Either input circuit can be selected via software control. It is important to note that a wide dynamic range is assured by the differential internal analog architecture and by the separate analog and digital voltage supplies and grounds.

A/D bandpass filter, A/D bandpass filter clocking, and A/D conversion timing

The A/D bandpass filter can be selected or bypassed via software control. The frequency response of this filter is presented in the following pages. This response results when the switched-capacitor filter clock frequency is 288 kHz. Several possible options can be used to attain a 288-kHz switched-capacitor filter clock. When the filter clock frequency is not 288 kHz, the filter transfer function is frequency-scaled by the ratio of the actual clock frequency to 288 kHz. The low-frequency roll-off of the high-pass section is 300 Hz. However, the high-pass section low-frequency roll-off is less steep for the TLC32042 than for the TLC32040 and TLC32041.

The Internal Timing Configuration and AIC DX Data Word Format sections of this data sheet indicate the many options for attaining a 288-kHz bandpass switched-capacitor filter clock. These sections indicate that the RX Counter A can be programmed to give a 288-kHz bandpass switched-capacitor filter clock for several Master Clock input frequencies.

The A/D conversion rate is then attained by frequency-dividing the 288-kHz bandpass switched-capacitor filter clock with the RX Counter B. Thus, unwanted aliasing is prevented because the A/D conversion rate is an integral submultiple of the bandpass switched-capacitor filter sampling rate, and the two rates are synchronously locked.

A/D converter performance specifications

Fundamental performance specifications for the A/D converter circuitry are presented in the A/D converter operating characteristics section of this data sheet. The realization of the A/D converter circuitry with switched-capacitor techniques provides an inherent sample-and-hold.

analog output

The analog output circuitry is an analog output power amplifier. Both noninverting and inverting amplifier outputs are brought out of the IC. This amplifier can drive transformer hybrids or low-impedance loads directly in either a differential or single-ended configuration.

D/A low-pass filter, D/A low-pass filter clocking, and D/A conversion timing

The frequency response of this filter is presented in the following pages. This response results when the low-pass switched-capacitor filter clock frequency is 288 kHz. Like the A/D filter, the transfer function of this filter is frequency-scaled when the clock frequency is not 288 kHz. A continuous-time filter is provided on the output of the D/A low-pass filter to greatly attenuate any switched-capacitor clock feedthrough.

The D/A conversion rate is then attained by frequency-dividing the 288-kHz switched-capacitor filter clock with TX Counter B. Thus, unwanted aliasing is prevented because the D/A conversion rate is an integral submultiple of the switched-capacitor low-pass filter sampling rate, and the two rates are synchronously locked.

PRINCIPLES OF OPERATION (continued)

asynchronous versus synchronous operation

If the transmit section of the AIC (low-pass filter and DAC) and receive section (bandpass filter and ADC) are operated asynchronously, the low-pass and band-pass filter clocks are independently generated from the Master Clock signal. Also, the D/A and A/D conversion rates are independently determined. If the transmit and receive sections are operated synchronously, the low-pass filter clock drives both low-pass and bandpass filters. In synchronous operation, the A/D conversion timing is derived from, and is equal to, the D/A conversion timing. (See description of the WORD/$\overline{\text{BYTE}}$ pin in the Pin Functional Description Section.)

D/A converter performance specifications

Fundamental performance specifications for the D/A converter circuitry are presented in the D/A converter operating characteristics section of the data sheet. The D/A converter has a sample-and-hold that is realized with a switched-capacitor ladder.

system frequency response correction

Sin x/x correction circuitry is performed in digital signal processor software. The system frequency response can be corrected via DSP software to ±0.1 dB accuracy to a band-edge of 3000 Hz for all sampling rates. This correction is accomplished with a first-order digital correction filter, which requires only seven TMS320 instruction cycles. With a 200-ns instruction cycle, seven instructions represent an overhead factor of only 1.1% and 1.3% for sampling rates of 8 and 9.6 kHz, respectively (see the sin x/x Correction Section for more details).

serial port

The serial port has four possible modes that are described in detail in the Functional Pin Description Section. These modes are briefly described below and in the Functional Description for Pin 13, WORD/$\overline{\text{BYTE}}$.

1. The transmit and receive sections are operated asynchronously, and the serial port interfaces directly with the TMS32011 and TMS320C17.
2. The transmit and receive sections are operated asynchronously, and the serial port interfaces directly with the TMS32020 and the TMS320C25.
3. The transmit and receive sections are operated synchronously, and the serial port interfaces directly with the TMS32011 and TMS320C17.
4. The transmit and receive sections are operated synchronously, and the serial port interfaces directly with the TMS32020, TMS320C25, or two SN74299 serial-to-parallel shift registers, which can then interface in parallel to the TMS32010, TMS320C15, to any other digital signal processor, or to external FIFO circuitry.

testing

An addendum accompanying this data sheet fully describes the test capabilities of the IC, provided by the design.

operation of TLC32040 or TLC32042 with internal voltage reference

The internal reference of the TLC32040 and TLC32042 eliminates the need for an external voltage reference and provides overall circuit cost reduction. Thus, the internal reference eases the design task and provides complete control over the performance of the IC. The internal reference is brought out to a pin and is available to the designer. To keep the amount of noise on the reference signal to a minimum, an external capacitor may be connected between REF and ANLG GND.

PRINCIPLES OF OPERATION (continued)

operation of TLC32040, TLC32041, or TLC32042 with external voltage reference

The REF pin may be driven from an external reference circuit if so desired. This external circuit must be capable of supplying 250 μA and must be adequately protected from noise such as crosstalk from the analog input.

reset

A reset function is provided to initiate serial communications between the AIC and DSP and to allow fast, cost-effective testing during manufacturing. The reset function will initialize all AIC registers, including the control register. After a negative-going pulse on the $\overline{\text{RESET}}$ pin, the AIC will be initialized. This initialization allows normal serial port communications activity to occur between AIC and DSP (see AIC DX Data Word Format section).

loopback

This feature allows the user to test the circuit remotely. In loopback, the OUT + and OUT − pins are internally connected to the IN + and IN − pins. Thus, the DAC bits (d15 to d2), which are transmitted to the DX pin, can be compared with the ADC bits (d15 to d2), which are received from the DR pin. An ideal comparison would be that the bits on the DR pin equal the bits on the DX pin. However, in practice there will be some difference in these bits due to the ADC and DAC output offsets.

In loopback, if the IN + and IN − pins are enabled, the external signals on the IN + and IN − pins are ignored. If the AUX IN + and AUX IN − pins are enabled, the external signals on these pins are added to the OUT + and OUT − signals in loopback operation.

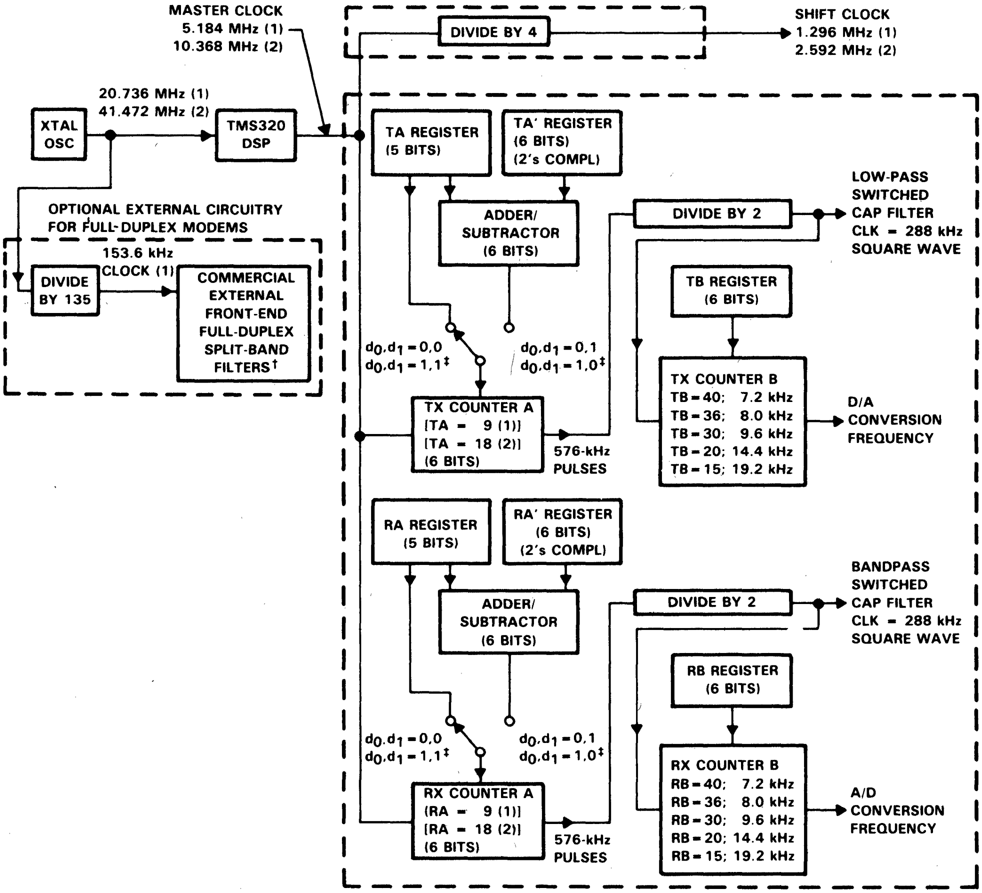
The loopback feature is implemented with digital signal processor control by transmitting the appropriate serial port bit to the control register (see AIC Data Word Format section).

PIN		I/O	DESCRIPTION
NAME	NO.		
ANLG GND	17,18		Analog ground return for all internal analog circuits. Not internally connected to DGTL GND.
AUX IN +	24	I	Noninverting auxiliary analog input stage. This input can be switched into the bandpass filter and A/D converter path via software control. If the appropriate bit in the Control register is a 1, the auxiliary inputs will replace the IN + and IN − inputs. If the bit is a 0, the IN + and IN − inputs will be used (see the AIC DX Data Word Format section).
AUX IN −	23	I	Inverting auxiliary analog input (see the above AUX IN + pin description).
DGTL GND	9		Digital ground for all internal logic circuits. Not internally connected to ANLG GND.
DR	5	O	This pin is used to transmit the ADC output bits from the AIC to the TMS320 serial port. This transmission of bits from the AIC to the TMS320 serial port is synchronized with the SHIFT CLK signal.
DX	12	I	This pin is used to receive the DAC input bits and timing and control information from the TMS320. This serial transmission from the TMS320 serial port to the AIC is synchronized with the SHIFT CLK signal.
$\overline{\text{EODR}}$	3	O	(See the WORD/$\overline{\text{BYTE}}$ pin description and the Serial Port Timing Diagram.) During the word-mode timing, this signal is a low-going pulse that occurs immediately after the 16 bits of A/D information have been transmitted from the AIC to the TMS320 serial port. This signal can be used to interrupt a microprocessor upon completion of serial communications. Also, this signal can be used to strobe and enable external serial-to-parallel shift registers, latches, or external FIFO RAM, and to facilitate parallel data bus communications between the AIC and the serial-to-parallel shift registers. During the byte-mode timing, this signal goes low after the first byte has been transmitted from the AIC to the TMS320 serial port and is kept low until the second byte has been transmitted. The TMS32011 or TMS320C17 can use this low-going signal to differentiate between the two bytes as to which is first and which is second.

PIN NAME	 NO.	I/O	DESCRIPTION
$\overline{\text{EODX}}$	11	O	(See the WORD/$\overline{\text{BYTE}}$ pin description and the Serial Port Timing Diagram.) During the word-mode timing, this signal is a low-going pulse that occurs immediately after the 16 bits of D/A converter and control or register information have been transmitted from the TMS320 serial port to the AIC. This signal can be used to interrupt a microprocessor upon the completion of serial communications. Also, this signal can be used to strobe and enable external serial-to-parallel shift registers, latches, or an external FIFO RAM, and to facilitate parallel data-bus communications between the AIC and the serial-to-parallel shift registers. During the byte-mode timing, this signal goes low after the first byte has been transmitted from the TMS320 serial port to the AIC and is kept low until the second byte has been transmitted. The TMS32011 or TMS320C17 can use this low-going signal to differentiate between the two bytes as to which is first and which is second.
$\overline{\text{FSR}}$	4	O	In the serial transmission modes, which are described in the WORD/$\overline{\text{BYTE}}$ pin description, the $\overline{\text{FSR}}$ pin is held low during bit transmission. When the $\overline{\text{FSR}}$ pin goes low, the TMS320 serial port will begin receiving bits from the AIC via the DR pin of the AIC. The most significant DR bit will be present on the DR pin before $\overline{\text{FSR}}$ goes low. (See Serial Port Timing and Internal Timing Configuration Diagrams.)
$\overline{\text{FSX}}$	14	O	When this pin goes low, the TMS320 serial port will begin transmitting bits to the AIC via the DX pin of the AIC. In all serial transmission modes, which are described in the WORD/$\overline{\text{BYTE}}$ pin description, the $\overline{\text{FSX}}$ pin is held low during bit transmission (see Serial Port Timing and Internal Timing Configuration Diagrams).
IN +	26	I	Noninverting input to analog input amplifier stage
IN −	25	I	Inverting input to analog input amplifier stage
MSTR CLK	6	I	The Master Clock signal is used to derive all the key logic signals of the AIC, such as the Shift Clock, the switched-capacitor filter clocks, and the A/D and D/A timing signals. The Internal Timing Configuration diagram shows how these key signals are derived. The frequencies of these key signals are synchronous submultiples of the Master Clock frequency to eliminate unwanted aliasing when the sampled analog signals are transferred between the switched-capacitor filters and the A/D and D/A converters (see the Internal Timing Configuration).
OUT +	22	O	Noninverting output of analog output power amplifier. Can drive transformer hybrids or high-impedance loads directly in either a differential or a single-ended configuration.
OUT −	21	O	Inverting output of analog output power amplifier. Functionally identical with and complementary to OUT +.
REF	8	I/O	For the TLC32040 and TLC32042, the internal voltage reference is brought out on this pin. For the TLC32040, TLC32041, and TLC32042, an external voltage reference can be applied to this pin.
$\overline{\text{RESET}}$	2	I	A reset function is provided to initialize the TA, TA′, TB, RA, RA′, RB, and control registers. This reset function initiates serial communications between the AIC and DSP. The reset function will initialize all AIC registers including the control register. After a negative-going pulse on the $\overline{\text{RESET}}$ pin, the AIC registers will be initialized to provide an 8-kHz data conversion rate for a 5.184-MHz master clock input signal. The conversion rate adjust registers, TA′ and RA′, will be reset to 1. The CONTROL register bits will be reset as follows (see AIC DX Data Word Format section). d7 = 1, d6 = 1, d5 = 1, d4 = 0, d3 = 0, d2 = 1 This initialization allows normal serial-port communication to occur between AIC and DSP.
SHIFT CLK	10	O	The Shift Clock signal is obtained by dividing the Master Clock signal frequency by four. This signal is used to clock the serial data transfers of the AIC, described in the WORD/$\overline{\text{BYTE}}$ pin description below (see the Serial Port Timing and Internal Timing Configuration diagram).
V_{DD}	7		Digital supply voltage, 5 V ±5%
V_{CC+}	20		Positive analog supply voltage, 5 V ±5%
V_{CC-}	19		Negative analog supply voltage −5 V ±5%

PIN		I/O	DESCRIPTION
NAME	NO.		
WORD/$\overline{\text{BYTE}}$	13	I	This pin, in conjunction with a bit in the CONTROL register, is used to establish one of four serial modes. These four serial modes are described below. *AIC transmit and receive sections are operated asynchronously.* The following description applies when the AIC is configured to have asynchronous transmit and receive sections. If the appropriate data bit in the Control register is a 0 (see the AIC DX Data Word Format), the transmit and receive sections will be asynchronous. L Serial port directly interfaces with the serial port of the TMS32011 or TMS320C17 and communicates in two 8-bit bytes. The operation sequence is as follows (see Serial Port Timing diagrams). 1. The $\overline{\text{FSX}}$ or $\overline{\text{FSR}}$ pin is brought low. 2. One 8-bit byte is transmitted or one 8-bit byte is received. 3. The $\overline{\text{EODX}}$ or $\overline{\text{EODR}}$ pin is brought low. 4. The $\overline{\text{FSX}}$ or $\overline{\text{FSR}}$ pin emits a positive frame-sync pulse that is four Shift Clock cycles wide. 5. One 8-bit byte is transmitted or one 8-bit byte is received. 6. The $\overline{\text{EODX}}$ or $\overline{\text{EODR}}$ pin is brought high. 7. The $\overline{\text{FSX}}$ or $\overline{\text{FSR}}$ pin is brought high. H Serial port directly interfaces with the serial port of the TMS32020, TMS320C25, or TMS320C30 and communicates in one 16-bit word. The operation sequence is as follows (see Serial Port Timing diagrams): 1. The $\overline{\text{FSX}}$ or $\overline{\text{FSR}}$ pin is brought low. 2. One 16-bit word is transmitted or one 16-bit word is received. 3. The $\overline{\text{FSX}}$ or $\overline{\text{FSR}}$ pin is brought high. 4. The $\overline{\text{EODX}}$ or $\overline{\text{EODR}}$ pin emits a low-going pulse. *AIC transmit and receive sections are operated synchronously.* If the appropriate data bit in the Control register is a 1, the transmit and receive sections will be configured to be synchronous. In this case, the bandpass switched-capacitor filter and the A/D conversion timing will be derived from the TX Counter A, TX Counter B, and TA, TA′, and TB registers, rather than the RX Counter A, RX Counter B, and RA, RA′, and RB registers. In this case, the AIC $\overline{\text{FSX}}$ and $\overline{\text{FSR}}$ timing will be identical during primary data communication; however, $\overline{\text{FSR}}$ will not be asserted during secondary data communication since there is no new A/D conversion result. The synchronous operation sequences are as follows (see Serial Port Timing diagrams). L Serial port directly interfaces with the serial port of the TMS32011 or TMS320C17 and communicates in two 8-bit bytes. The operation sequence is as follows (see Serial Port Timing diagrams): 1. The $\overline{\text{FSX}}$ and $\overline{\text{FSR}}$ pins are brought low. 2. One 8-bit byte is transmitted and one 8-bit byte is received. 3. The $\overline{\text{EODX}}$ and $\overline{\text{EODR}}$ pins are brought low. 4. The $\overline{\text{FSX}}$ and $\overline{\text{FSR}}$ pins emit positive frame-sync pulses that are four Shift Clock cycles wide. 5. One 8-bit byte is transmitted and one 8-bit byte is received. 6. The $\overline{\text{EODX}}$ and $\overline{\text{EODR}}$ pins are brought high. 7. The $\overline{\text{FSX}}$ and $\overline{\text{FSR}}$ pins are brought high. H Serial port directly interfaces with the serial port of the TMS32020, TMS320C25, or TMS320C30 and communicates in one 16-bit word. The operation sequence is as follows (see Serial Port Timing diagrams): 1. The $\overline{\text{FSX}}$ and $\overline{\text{FSR}}$ pins are brought low. 2. One 16-bit word is transmitted and one 16-bit word is received. 3. The $\overline{\text{FSX}}$ and $\overline{\text{FSR}}$ pins are brought high. 4. The $\overline{\text{EODX}}$ or $\overline{\text{EODR}}$ pins emit low-going pulses. Since the transmit and receive sections of the AIC are now synchronous, the AIC serial port, with additional NOR and AND gates, will interface to two SN74299 serial-to-parallel shift registers. Interfacing the AIC to the SN74299 shift register allows the AIC to interface to an external FIFO RAM and facilitates parallel, data bus communications between the AIC and the digital signal processor. The operation sequence is the same as the above sequence (see Serial Port Timing diagrams).

INTERNAL TIMING CONFIGURATION



$$\text{SCF Clock Frequency} = \frac{\text{Master Clock Frequency}}{2 \times \text{Contents of Counter A}}$$

NOTE: Frequency 1, 20.736 MHz, is used to show how 153.6 kHz (for a commercially available modem split-band filter clock), popular speech and modem sampling signal frequencies, and an internal 288-kHz switched-capacitor filter clock can be derived synchronously and as submultiples of the crystal oscillator frequency. Since these derived frequencies are synchronous submultiples of the crystal frequency, aliasing does not occur as the sampled analog signal passes between the analog converter and switched-capacitor filter stages. Frequency 2, 41.472 MHz, is used to show that the AIC can work with high-frequency signals, which are used by high-speed digital signal processors.

†Split-band filtering can alternatively be performed after the analog input function via software in the TMS320.

‡These control bits are described in the AIC DX Data Word Format section.

explanation of internal timing configuration

All of the internal timing of the AIC is derived from the high-frequency clock signal that drives the Master Clock input pin. The Shift Clock signal, which strobes the serial port data between the AIC and DSP, is derived by dividing the Master Clock input signal frequency by four.

$$\text{SCF Clock Frequency} = \frac{\text{Master Clock Frequency}}{2 \times \text{Contents of Counter A}}$$

$$\text{Conversion Frequency} = \frac{\text{SCF Clock Frequency}}{\text{Contents of Counter B}}$$

$$\text{Shift Clock Frequency} = \frac{\text{Master Clock Frequency}}{4}$$

TX Counter A and TX Counter B, which are driven by the Master Clock signal, determine the D/A conversion timing. Similarly, RX Counter A and RX Counter B determine the A/D conversion timing. In order for the switched-capacitor low-pass and bandpass filters to meet their transfer function specifications, the frequency of the clock inputs of the switched-capacitor filters must be 288 kHz. If the frequencies of the clock inputs are not 288 kHz, the filter transfer function frequencies are scaled by the ratios of the clock frequencies to 288 kHz. Thus, to obtain the specified filter responses, the combination of Master Clock frequency and TX Counter A and RX Counter A values must yield 288-kHz switched-capacitor clock signals. These 288-kHz clock signals can then be divided by the TX Counter B and RX Counter B to establish the D/A and A/D conversion timings.

TX Counter A and TX Counter B are reloaded every D/A conversion period, while RX Counter A and RX Counter B are reloaded every A/D conversion period. The TX Counter B and RX Counter B are loaded with the values in the TB and RB Registers, respectively. Via software control, the TX Counter A can be loaded with either the TA Register, the TA Register less the TA′ Register, or the TA Register plus the TA′ Register. By selecting the TA Register less the TA′ Register option, the upcoming conversion timing will occur earlier by an amount of time that equals TA′ times the signal period of the Master Clock. By selecting the TA Register plus the TA′ Register option, the upcoming conversion timing will occur later by an amount of time that equals TA′ times the signal period of the Master Clock. Thus, the D/A conversion timing can be advanced or retarded. An identical ability to alter the A/D conversion timing is provided. In this case, however, the RX Counter A can be programmed via software control with the RA Register, the RA Register less the RA′ Register, or the RA Register plus the RA′ Register.

The ability to advance or retard conversion timing is particularly useful for modem applications. This feature allows controlled changes in the A/D and D/A conversion timing. This feature can be used to enhance signal-to-noise performance, to perform frequency-tracking functions, and to generate nonstandard modem frequencies.

If the transmit and receive sections are configured to be synchronous (see WORD/$\overline{\text{BYTE}}$ pin description), then both the low-pass and bandpass switched-capacitor filter clocks are derived from TX Counter A. Also, both the D/A and A/D conversion timing are derived from the TX Counter A and TX Counter B. When the transmit and receive sections are configured to be synchronous, the RX Counter A, RX Counter B, RA Register, RA′ Register, and RB Registers are not used.

AIC DR or DX word bit pattern

A/D or D/A MSB,
1st bit sent ↓ (D15) — 1st bit sent of 2nd byte ↓ (D7) — A/D or D/A LSB ↓ (D2)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

AIC DX data word format section

d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d2	d1	d0	COMMENTS
primary DX serial communication protocol															
← d15 (MSB) through d2 go to the D/A converter register →													0	0	The TX and RX Counter A's are loaded with the TA and RA register values. The TX and RX Counter B's are loaded with TB and RB register values.
← d15 (MSB) through d2 go to the D/A converter register →													0	1	The TX and RX Counter A's are loaded with the TA + TA' and RA + RA' register values. The TX and RX Counter B's are loaded with the TB and RB register values. NOTE: d1 = 0, d0 = 1 will cause the next D/A and A/D conversion periods to be changed by the addition of TA' and RA' Master Clock cycles, in which TA' and RA' can be positive or negative or zero. Please refer to Table 1. AIC Responses to Improper Conditions.
← d15 (MSB) through d2 go to the D/A converter register →													1	0	The TX and RX Counter A's are loaded with the TA − TA' and RA − RA' register values. The TX and RX Counter B's are loaded with the TB and RB register values. NOTE: d1 = 1, d0 = 0 will cause the next D/A and A/D conversion periods to be changed by the subtraction of TA' and RA' Master Clock cycles, in which TA' and RA' can be positive or negative or zero. Please refer to Table 1. AIC Responses to Improper Conditions.
← d15 (MSB) through d2 go to the D/A converter register →													1	1	The TX and RX Counter A's are loaded with the TA and RA register values. The TX and RX Counter B's are loaded with the TB and RB register values. After a delay of four Shift Clock cycles, a secondary transmission will immediately follow to program the AIC to operate in the desired configuration.

NOTE: Setting the two least significant bits to 1 in the normal transmission of DAC information (Primary Communications) to the AIC will initiate Secondary Communications upon completion of the Primary Communications.

Upon completion of the Primary Communication, $\overline{\text{FSX}}$ will remain high for four SHIFT CLOCK cycles and will then go low and initiate the Secondary Communication. The timing specifications for the Primary and Secondary Communications are identical. In this manner, the Secondary Communication, if initiated, is interleaved between successive Primary Communications. This interleaving prevents the Secondary Communication from interfering with the Primary Communications and DAC timing, thus preventing the AIC from skipping a DAC output. It is important to note that in the synchronous mode, $\overline{\text{FSR}}$ will not be asserted during Secondary Communications.

secondary DX serial communication protocol

x x | ← to TA register →| x x| ← to RA register → | 0 0	d13 and d6 are MSBs (unsigned binary)
x | ← to TA′ register →| x | ← to RA′ register → | 0 1	d14 and d7 are 2's complement sign bits
x | ← to TB register →| x | ← to RB register → | 1 0	d14 and d7 are MSBs (unsigned binary)

x x x x x x x x d7 d6 d5 d4 d3 d2 1 1
|← CONTROL REGISTER →|
	d2 = 0/1 deletes/inserts the bandpass filter
	d3 = 0/1 disables/enables the loopback function
	d4 = 0/1 disables/enables the AUX IN+ and AUX IN− pins
	d5 = 0/1 asynchronous/synchronous transmit and receive sections
	d6 = 0/1 gain control bits (see Gain Control Section)
	d7 = 0/1 gain control bits (see Gain Control Section)

reset function

A reset function is provided to initiate serial communications between the AIC and DSP. The reset function will initialize all AIC registers, including the control register. After power has been applied to the AIC, a negative-going pulse on the $\overline{\text{RESET}}$ pin will initialize the AIC registers to provide an 8-kHz A/D and D/A conversion rate for a 5.184 MHz master clock input signal. The AIC, excepting the CONTROL register, will be initialized as follows (see AIC DX Data Word Format section):

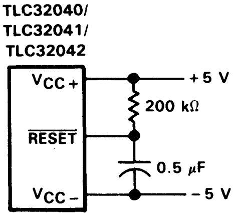
REGISTER	INITIALIZED REGISTER VALUE (HEX)
TA	9
TA′	1
TB	24
RA	9
RA′	1
RB	24

The CONTROL register bits will be reset as follows (see AIC DX Data Word Format section):

d7 = 1, d6 = 1, d5 = 1, d4 = 0, d3 = 0, d2 = 1

This initialization allows normal serial port communications to occur between AIC and DSP. If the transmit and receive sections are configured to operate synchronously and the user wishes to program different conversion rates, only the TA, TA′, and TB register need to be programmed, since both transmit and receive timing are synchronously derived from these registers (see the Pin Descriptions and AIC DX Word Format sections).

The circuit shown below will provide a reset on power-up when power is applied in the sequence given under Power-Up Sequence. The circuit depends on the power supplies' reaching their recommended values a minimum of 800 ns before the capacitor charges to 0.8 V above DGTL GND.



power-up sequence

To ensure proper operation of the AIC, and as a safeguard against latch-up, it is recommmended that a Schottky diode with a forward voltage less than or equal to 0.4 V be connected from V_{CC-} to ANLG GND (see Figure 17). In the absence of such a diode, power should be applied in the following sequence: ANLG GND and DGTL GND, V_{CC-}, then V_{CC+} and V_{DD}. Also, no input signal should be applied until after power-up.

AIC responses to improper conditions

The AIC has provisions for responding to improper conditions. These improper conditions and the response of the AIC to these conditions are presented in Table 1 below.

AIC register constraints

The following constraints are placed on the contents of the AIC registers:

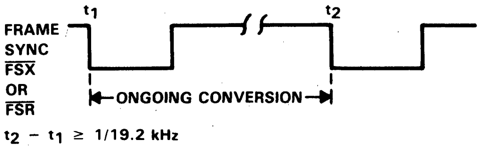
1. TA register must be > 1.
2. TA′ register can be either positive, negative, or zero.
3. RA register must be > 1.
4. RA′ register can be either positive, negative, or zero.
5. (TA register ± TA′ register) must be > 1.
6. (RA register ± RA′ register) must be > 1.
7. TB register must be > 1.

TABLE 1. AIC RESPONSES TO IMPROPER CONDITIONS

IMPROPER CONDITION	AIC RESPONSE
TA register + TA′ register = 0 or 1 TA register − TA′ register = 0 or 1	Reprogram TX Counter A with TA register value
TA register + TA′ register < 0	MODULO 64 arithmetic is used to ensure that a positive value is loaded into the TX Counter A, i.e., TA register + TA′ register + 40 HEX is loaded into TX Counter A
RA register + RA′ register = 0 or 1 RA register − RA′ register = 0 or 1	Reprogram RX Counter A with RA register value
RA register + RA′ register = 0 or 1	MODULO 64 arithmetic is used to ensure that a positive value is loaded into RX Counter A, i.e., RA register + RA′ register + 40 HEX is loaded into RX Counter A
TA register = 0 or 1 RA register = 0 or 1	AIC is shut down
TB register = 0 or 1	Reprogram TB register with 24 HEX
RB register = 0 or 1	Reprogram RB register with 24 HEX
AIC and DSP cannot communicate	Hold last DAC output

improper operation due to conversion times being too close together

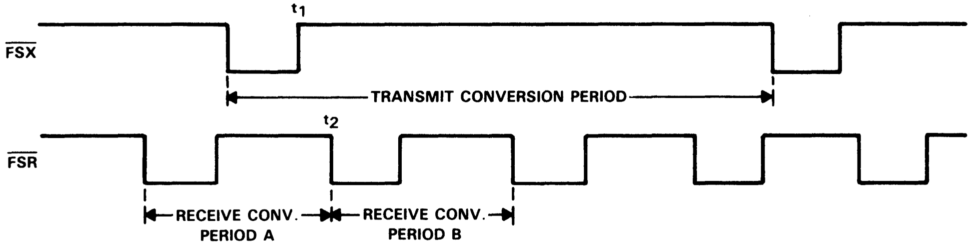
If the difference between two successive D/A conversion frame syncs is less that 1/19.2 kHz, the AIC operates improperly. In this situation, the second D/A conversion frame sync occurs too quickly and there is not enough time for the ongoing conversion to be completed. This situation can occur if the A and B registers are improperly programmed or if the A + A′ register or A − A′ register result is too small. When incrementally adjusting the conversion period via the A + A′ register options, the designer should be very careful not to violate this requirement (see diagram below).



$t_2 - t_1 \geq 1/19.2$ kHz

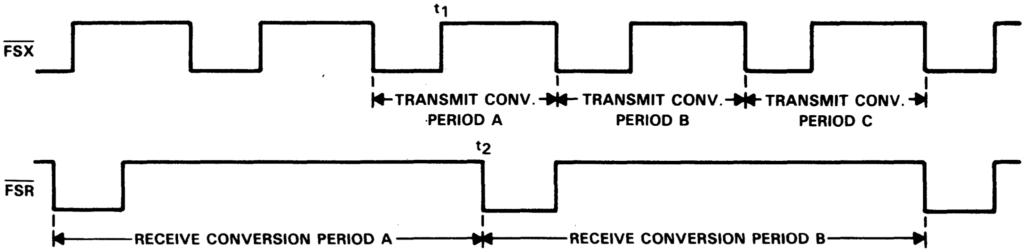
asynchronous operation — more than one receive frame sync occurring between two transmit frame syncs

When incrementally adjusting the conversion period via the A + A′ or A − A′ register options, a specific protocol is followed. The command to use the incremental conversion period adjust option is sent to the AIC during a $\overline{\text{FSX}}$ frame sync. The ongoing conversion period is then adjusted. However, either Receive Conversion Period A or B may be adjusted. For both transmit and receive conversion periods, the incremental conversion period adjustment is performed near the end of the conversion period. Therefore, if there is sufficient time between t1 and t2, the receive conversion period adjustment will be performed during Receive Conversion Period A. Otherwise, the adjustment will be performed during Receive Conversion Period B. The adjustment command only adjusts one transmit conversion period and one receive conversion period. To adjust another pair of transmit and receive conversion periods, another command must be issued during a subsequent $\overline{\text{FSX}}$ frame (see figure below).



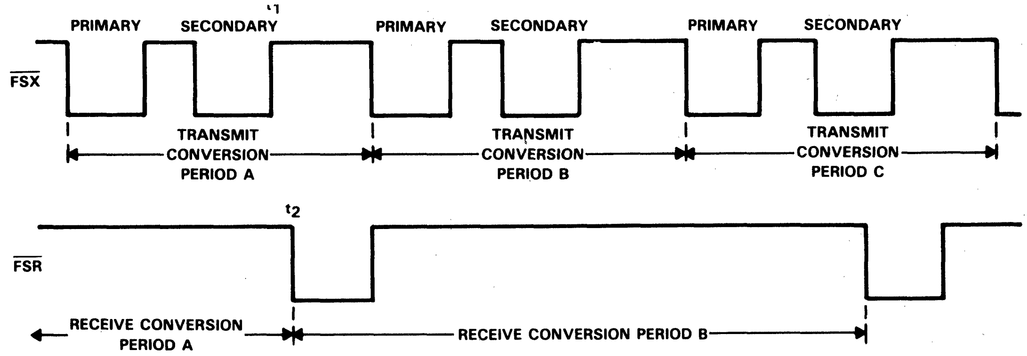
asynchronous operation — more than one transmit frame sync occurring between two receive frame syncs

When incrementally adjusting the conversion period via the A + A′ or A − A′ register options, a specific protocol is followed. For both transmit and receive conversion periods, the incremental conversion period adjustment is performed near the end of the conversion period. The command to use the incremental conversion period adjust options is sent to the AIC during a $\overline{\text{FSX}}$ frame sync. The ongoing transmit conversion period is then adjusted. However, three possibilities exist for the receive conversion period adjustment in the diagram as shown in the figure below. If the adjustment command is issued during Transmit Conversion Period A, Receive Conversion Period A will be adjusted if there is sufficient time between t1 and t2. Or, if there is not sufficient time between t1 and t2, Receive Conversion Period B will be adjusted. Or, the receive portion of an adjustment command may be ignored if the adjustment command is sent during a receive conversion period, which is already being or will be adjusted due to a prior adjustment command. For example, if adjustment commands are issued during Transmit Conversion Periods A, B, and C, the first two commands may cause Receive Conversion Periods A and B to be adjusted, while the third receive adjustment command is ignored. The third adjustment command is ignored since it was issued during Receive Conversion Period B, which already will be adjusted via the Transmit Conversion Period B adjustment command.



asynchronous operation — more than one set of primary and secondary DX serial communication occurring between two receive frame sync (see AIC DX Data Word Format section)

The TA, TA′, TB, and control register information that is transmitted in the secondary communications is always accepted and is applied during the ongoing transmit conversion period. If there is sufficient time between t1 and t2, the TA, RA′, and RB register information, which is sent during Transmit Conversion Period A, will be applied to Receive Conversion Period A. Otherwise, this information will be applied during Receive Conversion Period B. If RA, RA′, and RB register information has already been received and is being applied during an ongoing conversion period, any subsequent RA, RA′, or RB information that is received during this receive conversion period will be disregarded (see diagram below).


t1
PRIMARY
SECONDARY
PRIMARY
SECONDARY
PRIMARY
SECONDARY
FSX
TRANSMIT CONVERSION PERIOD A
TRANSMIT CONVERSION PERIOD B
TRANSMIT CONVERSION PERIOD C
t2
FSR
RECEIVE CONVERSION PERIOD A
RECEIVE CONVERSION PERIOD B

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC+} (see Note 1)	−0.3 V to 15 V
Supply voltage, V_{DD}	−0.3 V to 15 V
Output voltage, V_O	−0.3 V to 15 V
Input voltage, V_I	−0.3 V to 15 V
Digital ground voltage	−0.3 V to 15 V
Operating free-air temperature range: TLC32040I, TLC32041I, TLC32042I	−40°C to 85°C
TLC32040C, TLC32041C, TLC32042C	0°C to 70°C
Storage temperature range	−65°C to 150°C
Case temperature for 10 seconds: FN package	260°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds: N package	260°C

NOTE 1: Voltage values for maximum ratings are with respect to VCC−.

recommended operating conditions

PARAMETER		MIN	NOM	MAX	UNIT
Supply voltage, V_{CC+} (see Note 2)		4.75	5	5.25	V
Supply voltage, V_{CC-} (see Note 2)		−4.75	−5	−5.25	V
Digital supply voltage, V_{DD} (see Note 2)		4.75	5	5.25	V
Digital ground voltage with respect to ANLG GND, DGTL GND			0		V
Reference input voltage, $V_{ref(ext)}$ (see Note 2)		2		4	V
High-level input voltage, V_{IH}		2		$V_{DD}+0.3$	V
Low-level input voltage, V_{IL} (see Note 3)		−0.3		0.8	V
Load resistance at OUT+ and/or OUT−, R_L		300			Ω
Load capacitance at OUT+ and/or OUT−, C_L				100	pF
MSTR CLK frequency (see Note 4)		0.075	5	10.368	MHz
Analog input amplifier common mode input voltage (see Note 5)				±1.5	V
A/D or D/A conversion rate				19.2	kHz
Conversion rate		1		20	kHz
Operating free-air temperature, T_A	TLC32040I, TLC32041I, TLC32042I	−40		85	°C
	TLC32040C, TLC32041C, TLC32042C	0		70	

NOTES: 2. Voltages at analog inputs and outputs, REF, V_{CC+}, and V_{CC-}, are with respect to the ANLG GND terminal. Voltages at digital inputs and outputs and V_{DD} are with respect to the DGTL GND terminal.

3. The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels and temperature only.

4. The bandpass and low-pass switched-capacitor filter response specifications apply only when the switched-capacitor clock frequency is 288 kHz. For switched-capacitor filter clocks at frequencies other than 288 kHz, the filter response is shifted by the ratio of switched-capacitor filter clock frequency to 288 kHz.

5. This range applies when (IN+ − IN−) or (AUX IN+ − AUX IN−) equals ±6 V.

electrical characteristics over recommended operating free-air temperature range, $V_{CC+} = 5$ V, $V_{CC-} = -5$ V, $V_{DD} = 5$ V (unless otherwise noted)

total device, MSTR CLK frequency = 5.184 MHz, outputs not loaded

PARAMETER			TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage		$V_{DD} = 4.75$ V, $I_{OH} = -300$ μA	2.4			V
V_{OL}	Low-level output voltage		$V_{DD} = 4.75$ V, $I_{OL} = 2$ mA			0.4	V
I_{CC+}	Supply current from V_{CC+}	TLC3204_C				35	mA
		TLC3204_I				40	
I_{CC-}	Supply current from V_{CC-}	TLC3204_C				−35	mA
		TLC3204_I				−40	
I_{DD}	Supply current from V_{DD}		$f_{MSTR\ CLK} = 5.184$ MHz			7	mA
V_{ref}	Internal reference output voltage			3		3.3	V
αV_{ref}	Temperature coefficient of internal reference voltage				100		ppm/°C
r_o	Output resistance at REF				100		kΩ

receive amplifier input

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
	A/D converter offset error (filters bypassed)			25	65	mV
	A/D converter offset error (filters in)			25	65	mV
CMRR	Common-mode rejection ratio at IN+, IN−, or AUX IN+, AUX IN−	See Note 6		55		dB
r_I	Input resistance at IN+, IN− or AUX IN+, AUX IN−, REF			100		kΩ

transmit filter output

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OO}	Output offset voltage at OUT+ or OUT− (single-ended relative to ANLG GND)			15	75	mV
V_{OM}	Maximum peak output voltage swing across R_L at OUT+ or OUT− (single-ended)	$R_L \geq 300$ Ω, Offset voltage = 0	±3			V
V_{OM}	Maximum peak output voltage swing between OUT+ and OUT− (differential output)	$R_L \geq 600$ Ω	±6			V

†All typical values are at $T_A = 25$ °C.

NOTE 6: The test condition is a 0-dBm, 1-kHz input signal with an 8-kHz conversion rate.

electrical characteristics over recommended operating free-air temperature range, V_{CC+} = 5 V, V_{CC-} = −5 V, V_{DD} = 5 V (unless otherwise noted)

system distortion specifications, SCF clock frequency = 288 kHz

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
Attenuation of second harmonic of A/D input signal	single-ended	V_{in} = −0.1 dB to −24 dB referred to V_{ref}, See Note 7		70		dB
	differential		62	70		
Attenuation of third and higher harmonics of A/D input signal	single-ended	V_{in} = −0.1 dB to −24 dB referred to V_{ref}, See Note 7		65		dB
	differential		57	65		
Attenuation of second harmonic of D/A input signal	single-ended	V_{in} = −0 dB to −24 dB referred to V_{ref}, See Note 7		70		dB
	differential		62	70		
Attenuation of third and higher harmonics of D/A input signal	single-ended	V_{in} = −0 dB to −24 dB referred to V_{ref}, See Note 7		65		dB
	differential		57	65		

A/D channel signal-to-distortion ratio

PARAMETER	TEST CONDITIONS (see Note 7)	A_V = 1‡		A_V = 2‡		A_V = 4‡		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
A/D channel signal-to-distortion ratio	V_{in} = −6 dB to −0.1 dB	58		>58§		>58§		dB
	V_{in} = −12 dB to −6 dB	58		58		>58§		
	V_{in} = −18 dB to −12 dB	56		58		58		
	V_{in} = −24 dB to −18 dB	50		56		58		
	V_{in} = −30 dB to −24 dB	44		50		56		
	V_{in} = −36 dB to −30 dB	38		44		50		
	V_{in} = −42 dB to −36 dB	32		38		44		
	V_{in} = −48 dB to −42 dB	26		32		38		
	V_{in} = −54 dB to −48 dB	20		26		32		

D/A channel signal-to-distortion ratio

PARAMETER	TEST CONDITIONS (see Note 7)	MIN	MAX	UNIT
D/A channel signal-to-distortion ratio	V_{in} = −6 dB to −0.1 dB	58		dB
	V_{in} = −12 dB to −6 dB	58		
	V_{in} = −18 dB to −12 dB	56		
	V_{in} = −24 dB to −18 dB	50		
	V_{in} = −30 dB to −24 dB	44		
	V_{in} = −36 dB to −30 dB	38		
	V_{in} = −42 dB to −36 dB	32		
	V_{in} = −48 dB to −42 dB	26		
	V_{in} = −54 dB to −48 dB	20		

gain and dynamic range

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
Absolute transmit gain tracking error while transmitting into 600 Ω	−48 dB to 0 dB signal range, See Note 8		±0.05	±0.15	dB
Absolute receive gain tracking error	−48 dB to 0 dB signal range, See Note 8		±0.05	±0.15	dB

† All typical values are at T_A = 25°C.
‡ A_V is the programmable gain of the input amplifier.
§ A value > 58 is overrange and signal clipping occurs.

NOTES: 7. The test condition is a 1-kHz input signal with an 8-kHz conversion rate (0 dB relative to V_{ref}). The load impedance for the DAC is 600 Ω.
8. Gain tracking is relative to the absolute gain at 1 kHz and 0 dB (0 db relative to V_{ref}).

power supply rejection and crosstalk attenuation

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{CC+} or V_{CC-} supply voltage rejection ratio, receive channel	f = 0 to 30 kHz	Idle channel, supply signal at 200 mV p-p measured at DR (ADC output)		30		dB
	f = 30 kHz to 50 kHz			45		
V_{CC+} or V_{CC-} supply voltage rejection ratio, transmit channel (single-ended)	f = 0 to 30 kHz	Idle channel, supply signal at 200 mV p-p measured at OUT+		30		dB
	f = 30 kHz to 50 kHz			45		
Crosstalk attenuation, transmit-to-receive (single-ended)				80		dB

delay distortion, SCF clock frequency = 288 kHz ±2%, input (IN+ − IN−) is ±3-V sinewave

Please refer to filter response graphs for delay distortion specifications.

TLC32040 and TLC32041 bandpass filter transfer function (see curves), SCF clock frequency = 288 kHz ±2%, input (IN+ − IN−) is a ±3-V sinewave (see Note 9)

PARAMETER	TEST CONDITIONS		MIN	MAX	UNIT
Filter Gain (see Note 10)	Input signal reference is 0 dB	f = 100 Hz		−42	dB
		f = 170 Hz		−25	
		300 Hz ≤ f ≤ 3.4 kHz	−0.5	0.5	
		f = 4 kHz		−16	
		f ≥ 4.6 kHz		−58	

TLC32042 bandpass filter transfer function (see curves), SCF clock frequency = 288 kHz ±2%, input (IN+ − IN−) is a ±3-V sinewave (see Note 9)

PARAMETER	TEST CONDITIONS		MIN	MAX	UNIT
Filter Gain (see Note 10)	Input signal reference is 0 dB	f = 100 Hz		−27	dB
		f = 170 Hz		−2	
		300 Hz ≤ f ≤ 3.4 kHz	−0.5	0.5	
		f = 4 kHz		−16	
		f ≥ 4.6 kHz		−58	

low-pass filter transfer function, SCF clock frequency = 288 kHz ±2% (see Note 9)

.PARAMETER	TEST CONDITIONS		MIN	MAX	UNIT
Filter Gain (see Note 10)	Output signal reference is 0 dB	f ≤ 3.4 kHz	−0.5	0.5	dB
		f = 3.6 kHz		−4	
		f = 4 kHz		−30	
		f ≥ 4.4 kHz		−58	

serial port

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage	I_{OH} = −300 μA	2.4			V
V_{OL}	Low-level output voltage	I_{OL} = 2 mA			0.4	V
I_I	Input current				±10	μA
C_I	Input capacitance			15		pF
C_O	Output capacitance			15		pF

† All typical values are at T_A = 25°C.

NOTES: 9. The above filter specifications are for a switched-capacitor filter clock range of 288 kHz ±2%. For switched-capacitor filter clocks at frequencies other than 288 kHz ±2%, the filter response is shifted by the ratio of switched-capacitor filter clock frequency to 288 kHz.

10. The filter gain outside of the passband is measured with respect to the gain at 1 kHz. The filter gain within the passband is measured with respect to the average gain within the passband. The passbands are 300 to 3400 Hz and 0 to 3400 Hz for the bandpass and lowpass filters respectively.

operating characteristics over recommended operating free-air temperature range, V_{CC+} = 5 V, V_{CC-} = −5 V, V_{DD} = 5 V

noise (measurement includes low-pass and bandpass switched-capacitor filters)

PARAMETER		TEST CONDITIONS	TYP†	MAX	UNIT
Transmit noise	single-ended	DX input = 00000000000000, constant input code	200		µV rms
	differential		300	500	µV rms
			20		dBrnc0
Receive noise (see Note 11)		Inputs grounded, gain = 1	300	475	µV rms
			20		dBrnc0

timing requirements

serial port recommended input signals

PARAMETER		MIN	MAX	UNIT
$t_{c(MCLK)}$	Master clock cycle time	95		ns
$t_{r(MCLK)}$	Master clock rise time		10	ns
$t_{f(MCLK)}$	Master clock fall time		10	ns
	Master clock duty cycle	42%	58%	
	$\overline{\text{RESET}}$ pulse duration (see Note 12)	800		ns
$t_{su(DX)}$	DX setup time before SCLK↓	20		ns
$t_{h(DX)}$	DX hold time after SCLK↓	$t_{c(SCLK)}/4$		ns

† All typical values are at T_A = 25°C.

NOTES: 11. This noise is referred to the input with a buffer gain of one. If the buffer gain is two or four, the noise figure will be correspondingly reduced. The noise is computed by statistically evaluating the digital output of the A/D converter.

12. $\overline{\text{RESET}}$ pulse duration is the amount of time that the reset pin is held below 0.8 V after the power supplies have reached their recommended values.

operating characteristics over recommended operating free-air temperature range, V_{CC+} = 5 V, V_{CC-} = −5 V, V_{DD} = 5 V (continued)

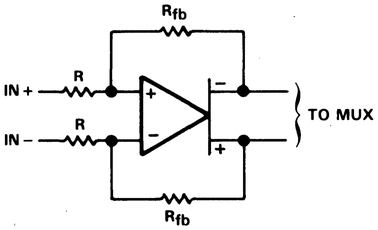
serial port — AIC output signals

PARAMETER		MIN	MAX	UNIT
$t_{c(SCLK)}$	Shift clock (SCLK) cycle time	380		ns
$t_{f(SCLK)}$	Shift clock (SCLK) fall time		50	ns
$t_{r(SCLK)}$	Shift clock (SCLK) rise time		50	ns
	Shift clock (SCLK) duty cycle	45	55	%
$t_{d(CH\text{-}FL)}$	Delay from SCLK↑ to $\overline{\mathrm{FSR/FSX}}$↓		90	ns
$t_{d(CH\text{-}FH)}$	Delay from SCLK↑ to $\overline{\mathrm{FSR/FSX}}$↑		90	ns
$t_{d(CH\text{-}DR)}$	DR valid after SCLK↑		90	ns
$t_{dw(CH\text{-}EL)}$	Delay from SCLK↑ to $\overline{\mathrm{EODX/EODR}}$↓ in word mode		90	ns
$t_{dw(CH\text{-}EH)}$	Delay from SCLK↑ to $\overline{\mathrm{EODX/EODR}}$↑ in word mode		90	ns
$t_{f(EODX)}$	$\overline{\mathrm{EODX}}$ fall time		15	ns
$t_{f(EODR)}$	$\overline{\mathrm{EODR}}$ fall time		15	ns
$t_{db(CH\text{-}EL)}$	Delay from SCLK↑ to $\overline{\mathrm{EODX/EODR}}$↓ in byte mode		100	ns
$t_{db(CH\text{-}EH)}$	Delay from SCLK↑ to $\overline{\mathrm{EODX/EODR}}$↑ in byte mode		100	ns

TABLE 2. GAIN CONTROL TABLE (ANALOG INPUT SIGNAL REQUIRED FOR FULL-SCALE A/D CONVERSION)

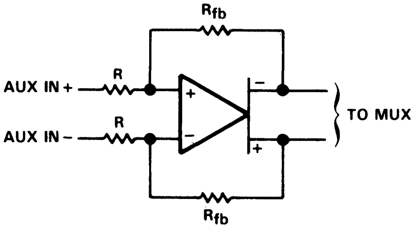
INPUT CONFIGURATIONS	CONTROL REGISTER BITS		ANALOG INPUT[†]	A/D CONVERSION RESULT
	d6	d7		
Differential configuration	1	1	±6 V	full-scale
Analog input = IN+ − IN−	0	0		
= AUX IN+ − AUX IN−	1	0	±3 V	full-scale
	0	1	±1.5 V	full-scale
Single-ended configuration	1	1	±3 V	half-scale
Analog input = IN+ − ANLG GND	0	0		
= AUX IN+ − ANLG GND	1	0	±3 V	full-scale
	0	1	±1.5 V	full-scale

[†] In this example, V_{ref} is assumed to be 3 V. In order to minimize distortion, it is recommended that the analog input not exceed 0.1 dB below full scale.



R_{fb} = R for d6 = 1, d7 = 1
d6 = 0, d7 = 0
R_{fb} = 2R for d6 = 1, d7 = 0
R_{fb} = 4R for d6 = 0, d7 = 1

FIGURE 1. IN+ AND IN− GAIN CONTROL CIRCUITRY



R_{fb} = R for d6 = 1, d7 = 1
d6 = 0, d7 = 0
R_{fb} = 2R for d6 = 1, d7 = 0
R_{fb} = 4R for d6 = 0, d7 = 1

FIGURE 2. AUX IN+ AND AUX IN− GAIN CONTROL CIRCUITRY

sin x/x correction section

The AIC does not have sin x/x correction circuitry after the digital-to-analog converter. Sin x/x correction can be accomplished easily and efficiently in digital signal processor (DSP) software. Excellent correction accuracy can be achieved to a band edge of 3000 Hz by using a first-order digital correction filter. The results, which are shown below, are typical of the numerical correction accuracy that can be achieved for sample rates of interest. The filter requires only seven instruction cycles per sample on the TMS320 DSPs. With a 200-ns instruction cycle, nine instructions per sample represents an overhead factor of 1.4% and 1.7% for sampling rates of 8000 Hz and 9600 Hz, respectively. This correction will add a slight amount of group delay at the upper edge of the 300—3000-Hz band.

sin x/x roll-off for a zero-order hold function

The sin x/x roll-off for the AIC DAC zero-order hold function at a band-edge frequency of 3000 Hz for the various sampling rates is shown in the table below.

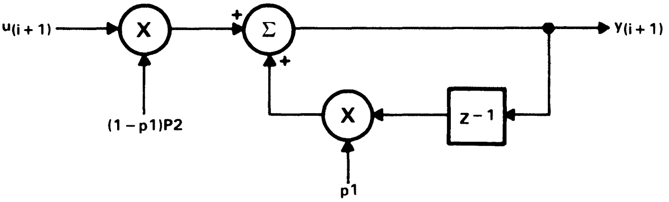
TABLE 3. sin x/x ROLL-OFF

f_s (Hz)	$20 \log \frac{\sin \pi f/f_s}{\pi f/f_s}$ (f = 3000 Hz) (dB)
7200	−2.64
8000	−2.11
9600	−1.44
14400	−0.63
19200	−0.35

Note that the actual AIC sin x/x roll-off will be slightly less than the above figures, because the AIC has less tnan a 100-% duty cycle hold interval.

correction filter

To compensate for the sin x/x roll-off of the AIC, a first-order correction filter shown below, is recommended.



The difference equation for this correction filter is:

$$y_{i+1} = p2(1 - p1)\ (u_{i+1}) + p1\ y_i$$

where the constant p1 determines the pole locations.

The resulting squared magnitude transfer function is:

$$|H(f)|^2 = \frac{p2^2\ (1 - p1)^2}{1\ -\ 2p1\ \cos(2\ \pi\ f/f_S)\ +\ p1^2}$$

correction results

Table 4 below shows the optimum p values and the corresponding correction results for 8000-Hz and 9600-Hz sampling rates.

TABLE 4

f (Hz)	ERROR (dB)
	f_s = 8000 Hz
	p1 = −0.14813
	p2 = 0.9888
		ERROR (dB)
		f_s = 9600 Hz
		p1 = −0.1307
		p2 = 0.9951

300	−0.099	−0.043
600	−0.089	−0.043
900	−0.054	0
1200	−0.002	0
1500	0.041	0
1800	0.079	0.043
2100	0.100	0.043
2400	0.091	0.043
2700	−0.043	0
3000	−0.102	−0.043

TMS320 software requirements

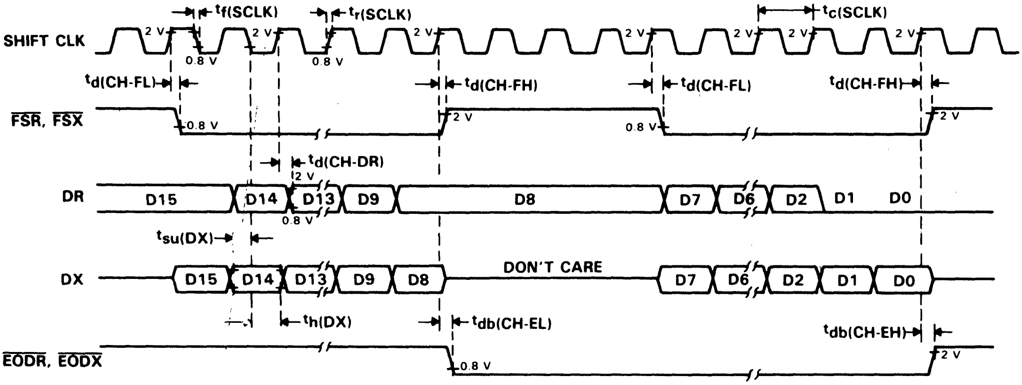
The digital correction filter equation can be written in state variable form as follows:

$$Y = k1Y + k2U$$

where k1 equals p1 (from the preceding page), k2 equals (1 − p1)p2 (from the preceding page), Y is the filter state, and U is the next I/O sample. The coefficients k1 and k2 must be represented as 16-bit integers. The SACH instruction (with the proper shift) will yield the correct result. With the assumption that the TMS320 processor page pointer and memory configuration are properly initialized, the equation can be executed in seven instructions or seven cycles with the following program:

```
ZAC
LT K2
MPY U
LTA K1
MPY Y
APAC
SACH (dma), (shift)
```

byte-mode timing



word-mode timing

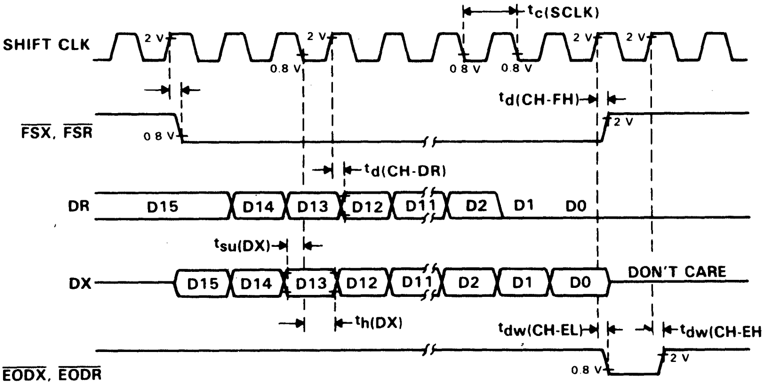


FIGURE 3. SERIAL PORT TIMING

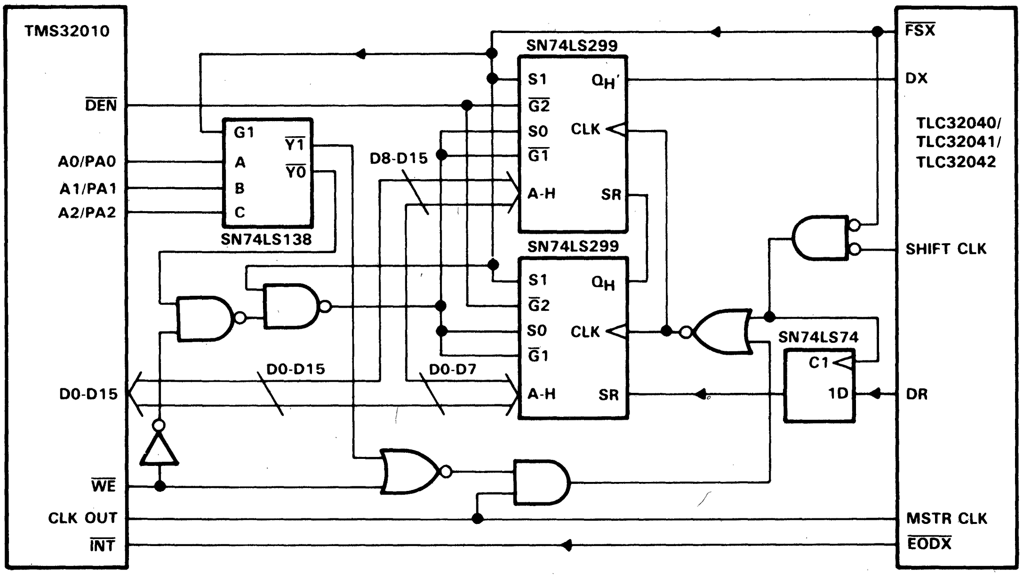


FIGURE 4. TMS32010-TLC32040/TLC32041/TLC32042 INTERFACE CIRCUIT

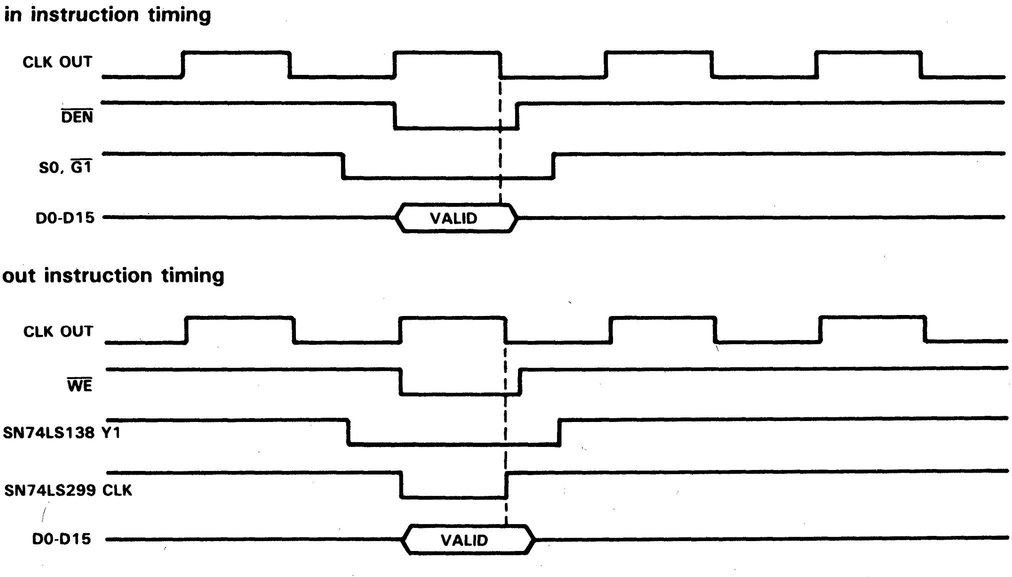
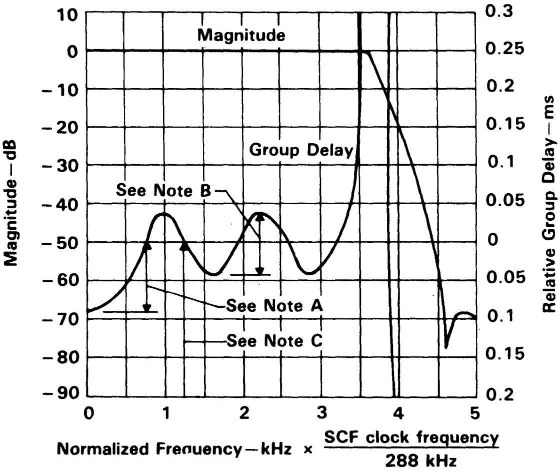


FIGURE 5. TMS32010-TLC32040/TLC32041/TLC32042 INTERFACE TIMING

TYPICAL CHARACTERISTICS

AIC TRANSMIT CHANNEL FILTER

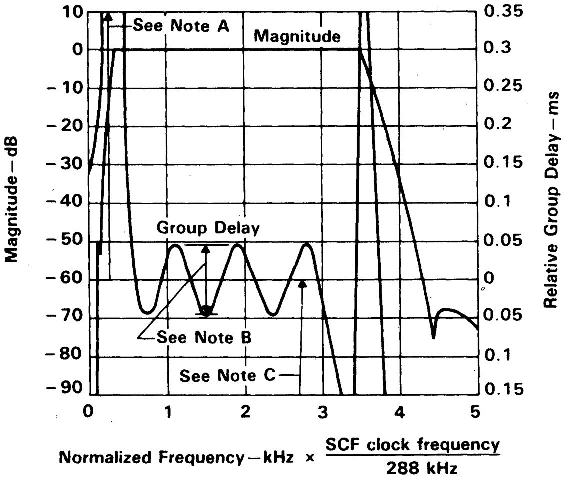


NOTES: A. Maximum relative delay (0 Hz to 600 Hz) = 125 μs.
B. Maximum relative delay (600 Hz to 3000 Hz) = ±50 μs.
C. Absolute delay (600 Hz to 3000 Hz) = 700 μs.
D. Test conditions are V_{CC+}, V_{CC-}, and V_{DD} within recommended operating conditions, SCF clock f = 288 kHz ±2%, input = ±3-V sinewave, and T_A = 25°C.

FIGURE 6

TYPICAL CHARACTERISTICS

TLC32040 and TLC32041
RECEIVE CHANNEL FILTER



NOTES: A. Maximum relative delay (200 Hz to 600 Hz) = 3350 μs.

B. Maximum relative delay (600 Hz to 3000 Hz) = ±50 μs.

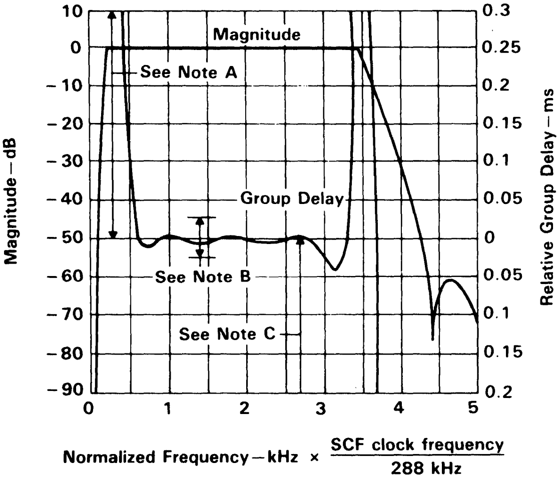
C. Absolute delay (600 Hz to 3000 Hz) = 1230 μs

D. Test conditions are V_{CC+}, V_{CC-}, and V_{DD} within recommended operating conditions, SCF clock f = 288 kHz ±2%, input = ±3-V sinewave, and T_A = 25°C.

FIGURE 7

TYPICAL CHARACTERISTICS

TLC32042
RECEIVE CHANNEL FILTER



NOTES: A. Maximum relative delay (200 Hz to 600 Hz) = 3350 μs.
B. Maximum relative delay (600 Hz to 3000 Hz) = ±50 μs.
C. Absolute delay (600 Hz to 3000 Hz) = 1080 μs.
D. Test conditions are V_{CC+}, V_{CC-}, and V_{DD} within recommended operating conditions, SCF clock f = 288 kHz ±2%, input = ±3-V sinewave, and T_A = 25°C.

FIGURE 8

TYPICAL CHARACTERISTICS

A/D SIGNAL-TO-DISTORTION RATIO
vs
INPUT SIGNAL

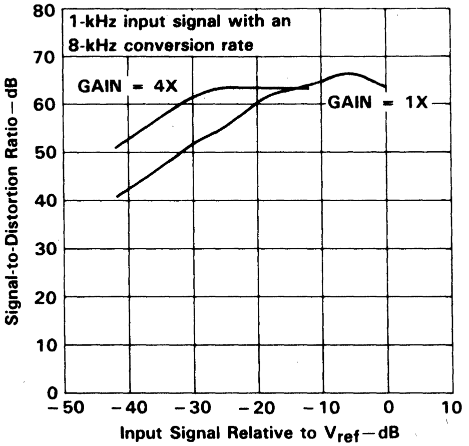


FIGURE 9

A/D GAIN TRACKING
(GAIN RELATIVE TO GAIN
AT 0 dB INPUT SIGNAL)

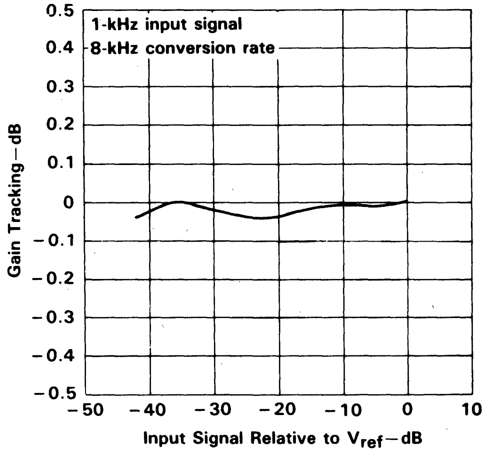


FIGURE 10

D/A CONVERTER SIGNAL-TO-DISTORTION RATIO
vs
INPUT SIGNAL

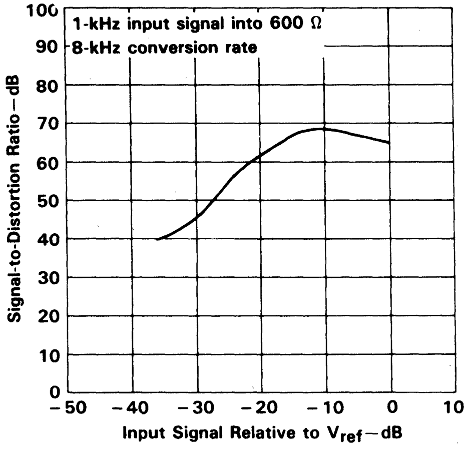


FIGURE 11

D/A GAIN TRACKING
vs
(GAIN RELATIVE TO GAIN
AT 0 dB INPUT SIGNAL)

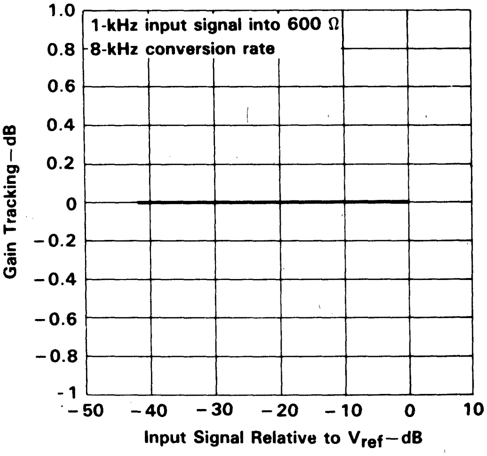


FIGURE 12

NOTE: Test conditions are V_{CC+}, V_{CC-}, and V_{DD} within recommenaed operating conditions set clock f = 288 kHz ±2%, and T_A = 25°C.

TYPICAL CHARACTERISTICS

ATTENUATION OF SECOND HARMONIC OF A/D INPUT
vs
INPUT SIGNAL

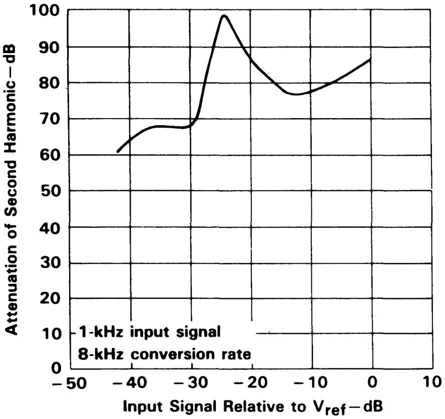


FIGURE 13

ATTENUATION OF THIRD HARMONIC OF A/D INPUT
vs
INPUT SIGNAL

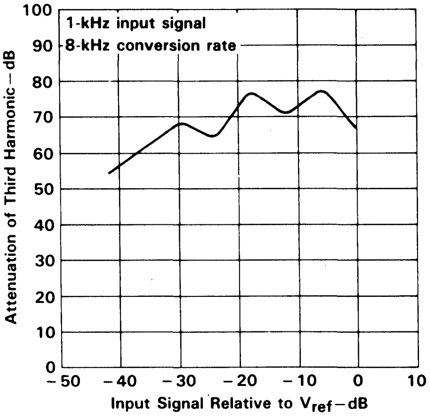


FIGURE 14

ATTENUATION OF SECOND HARMONIC OF D/A INPUT
vs
INPUT SIGNAL

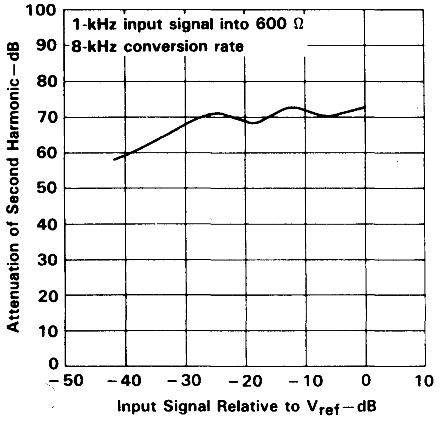


FIGURE 15

ATTENUATION OF THIRD HARMONIC OF D/A INPUT
vs
INPUT SIGNAL

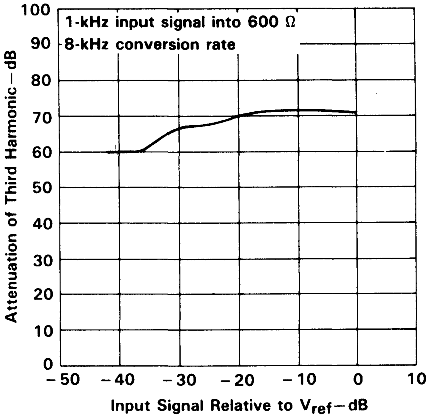


FIGURE 16

NOTE: Test conditions are V_{CC+}, V_{CC-}, and V_{DD} within recommended operating conditions set clock f = 288 kHz ± 2%, and T_A = 25 °C.

TYPICAL APPLICATION INFORMATION

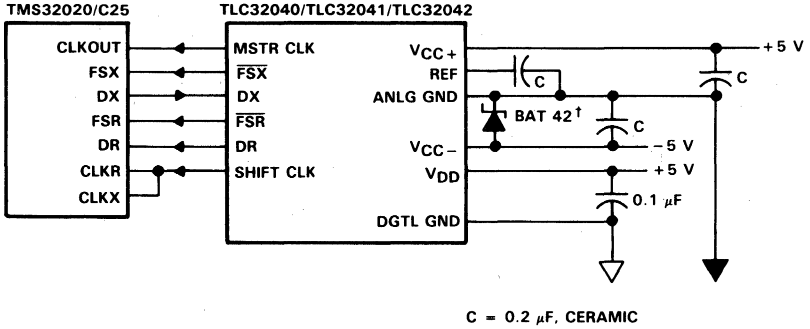


FIGURE 17. AIC INTERFACE TO THE TMS32020/C25 SHOWING DECOUPLING CAPACITORS AND SCHOTTKY DIODE†

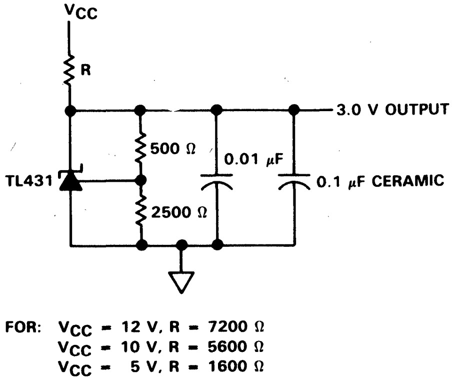

V_{CC}
R
3.0 V OUTPUT
500 Ω
0.01 μF
TL431
0.1 μF CERAMIC
2500 Ω
FOR: V_{CC} = 12 V, R = 7200 Ω
V_{CC} = 10 V, R = 5600 Ω
V_{CC} = 5 V, R = 1600 Ω

FIGURE 18. EXTERNAL REFERENCE CIRCUIT FOR TLC32041

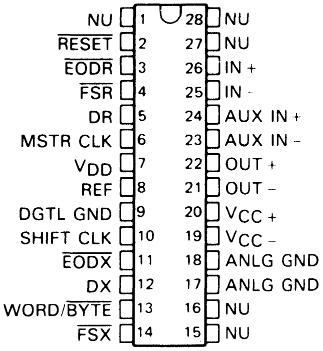
†Thomson Semiconductors

TLC32044I, TLC32044C
VOICE-BAND ANALOG INTERFACE CIRCUITS

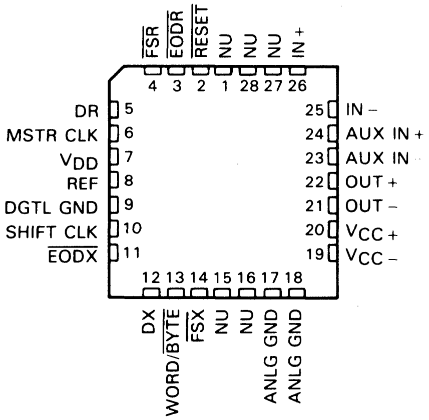
D3098, MARCH 1988–REVISED DECEMBER 1988

- **Advanced LinCMOS™ Silicon-Gate Process Technology**
- **14-Bit Dynamic Range ADC and DAC**
- **16-Bit Dynamic Range Input with Programmable Gain**
- **Variable ADC and DAC Sampling Rate Up to 19,200 Samples per Second**
- **Switched-Capacitor Antialiasing Input Filter and Output-Reconstruction Filter**
- **Serial Port for Direct Interface to TMS320C17, TMS32020, TMS320C25, and TMS320C30 Digital Processors**
- **Synchronous or Asynchronous ADC and DAC Conversion Rates with Programmable Incremental ADC and DAC Conversion Timing Adjustments**
- **Serial Port Interface to SN74299 Serial-to-Parallel Shift Register for Parallel Interface to TMS32010, TMS320C15, or Other Digital Processors**
- **Internal Reference for Normal Operation and External Purposes, or Can Be Overridden by External Reference**

N PACKAGE
(TOP VIEW)



FN PACKAGE
(TOP VIEW)



NU—Nonusable; no external connection should be made to these pins. See Table 2.

description

The TLC32044 is a complete analog-to-digital and digital-to-analog input/output system on a single monolithic CMOS chip. This device integrates a bandpass switched-capacitor antialiasing input filter, a 14-bit-resolution A/D converter, four microprocessor-compatible serial port modes, a 14-bit-resolution D/A converter, and a low-pass switched-capacitor output-reconstruction filter. The device offers numerous combinations of Master Clock input frequencies and conversion/sampling rates, which can be changed via digital processor control.

Typical applications for this IC include speech encryption for digital transmission, speech recognition/storage systems, speech synthesis, modems (7.2-, 8-, 9.6-, 14.4-, and 19.2-kHz sampling rate), analog interface for digital signal processors (DSPs), industrial process control, biomedical instrumentation, acoustical signal processing, spectral analysis, data acquisition, and instrumentation recorders. Four serial modes, which allow direct interface to the TMS320C17, TMS32020, TMS320C25, and TMS320C30 digital signal

Advanced LinCMOS™ is a trademark of Texas Instruments Incorporated

description (continued)

processors, are provided. Also, when the transmit and receive sections of the Analog Interface Circuit (AIC) are operating synchronously, it will interface to two SN74299 serial-to-parallel shift registers. These serial-to-parallel shift registers can then interface in parallel to the TMS32010, TMS320C15, other digital signal processors, or external FIFO circuitry. Output data pulses are emitted to inform the processor that data transmission is complete or to allow the DSP to differentiate between two transmitted bytes. A flexible control scheme is provided so that the functions of the IC can be selected and adjusted coincidentally with signal processing via software control.

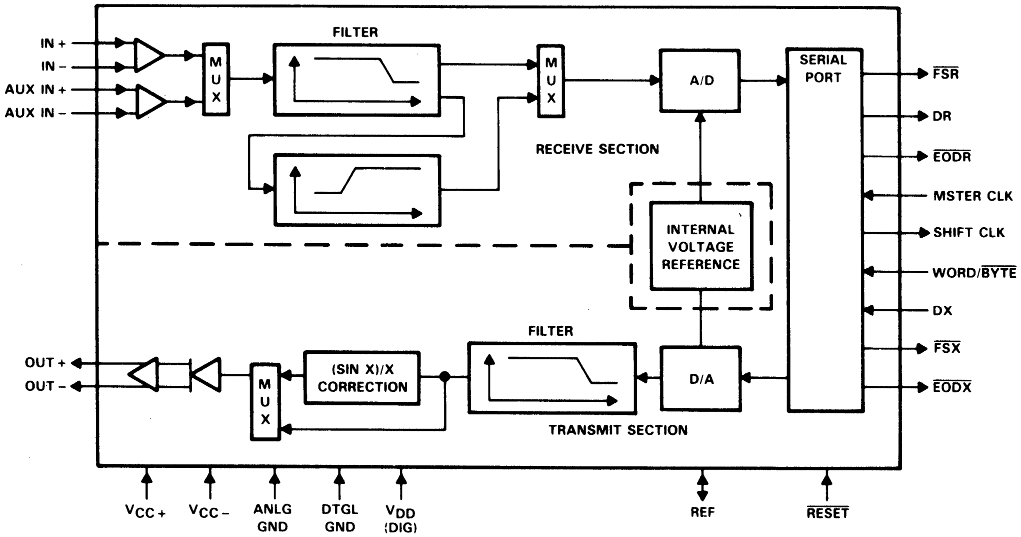
The antialiasing input filter comprises eighth-order and fourth-order CC-type (Chebyshev/elliptic transitional) low-pass and high-pass filters, respectively. The input filter is implemented in switched-capacitor technology and is preceded by a continuous time filter to eliminate any possibility of aliasing caused by sampled data filtering. When only low-pass filtering is desired, the high-pass filter can be switched out of the signal path. A selectable, auxiliary, differential analog input is provided for applications where more than one analog input is required.

The A/D and D/A architectures ensure no missing codes and monotonic operation. An internal voltage reference is provided to ease the design task and to provide complete control over the performance of the IC. The internal voltage reference is brought out to a pin and is available to the designer. Separate analog and digital voltage supplies and grounds are provided to minimize noise and ensure a wide dynamic range. Also, the analog circuit path contains only differential circuitry to keep noise to an absolute minimum. The only exception is the DAC sample-and-hold, which utilizes pseudo-differential circuitry.

The output-reconstruction filter is an eighth-order CC-type (Chebyshev/elliptic transitional low-pass filter) followed by a second-order (sin x)/x correction filter) and is implemented in switched-capacitor technology. This filter is followed by a continuous-time filter to eliminate images of the digitally encoded signal. The on-board (sin x)/x correction filter can be switched out of the signal path using digital signal processor control, if desired.

The TLC32044I is characterized for operation from −40 °C to 85 °C, and the TLC32044C is characterized for operation from 0 °C to 70 °C.

functional block diagram



PRINCIPLES OF OPERATION

analog input

Two sets of analog inputs are provided. Normally, the IN + and IN − input set is used; however, the auxiliary input set, AUX IN + and AUX IN −, can be used if a second input is required. Each input set can be operated in either differential or single-ended modes, since sufficient common-mode range and rejection are provided. The gain for the IN +, IN −, AUX IN +, and AUX IN − inputs can be programmed to be either 1, 2, or 4 (see Table 2). Either input circuit can be selected via software control. It is important to note that a wide dynamic range is assured by the differential internal analog archltecture and by the separate analog and digital voltage supplies and grounds.

A/D bandpass filter, A/D bandpass filter clocking, and A/D conversion timing

The A/D high-pass filter can be selected or bypassed via software control. The frequency response of this filter is presented in the following pages. This response results when the switched-capacitor filter clock frequency is 288 kHz and the A/D sample rate is 8 kHz. Several possible options can be used to attain a 288-kHz switched-capacitor filter clock. When the filter clock frequency is not 288 kHz, the low-pass filter transfer function is frequency-scaled by the ratio of the actual clock frequency to 288 kHz. The ripple bandwidth and 3-dB low-frequency roll-off points of the high-pass section are 150 and 100 Hz, respectively. However, the high-pass section low-frequency roll-off is frequency-scaled by the ratio of the A/D sample rate to 8 kHz.

The Internal Timing Configuration and AIC DX Data Word Format sections of this data sheet indicate the many options for attaining a 288-kHz bandpass switched-capacitor filter clock. These sections indicate that the RX Counter A can be programmed to give a 288-kHz bandpass switched-capacitor filter clock for several Master Clock input frequencies.

PRINCIPLES OF OPERATION (continued)

The A/D conversion rate is then attained by frequency-dividing the 288-kHz bandpass switched-capacitor filter clock with the RX Counter B. Thus, unwanted aliasing is prevented because the A/D conversion rate is an integral submultiple of the bandpass switched-capacitor filter sampling rate, and the two rates are synchronously locked.

A/D converter performance specifications

Fundamental performance specifications for the A/D converter circuitry are presented in the A/D converter operating characteristics section of this data sheet. The realization of the A/D converter circuitry with switched-capacitor techniques provides an inherent sample-and-hold.

analog output

The analog output circuitry is an analog output power amplifier. Both noninverting and inverting amplifier outputs are brought out of the IC. This amplifier can drive transformer hybrids or low-impedance loads directly in either a differential or single-ended configuration.

D/A low-pass filter, D/A low-pass filter clocking, and D/A conversion timing

The frequency response of this filter is presented in the following pages. This response results when the low-pass switched-capacitor filter clock frequency is 288 kHz. Like the A/D filter, the transfer function of this filter is frequency-scaled when the clock frequency is not 288 kHz. A continuous-time filter is provided on the output of the (sin x)/x filter to eliminate the periodic sample data signal information, which occurs at multiples of the 288-kHz switched-capacitor filter clock. The continuous time filter also greatly attenuates any switched-capacitor clock feedthrough.

The D/A conversion rate is attained by frequency-dividing the 288-kHz switched-capacitor filter clock with TX Counter B. Thus, unwanted aliasing is prevented because the D/A conversion rate is an integral submultiple of the switched-capacitor low-pass filter sampling rate, and the two rates are synchronously locked.

asynchronous versus synchronous operation

If the transmit section of the AIC (low-pass filter and DAC) and receive section (bandpass filter and ADC) are operated asynchronously, the low-pass and bandpass filter clocks are independently generated from the Master Clock signal. Also, the D/A and A/D conversion rates are independently determined. If the transmit and receive sections are operated synchronously, the low-pass filter clock drives both low-pass and bandpass filters. In synchronous operation, the A/D conversion timing is derived from, and is equal to, the D/A conversion timing. (See description of the WORD/$\overline{\text{BYTE}}$ pin in the Pin Functional Description Section.)

D/A converter performance specifications

Fundamental performance specifications for the D/A converter circuitry are presented in the D/A converter operating characteristics section of the data sheet. The D/A converter has a sample-and-hold that is realized with a switched-capacitor ladder.

system frequency response correction

(Sin x)/x correction for the D/A converter's zero-order sample-and-hold output can be provided by an on-board second-order (sin x)/x correction filter. This (sin x)/x correction filter can be inserted into or deleted from the signal path by digital signal processor control. When inserted, the (sin x)/x correction filter follows the switched-capacitor low-pass filter. When the TB register (see Internal Timing Configuration section) equals 36, the correction results of Figures 11 and 12 will be obtained.

PRINCIPLES OF OPERATION (continued)

(Sin x)/x correction can also be accomplished by deleting the on-board second-order correction filter and performing the (sin x)/x correction in digital signal processor software. The system frequency response can be corrected via DSP software to ±0.1 dB accuracy to a band-edge of 3000 Hz for all sampling rates. This correction is accomplished with a first-order digital correction filter, which requires only seven TMS320 instruction cycles. With a 200-ns instruction cycle, seven instructions represent an overhead factor of only 1.1% and 1.3% for sampling rates of 8 and 9.6 kHz, respectively (see the (sin x)/x Correction Section for more details).

serial port

The serial port has four possible modes that are described in detail in the Functional Pin Description Section. These modes are briefly described below and in the Functional Description for Pin 13, WORD/$\overline{\text{BYTE}}$.

1. The transmit and receive sections are operated asynchronously, and the serial port interfaces directly with the TMS320C17.
2. The transmit and receive sections are operated asynchronously, and the serial port interfaces directly with the TMS32020, TMS320C25, and the TMS320C30.
3. The transmit and receive sections are operated synchronously, and the serial port interfaces directly with the TMS320C17.
4. The transmit and receive sections are operated synchronously, and the serial port interfaces directly with the TMS32020, TMS320C25, TMS320C30, or two SN74299 serial-to-parallel shift registers, which can then interface in parallel to the TMS32010, TMS320C15, to any other digital signat processor, or to external FIFO circuitry.

operation of TLC32044 with internal voltage reference

The internal reference of the TLC32044 eliminates the need for an external voltage reference and provides overall circuit cost reduction. Thus, the internal reference eases the design task and provides complete control over the performance of the IC. The internal reference is brought out to a pin and is available to the designer. To keep the amount of noise on the reference signal to a minimum, an external capacitor mav be connected between REF and ANLG GND.

operation of TLC32044 with external voltage reference

The REF pin may be driven from an external reference circuit if so desired. This external circuit must be capable of supplying 250 μA and must be adequately protected from noise such as crosstalk from the analog input.

reset

A reset function is provided to initiate serial communications between the AIC and DSP and to allow fast, cost-effective testing during manufacturing. The reset function will initialize all AIC registers, including the control register. After a negative-going pulse on the $\overline{\text{RESET}}$ pin, the AIC will be initialized. This initialization allows normal serial port communications activity to occur between AIC and DSP (see AIC DX Data Word Format section).

PRINCIPLES OF OPERATION (continued)

loopback

This feature allows the user to test the circuit remotely. In loopback, the OUT + and OUT − pins are internally connected to the IN + and IN − pins. Thus, the DAC bits (d15 to d2), which are transmitted to the DX pin, can be compared with the ADC bits (d15 to d2), which are received from the DR pin. An ideal comparison would be that the bits on the DR pin equal the bits on the DX pin. However, in practice there will be some difference in these bits due to the ADC and DAC output offsets.

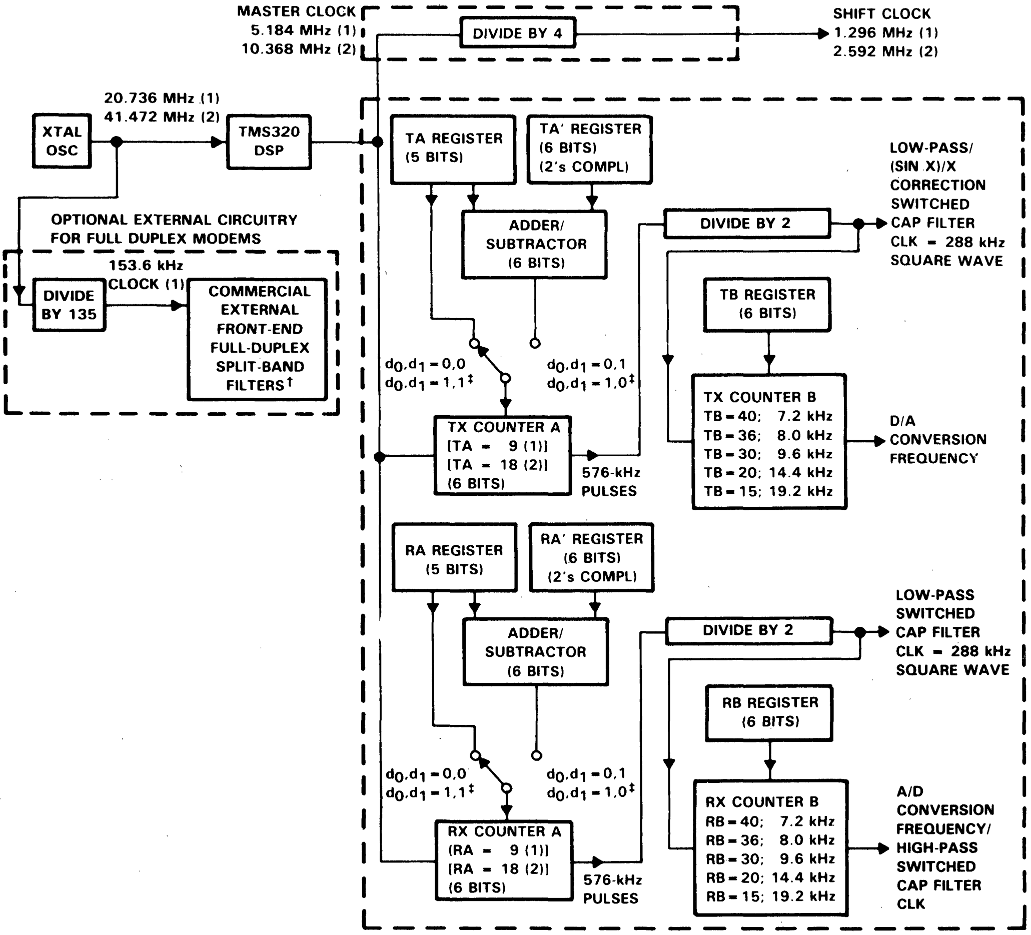
The loopback feature is implemented with digital signal processor control by transmitting the appropriate serial port bit to the control register (see AIC Data Word Format section).

PIN		I/O	DESCRIPTION
NAME	NO.		
ANLG GND	17,18		Analog ground return for all internal analog circuits. Not internally connected to DGTL GND.
AUX IN +	24	I	Noninverting auxiliary analog input stage. This input can be switched into the bandpass filter and A/D converter path via software control. If the appropriate bit in the Control register is a 1, the auxiliary inputs will replace the IN + and IN − inputs. If the bit is a 0, the IN + and IN − inputs will be used (see the AIC DX Data Word Format section).
AUX IN −	23	I	Inverting auxiliary analog input (see the above AUX IN + pin description).
DGTL GND	9		Digital ground for all internal logic circuits. Not internally connected to ANLG GND.
DR	5	O	This pin is used to transmit the ADC output bits from the AIC to the TMS320 serial port. This transmission of bits from the AIC to the TMS320 serial port is synchronized with the SHIFT CLK signal.
DX	12	I	This pin is used to receive the DAC input bits and timing and control information from the TMS320. This serial transmission from the TMS320 serial port to the AIC is synchronized with the SHIFT CLK signal.
$\overline{\text{EODR}}$	3	O	(See the WORD/$\overline{\text{BYTE}}$ pin description and the Serial Port Timing Diagram.) During the word-mode timing, this signal is a low-going pulse that occurs immediately after the 16 bits of A/D information have been transmitted from the AIC to the TMS320 serial port. This signal can be used to interrupt a microprocessor upon completion of serial communications. Also, this signal can be used to strobe and enable external serial-to-parallel shift registers, latches, or external FIFO RAM, and to facilitate parallel data bus communications between the AIC and the serial-to-parallel shift registers. During the byte-mode timing, this signal goes low after the first byte has been transmitted from the AIC to the TMS320 serial port and is kept low until the second byte has been transmitted. The TMS320C17 can use this low-going signal to differentiate between the two bytes as to which is first and which is second.

PIN		I/O	DESCRIPTION
NAME	NO.		
$\overline{\mathrm{EODX}}$	11	O	(See the WORD/$\overline{\mathrm{BYTE}}$ pin description and the Serial Port Timing Diagram.) During the word-mode timing, this signal is a low-going pulse that occurs immediately after the 16 bits of D/A converter and control or register information have been transmitted from the TMS320 serial port to the AIC. This signal can be used to interrupt a microprocessor upon the completion of serial communications. Also, this signal can be used to strobe and enable external serial-to-parallel shift registers, latches, or an external FIFO RAM, and to facilitate parallel data-bus communications between the AIC and the serial-to-parallel shift registers. During the byte-mode timing, this signal goes low after the first byte has been transmitted from the TMS320 serial port to the AIC and is kept low until the second byte has been transmitted. The TMS320C17 can use this low-going signal to differentiate between the two bytes as to which is first and which is second.
$\overline{\mathrm{FSR}}$	4	O	In the serial transmission modes, which are described in the WORD/$\overline{\mathrm{BYTE}}$ pin description, the $\overline{\mathrm{FSR}}$ pin is held low during bit transmission. When the $\overline{\mathrm{FSR}}$ pin goes low, the TMS320 serial port will begin receiving bits from the AIC via the DR pin of the AIC. The most significant DR bit will be present on the DR pin before $\overline{\mathrm{FSR}}$ goes low. (See Serial Port Timing and Internal Timing Configuration Diagrams.)
$\overline{\mathrm{FSX}}$	14	O	When this pin goes low, the TMS320 serial port will begin transmitting bits to the AIC via the DX pin of the AIC. In all serial transmission modes, which are described in the WORD/$\overline{\mathrm{BYTE}}$ pin description, the $\overline{\mathrm{FSX}}$ pin is held low during bit transmission (see Serial Port Timing and Internal Timing Configuration Diagrams).
IN +	26	I	Noninverting input to analog input amplifier stage
IN −	25	I	Inverting input to analog input amplifier stage
MSTR CLK	6	I	The Master Clock signal is used to derive all the key logic signals of the AIC, such as the Shift Clock, the switched-capacitor filter clocks, and the A/D and D/A timing signals. The Internal Timing Configuration diagram shows how these key signals are derived. The frequencies of these key signals are synchronous submultiples of the Master Clock frequency to eliminate unwanted aliasing when the sampled analog signals are transferred between the switched-capacitor filters and the A/D and D/A converters (see the Internal Timing Configuration).
OUT +	22	O	Noninverting output of analog output power amplifier. Can drive transformer hybrids or high-impedance loads directly in either a differential or a single-ended configuration.
OUT −	21	O	Inverting output of analog output power amplifier. Functionally identical with and complementary to OUT +.
REF	8	I/O	The internal voltage reference is brought out on this pin. An external voltage reference can also be applied to this pin.
$\overline{\mathrm{RESET}}$	2	I	A reset function is provided to initialize the TA, TA′, TB, RA, RA′, RB, and control registers. This reset function initiates serial communications between the AIC and DSP. The reset function will initialize all AIC registers including the control register. After a negative-going pulse on the $\overline{\mathrm{RESET}}$ pin, the AIC registers will be initialized to provide an 8-kHz data conversion rate for a 5.184-MHz master clock input signal. The conversion rate adjust registers, TA′ and RA′, will be reset to 1. The CONTROL register bits will be reset as follows (see AIC DX Data Word Format section). d9 = 1, d7 = 1, d6 = 1, d5 = 1, d4 = 0, d3 = 0, d2 = 1 This initialization allows normal serial-port communication to occur between AIC and DSP.
SHIFT CLK	10	O	The Shift Clock signal is obtained by dividing the Master Clock signal frequency by four. This signal is used to clock the serial data transfers of the AIC, described in the WORD/$\overline{\mathrm{BYTE}}$ pin description below (see the Serial Port Timing and Internal Timing Configuration diagram).
V_{DD}	7		Digital supply voltage, 5 V ±5%
V_{CC+}	20		Positive analog supply voltage, 5 V ±5%
V_{CC-}	19		Negative analog supply voltage −5 V ±5%

PIN		I/O	DESCRIPTION
NAME	NO.		
WORD/$\overline{\text{BYTE}}$	13	I	This pin, in conjunction with a bit in the CONTROL register, is used to establish one of four serial modes. These four serial modes are described below. *AIC transmit and receive sections are operated asynchronously.* The following description applies when the AIC is configured to have asynchronous transmit and receive sections. If the appropriate data bit in the Control register is a 0 (see the AIC DX Data Word Format), the transmit and receive sections will be asynchronous. L Serial port directly interfaces with the serial port of the TMS320C17 and communicates in two 8-bit bytes. The operation sequence is as follows (see Serial Port Timing diagrams). 1. The $\overline{\text{FSX}}$ or $\overline{\text{FSR}}$ pin is brought low. 2. One 8-bit byte is transmitted or one 8-bit byte is received. 3. The $\overline{\text{EODX}}$ or $\overline{\text{EODR}}$ pin is brought low. 4. The $\overline{\text{FSX}}$ or $\overline{\text{FSR}}$ pin emits a positive frame-sync pulse that is four Shift Clock cycles wide. 5. One 8-bit byte is transmitted or one 8-bit byte is received. 6. The $\overline{\text{EODX}}$ or $\overline{\text{EODR}}$ pin is brought high. 7. The $\overline{\text{FSX}}$ or $\overline{\text{FSR}}$ pin is brought high. H Serial port directly interfaces with the serial ports of the TMS32020, TMS320C25, and TMS320C30, and communicates in one 16-bit word. The operation sequence is as follows (see Serial Port Timing diagrams): 1. The $\overline{\text{FSX}}$ or $\overline{\text{FSR}}$ pin is brought low. 2. One 16-bit word is transmitted or one 16-bit word is received. 3. The $\overline{\text{FSX}}$ or $\overline{\text{FSR}}$ pin is brought high. 4. The $\overline{\text{EODX}}$ or $\overline{\text{EODR}}$ pin emits a low-going pulse. *AIC transmit and receive sections are operated synchronously.* If the appropriate data bit in the Control register is a 1, the transmit and receive sections will be configured to be synchronous. In this case, the bandpass switched-capacitor filter and the A/D conversion timing will be derived from the TX Counter A, TX Counter B, and TA, TA', and TB registers, rather than the RX Counter A, RX Counter B, and RA, RA', and RB registers. In this case, the AIC $\overline{\text{FSX}}$ and $\overline{\text{FSR}}$ timing will be identical during primary data communication; however, $\overline{\text{FSR}}$ will not be asserted during secondary data communication since there is no new A/D conversion result. The synchronous operation sequences are as follows (see Serial Port Timing diagrams). L Serial port directly interfaces with the serial port of the TMS320C17 and communicates in two 8-bit bytes. The operation sequence is as follows (see Serial Port Timing diagrams): 1. The $\overline{\text{FSX}}$ and $\overline{\text{FSR}}$ pins are brought low. 2. One 8-bit byte is transmitted and one 8-bit byte is received. 3. The $\overline{\text{EODX}}$ and $\overline{\text{EODR}}$ pins are brought low. 4. The $\overline{\text{FSX}}$ and $\overline{\text{FSR}}$ pins emit positive frame-sync pulses that are four Shift Clock cycles wide. 5. One 8-bit byte is transmitted and one 8-bit byte is received. 6. The $\overline{\text{EODX}}$ and $\overline{\text{EODR}}$ pins are brought high. 7. The $\overline{\text{FSX}}$ and $\overline{\text{FSR}}$ pins are brought high. H Serial port directly interfaces with the serial ports of the TMS32020, TMS320C25, and TMS320C30, and communicates in one 16-bit word. The operation sequence is as follows (see Serial Port Timing diagrams): 1. The $\overline{\text{FSX}}$ and $\overline{\text{FSR}}$ pins are brought low. 2. One 16-bit word is transmitted and one 16-bit word is received. 3. The $\overline{\text{FSX}}$ and $\overline{\text{FSR}}$ pins are brought high. 4. The $\overline{\text{EODX}}$ or $\overline{\text{EODR}}$ pins emit low-going pulses. Since the transmit and receive sections of the AIC are now synchronous, the AIC serial port, with additional NOR and AND gates, will interface to two SN74299 serial-to-parallel shift registers. Interfacing the AIC to the SN74299 shift register allows the AIC to interface to an external FIFO RAM and facilitates parallel, data bus communications between the AIC and the digital signal processor. The operation sequence is the same as the above sequence (see Serial Port Timing diagrams).

INTERNAL TIMING CONFIGURATION



NOTE: Frequency 1, 20.736 MHz, is used to show how 153.6 kHz (for a commercially available modem split-band filter clock), popular speech and modem sampling signal frequencies, and an internal 288-kHz switched-capacitor filter clock can be derived synchronously and as submultiples of the crystal oscillator frequency. Since these derived frequencies are synchronous submultiples of the crystal frequency, aliasing does not occur as the sampled analog signal passes between the analog converter and switched-capacitor filter stages. Frequency 2, 41.472 MHz, is used to show that the AIC can work with high-frequency signals, which are used by high-speed digital signal processors.

†Split-band filtering can alternatively be performed after the analog input function via software in the TMS320.

‡These control bits are described in the AIC DX Data Word Format section.

explanation of internal timing configuration

All of the internal timing of the AIC is derived from the high-frequency clock signal that drives the Master Clock input pin. The Shift Clock signal, which strobes the serial port data between the AIC and DSP, is derived by dividing the Master Clock input signal frequency by four.

Low-pass:

$$\text{SCF Clock Frequency (D/A or A/D Path)} = \frac{\text{Master Clock Frequency}}{2 \times \text{Contents of Counter A}}$$

$$\text{Conversion Frequency} = \frac{\text{SCF Clock Frequency (D/A or A/D Path)}}{\text{Contents of Counter B}}$$

High-pass:

$$\text{SCF Clock Frequency (A/D Path)} = \text{A/D Conversion Frequency}$$

$$\text{Shift Clock Frequency} = \frac{\text{Master Clock Frequency}}{4}$$

TX Counter A and TX Counter B, which are driven by the Master Clock signal, determine the D/A conversion timing. Similarly, RX Counter A and RX Counter B determine the A/D conversion timing. In order for the low-pass switched-capacitor filter in the D/A path to meet its transfer function specifications, the frequency of its clock input must be 288 kHz. If the clock frequency is not 288 kHz, the filter transfer function frequencies are frequency-scaled by the ratios of the clock frequency to 288 kHz. Thus, to obtain the specified filter response, the combination of Master Clock frequency and TX Counter A and RX Counter A values must yield a 288-kHz switched-capacitor clock signal. This 288-kHz clock signal can then be divided by the TX Counter B to establish the D/A conversion timing.

The transfer function of the bandpass switched-capacitor filter in the A/D path is a composite of its high-pass and low-pass section transfer functions. The high-frequency roll-off of the low-pass section will meet the bandpass filter transfer function specification when the low-pass section SCF is 288 kHz. Otherwise, the high-frequency roll-off will be frequency-scaled by the ratio of the high-pass section's SCF clock to 288 kHz. The low-frequency roll-off of the high-pass section will meet the bandpass filter transfer function specification when the A/D conversion rate is 8 kHz. Otherwise, the low-frequency roll-off of the high-pass section will be frequency-scaled by the ratio of the A/D conversion rate to 8 kHz.

TX Counter A and TX Counter B are reloaded every D/A conversion period, while RX Counter A and RX Counter B are reloaded every A/D conversion period. The TX Counter B and RX Counter B are loaded with the values in the TB and RB Registers, respectively. Via software control, the TX Counter A can be loaded with either the TA Register, the TA Register less the TA′ Register, or the TA Register plus the TA′ Register. By selecting the TA Register less the TA′ Register option, the upcoming conversion timing will occur earlier by an amount of time that equals TA′ times the signal period of the Master Clock. By selecting the TA Register plus the TA′ Register option, the upcoming conversion timing will occur later by an amount of time that equals TA′ times the signal period of the Master Clock. Thus, the D/A conversion timing can be advanced or retarded. An identical ability to alter the A/D conversion timing is provided. In this case, however, the RX Counter A can be programmed via software control with the RA Register, the RA Register less the RA′ Register, or the RA Register plus the RA′ Register.

The ability to advance or retard conversion timing is particularly useful for modem applications. This feature allows controlled changes in the A/D and D/A conversion timing. This feature can be used to enhance signal-to-noise performance, to perform frequency-tracking functions, and to generate nonstandard modem frequencies.

If the transmit and receive sections are configured to be synchronous (see WORD/$\overline{\text{BYTE}}$ pin description), then both the low-pass and bandpass switched-capacitor filter clocks are derived from TX Counter A. Also, both the D/A and A/D conversion timing are derived from the TX Counter A and TX Counter B. When the transmit and receive sections are configured to be synchronous, the RX Counter A, RX Counter B, RA Register, RA′ Register, and RB Registers are not used.

AIC DR or DX word bit pattern

A/D or D/A MSB, 1st bit sent ↓								1st bit sent of 2nd byte ↓					A/D or D/A LSB ↓		
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

AIC DX data word format section

d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0	COMMENTS
primary DX serial communication protocol																
← d15 (MSB) through d2 go to the D/A converter register →														0	0	The TX and RX Counter A's are loaded with the TA and RA register values. The TX and RX Counter B's are loaded with TB and RB register values.
← d15 (MSB) through d2 go to the D/A converter register →														0	1	The TX and RX Counter A's are loaded with the TA + TA′ and RA + RA′ register values. The TX and RX Counter B's are loaded with the TB and RB register values. NOTE: d1 = 0, d0 = 1 will cause the next D/A and A/D conversion periods to be changed by the addition of TA′ and RA′ Master Clock cycles, in which TA′ and RA′ can be positive or negative or zero. Please refer to Table 1. AIC Responses to Improper Conditions.
← d15 (MSB) through d2 go to the D/A converter register →														1	0	The TX and RX Counter A's are loaded with the TA − TA′ and RA − RA′ register values. The TX and RX Counter B's are loaded with the TB and RB register values. NOTE: d1 = 1, d0 = 0 will cause the next D/A and A/D conversion periods to be changed by the subtraction of TA′ and RA′ Master Clock cycles, in which TA′ and RA′ can be positive or negative or zero. Please refer to Table 1. AIC Responses to Improper Conditions.
← d15 (MSB) through d2 go to the D/A converter register →														1	1	The TX and RX Counter A's are loaded with the TA and RA register converter register values. The TX and RX Counter B's are loaded with the TB and RB register values. After a delay of four Shift Clock cycles, a secondary transmission will immediately follow to program the AIC to operate in the desired configuration.

NOTE: Setting the two least significant bits to 1 in the normal transmission of DAC information (Primary Communications) to the AIC will initiate Secondary Communications upon completion of the Primary Communications.

Upon completion of the Primary Communication, $\overline{\text{FSX}}$ will remain high for four SHIFT CLOCK cycles and will then go low and initiate the Secondary Communication. The timing specifications for the Primary and Secondary Communications are identical. In this manner, the Secondary Communication, if initiated, is interleaved between successive Primary Communications. This interleaving prevents the Secondary Communication from interfering with the Primary Communications and DAC timing, thus preventing the AIC from skipping a DAC output. It is important to note that in the synchronous mode, $\overline{\text{FSR}}$ will not be asserted during Secondary Communications.

secondary DX serial communication protocol

x x | ← to TA register → | x x | ← to RA register → | 0 0	d13 and d6 are MSBs (unsigned binary)
x | ← to TA′ register → | x | ← to RA′ register → | 0 1	d14 and d7 are 2′s complement sign bits
x | ← to TB register → | x | ← to RB register → | 1 0	d14 and d7 are MSBs (unsigned binary)

x x x x x x d9 x d7 d6 d5 d4 d3 d2 1 1
◄—— CONTROL REGISTER ——►
	d2 = 0/1 deletes/inserts the A/D high-pass filter
	d3 = 0/1 disables/enables the loopback function
	d4 = 0/1 disables/enables the AUX IN+ and AUX IN− pins
	d5 = 0/1 asynchronous/synchronous transmit and receive sections
	d6 = 0/1 gain control bits (see Gain Control Section)
	d7 = 0/1 gain control bits (see Gain Control Section)
	d9 = 0/1 delete/insert on-board second-order (sin x)/x correction filter

reset function

A reset function is provided to initiate serial communications between the AIC and DSP. The reset function will initialize all AIC registers, including the control register. After power has been applied to the AIC, a negative-going pulse on the $\overline{\text{RESET}}$ pin will initialize the AIC registers to provide an 8-kHz A/D and D/A conversion rate for a 5.184 MHz master clock input signal. The AIC, excepting the CONTROL register, will be initialized as follows (see AIC DX Data Word Format section):

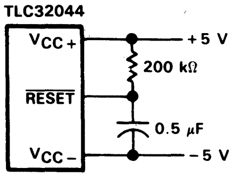
REGISTER	INITIALIZED REGISTER VALUE (HEX)
TA	9
TA′	1
TB	24
RA	9
RA′	1
RB	24

The CONTROL register bits will be reset as follows (see AIC DX Data Word Format section):

d9 = 1, d7 = 1, d6 = 1, d5 = 1, d4 = 0, d3 = 0, d2 = 1

This initialization allows normal serial port communications to occur between AIC and DSP. If the transmit and receive sections are configured to operate synchronously and the user wishes to program different conversion rates, only the TA, TA′, and TB register need to be programmed, since both transmit and receive timing are synchronously derived from these registers (see the Pin Descriptions and AIC DX Word Format sections).

The circuit shown below will provide a reset on power-up when power is applied in the sequence given under Power-Up Sequence. The circuit depends on the power supplies' reaching their recommended values a minimum of 800 ns before the capacitor charges to 0.8 V above DGTL GND.



power-up sequence

To ensure proper operation of the AIC, and as a safeguard against latch-up, it is recommmended that Schottky diodes with forward voltages less than or equal to 0.4 V be connected from V_{CC-} to ANLG GND and from V_{CC-} to DGTL GND (see Figure 21). In the absence of such diodes, power should be applied in the following sequence: ANLG GND and DGTL GND, V_{CC-}, then V_{CC+} and V_{DD}. Also, no input signal should be applied until after power-up.

AIC responses to improper conditions

The AIC has provisions for responding to improper conditions. These improper conditions and the response of the AIC to these conditions are presented in Table 1 below.

AIC register constraints

The following constraints are placed on the contents of the AIC registers:

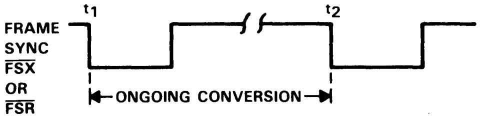
1. TA register must be > 1.
2. TA′ register can be either positive, negative, or zero.
3. RA register must be > 1.
4. RA′ register can be either positive, negative, or zero.
5. (TA register ± TA′ register) must be > 1.
6. (RA register ± RA′ register) must be > 1.
7. TB register must be > 1.

TABLE 1. AIC RESPONSES TO IMPROPER CONDITIONS

IMPROPER CONDITION	AIC RESPONSE
TA register + TA′ register = 0 or 1 TA register − TA′ register = 0 or 1	Reprogram TX Counter A with TA register value
TA register + TA′ register < 0	MODULO 64 arithmetic is used to ensure that a positive value is loaded into the TX Counter A, i.e., TA register + TA′ register + 40 HEX is loaded into TX Counter A
RA register + RA′ register = 0 or 1 RA register − RA′ register = 0 or 1	Reprogram RX Counter A with RA register value
RA register + RA′ register = 0 or 1	MODULO 64 arithmetic is used to ensure that a positive value is loaded into RX Counter A, i.e., RA register + RA′ register + 40 HEX is loaded into RX Counter A
TA register = 0 or 1 RA register = 0 or 1	AIC is shut down
TB register = 0 or 1	Reprogram TB register with 24 HEX
RB register = 0 or 1	Reprogram RB register with 24 HEX
AIC and DSP cannot communicate	Hold last DAC output

improper operation due to conversion times being too close together

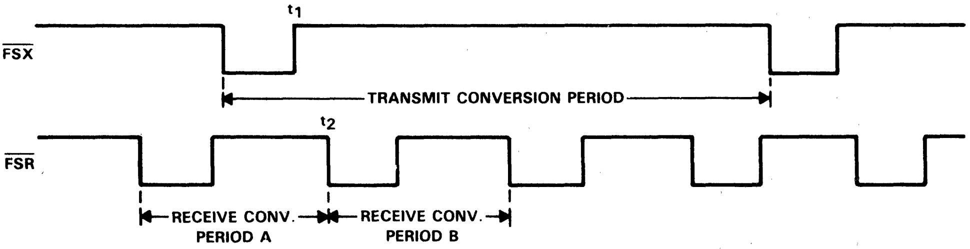
If the difference between two successive D/A conversion frame syncs is less that 1/19.2 kHz, the AIC operates improperly. In this situation, the second D/A conversion frame sync occurs too quickly and there is not enough time for the ongoing conversion to be completed. This situation can occur if the A and B registers are improperly programmed or if the A + A′ register or A − A′ register result is too small. When incrementally adjusting the conversion period via the A + A′ register options, the designer should be very careful not to violate this requirement (see diagram below).



$t_2 - t_1 \geq 1/19.2$ kHz

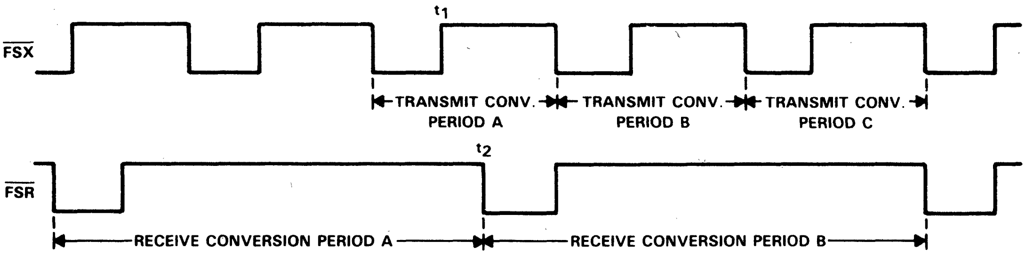
asynchronous operation — more than one receive frame sync occurring between two transmit frame syncs

When incrementally adjusting the conversion period via the A + A′ or A − A′ register options, a specific protocol is followed. The command to use the incremental conversion period adjust option is sent to the AIC during a $\overline{\text{FSX}}$ frame sync. The ongoing conversion period is then adjusted. However, either Receive Conversion Period A or B may be adjusted. For both transmit and receive conversion periods, the incremental conversion period adjustment is performed near the end of the conversion period. Therefore, if there is sufficient time between t1 and t2, the receive conversion period adjustment will be performed during Receive Conversion Period A. Otherwise, the adjustment will be performed during Receive Conversion Period B. The adjustment command only adjusts one transmit conversion period and one receive conversion period. To adjust another pair of transmit and receive conversion periods, another command must be issued during a subsequent $\overline{\text{FSX}}$ frame (see figure below).



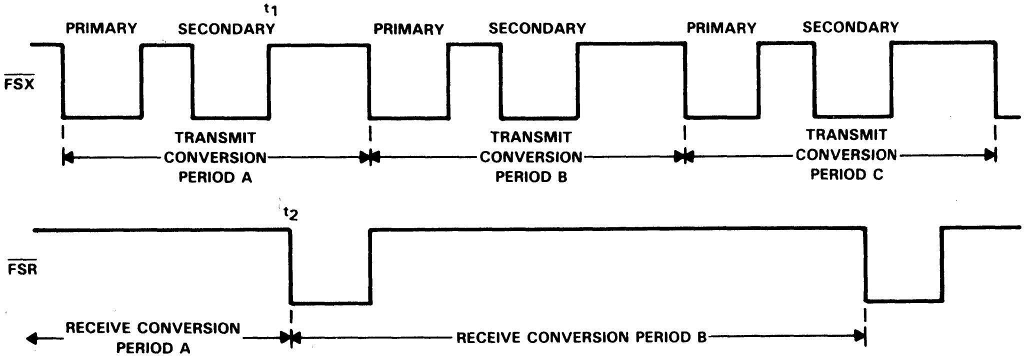
asynchronous operation — more than one transmit frame sync occurring between two receive frame syncs

When incrementally adjusting the conversion period via the A + A′ or A − A′ register options, a specific protocol is followed. For both transmit and receive conversion periods, the incremental conversion period adjustment is performed near the end of the conversion period. The command to use the incremental conversion period adjust options is sent to the AIC during a $\overline{\text{FSX}}$ frame sync. The ongoing transmit conversion period is then adjusted. However, three possibilities exist for the receive conversion period adjustment in the diagram as shown in the figure below. If the adjustment command is issued during Transmit Conversion Period A, Receive Conversion Period A will be adjusted if there is sufficient time between t1 and t2. Or, if there is not sufficient time between t1 and t2, Receive Conversion Period B will be adjusted. Or, the receive portion of an adjustment command may be ignored if the adjustment command is sent during a receive conversion period, which is already being or will be adjusted due to a prior adjustment command. For example, if adjustment commands are issued during Transmit Conversion Periods A, B, and C, the first two commands may cause Receive Conversion Periods A and B to be adjusted, while the third receive adjustment command is ignored. The third adjustment command is ignored since it was issued during Receive Conversion Period B, which already will be adjusted via the Transmit Conversion Period B adjustment command.



asynchronous operation — more than one set of primary and secondary DX serial communication occurring between two receive frame sync (see AIC DX Data Word Format section)

The TA, TA′, TB, and control register information that is transmitted in the secondary communications is always accepted and is applied during the ongoing transmit conversion period. If there is sufficient time between t1 and t2, the TA, RA′, and RB register information, which is sent during Transmit Conversion Period A, will be applied to Receive Conversion Period A. Otherwise, this information will be applied during Receive Conversion Period B. If RA, RA′, and RB register information has already been received and is being applied during an ongoing conversion period, any subsequent RA, RA′, or RB information that is received during this receive conversion period will be disregarded (see diagram below).


t_1
PRIMARY
SECONDARY
PRIMARY
SECONDARY
PRIMARY
SECONDARY
$\overline{FSX}$
TRANSMIT CONVERSION PERIOD A
TRANSMIT CONVERSION PERIOD B
TRANSMIT CONVERSION PERIOD C
t_2
$\overline{FSR}$
RECEIVE CONVERSION PERIOD A
RECEIVE CONVERSION PERIOD B

test modes[†]

The following paragraph provides information that allows the TLC32044 to be operated in special test modes. These test modes are used by Texas Instruments to facilitate testing of the device during manufacturing. They are not intended to be used in real applications, however, they allow the filters in the A/D and D/A paths to be used without using the A/D and D/A converters.

In normal operation, the nonusable (NU) pins are left unconnected. These NU pins are used by the factory to speed up testing of the TLC32044 Analog Interface Circuit (AIC). When the device is used in normal (non-test-mode) operation, the NU pin (pin 1) has an internal pull-down to −5 V. Externally connecting 0 V or 5 V to pin 1 puts the device in test-mode operation. Selecting one of the possible test modes is accomplished by placing a particular voltage on certain pins. A description of these modes is provided in Table 2 and Figures 1 and 2.

TABLE 2. LIST OF TEST MODES

<table>
<tr><th rowspan="2">TEST PINS</th><th>D/A PATH TEST (PIN 1 to 5 V)</th><th>A/D PATH TEST (PIN 1 to 0)</th></tr>
<tr><th>TEST FUNCTION</th><th>TEST FUNCTION</th></tr>
<tr><td>5</td><td>The low-pass switched-capacitor filter clock is brought out to pin 5. This clock signal is normally internal.</td><td>The bandpass switched-capacitor filter clock is brought out to pin 5. This clock signal is normally internal.</td></tr>
<tr><td>11</td><td>No change from normal operation. The $\overline{\text{EODX}}$ signal is brought out to pin 11.</td><td>The pulse that initiates the A/D conversion is brought out here. This signal is normally internal.</td></tr>
<tr><td>3</td><td>The pulse that initiates the D/A conversion is brought out here.</td><td>No change from normal operation. The EODR signal is brought out.</td></tr>
<tr><td>27 and 28</td><td>There are no test output signals provided on these pins.</td><td>The outputs of the A/D path low-pass or bandpass filter (depending upon control bit d2 — see AIC DX Data Word Format section) are brought out to these pins. If the high-pass section is inserted, the output will have a (sinx)/x droop. The slope of the droop will be determined by the ADC sampling frequency, which is the high-pass section clock frequency (see diagram of bandpass or low-pass filter test for receive section). These outputs will drive small (30-pF) loads.</td></tr>
<tr><td rowspan="3">15 and 16</td><th colspan="2">D/A PATH LOW-PASS FILTER TEST; PIN 13 (WORD/$\overline{\text{BYTE}}$) to −5 V</th></tr>
<tr><th colspan="2">TEST FUNCTION</th></tr>
<tr><td colspan="2">The inputs of the D/A path low-pass filter are brought out to pins 15 and 16. The D/A input to this filter is removed. If the (sin x)/x correction filter is inserted, the OUT+ and OUT− signals will have a flat response (see Figure 2). The common-mode range of these inputs must not exceed ±0.5 V.</td></tr>
</table>

[†] In the test mode, the AIC responds to the setting of Pin 13 to −5 V, as if Pin 13 were set to 0 V. Thus, the byte mode is selected for communicating between DSP and AIC. Either of the path tests (D/A or A/D) can be performed simultaneously with the D/A low-pass filter test. In this situation, Pin 13 must be connected to −5 V, which initiates byte-mode communications.

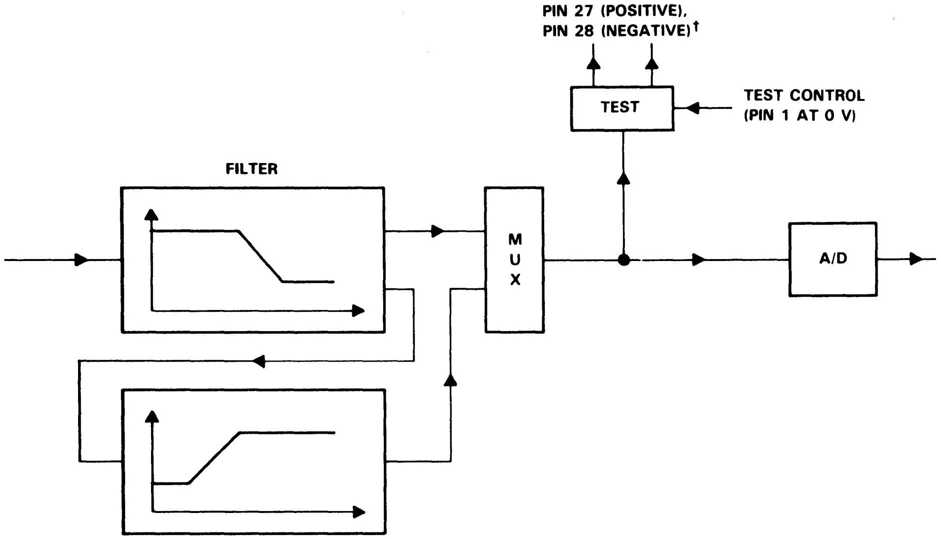


FIGURE 1. BANDPASS OR LOW-PASS FILTER TEST FOR RECEIVER SECTION

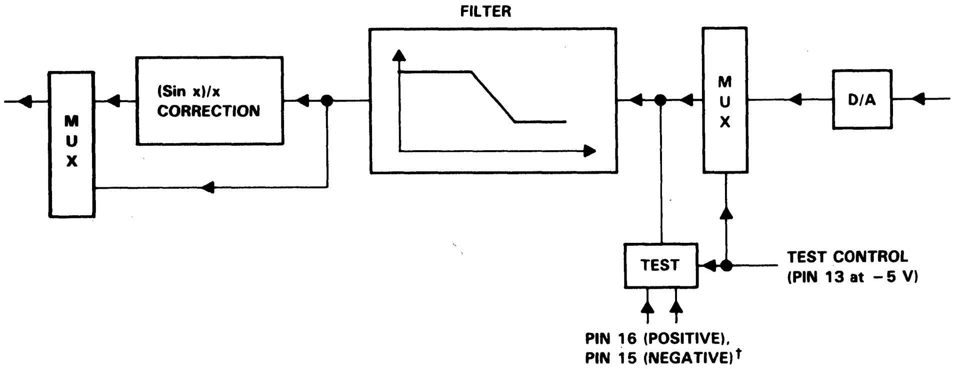


FIGURE 2. LOW-PASS FILTER TEST FOR TRANSMIT SECTION

†All analog signal paths have differential architecture and hence have positive and negative components.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC+} (see Note 1)		−0.3 V to 15 V
Supply voltage, V_{DD}		−0.3 V to 15 V
Output voltage, V_O		−0.3 V to 15 V
Input voltage, V_I		−0.3 V to 15 V
Digital ground voltage		−0.3 V to 15 V
Operating free-air temperature range:	TLC32044I	−40°C to 85°C
	TLC32044C	0°C to 70°C
Storage temperature range		−40°C to 125°C
Case temperature for 10 seconds: FN package		260°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds: N package		260°C

NOTE 1: Voltage values for maximum ratings are with respect to VCC−.

recommended operating conditions

PARAMETER		MIN	NOM	MAX	UNIT
Supply voltage, V_{CC+} (see Note 2)		4.75	5	5.25	V
Supply voltage, V_{CC-} (see Note 2)		−4.75	−5	−5.25	V
Digital supply voltage, V_{DD} (see Note 2)		4.75	5	5.25	V
Digital ground voltage with respect to ANLG GND, DGTL GND			0		V
Reference input voltage, $V_{ref(ext)}$ (see Note 2)		2		4	V
High-level input voltage, V_{IH}		2		V_{DD}+0.3	V
Low-level input voltage, V_{IL} (see Note 3)		−0.3		0.8	V
Load resistance at OUT+ and/or OUT−, R_L		300			Ω
Load capacitance at OUT+ and/or OUT−, C_L				100	pF
MSTR CLK frequency (see Note 4)		0.075	5	10.368	MHz
Analog input amplifier common mode input voltage (see Note 5)				±1.5	V
A/D or D/A conversion rate				19.2	kHz
Conversion rate				20	kHz
Operating free-air temperature, T_A	TLC32044I	−40		85	°C
	TLC32044C	0		70	

NOTES: 2. Voltages at analog inputs and outputs, REF, V_{CC+}, and V_{CC-}, are with respect to the ANLG GND terminal. Voltages at digital inputs and outputs and V_{DD} are with respect to the DGTL GND terminal.

3. The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels and temperature only.

4. The bandpass switched-capacitor filter (SCF) specifications apply only when the low-pass section SCF clock is 288 kHz and the high-pass section SCF clock is 8 kHz. If the low-pass SCF clock is shifted from 288 kHz, the low-pass roll-off frequency will shift by the ratio of the low-pass SCF clock to 288 kHz. If the high-pass SCF clock is shifted from 8 kHz, the high-pass roll-off frequency will shift by the ratio of the high-pass SCF clock to 8 kHz. Similarly, the low-pass switched-capacitor filter (SCF) specifications apply only when the SCF clock is 288 kHz. If the SCF clock is shifted from 288 kHz, the low-pass roll-off frequency will shift by the ratio of the SCF clock to 288 kHz.

5. This range applies when (IN+ − IN−) or (AUX IN+ − AUX IN−) equals ±6 V.

electrical characteristics over recommended operating free-air temperature range, V_{CC+} = 5 V, V_{CC-} = −5 V, V_{DD} = 5 V (unless otherwise noted)

total device, MSTR CLK frequency = 5.184 MHz, outputs not loaded

PARAMETER			TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage		V_{DD} = 4.75 V, I_{OH} = −300 μA	2.4			V
V_{OL}	Low-level output voltage		V_{DD} = 4.75 V, I_{OL} = 2 mA			0.4	V
I_{CC+}	Supply current from V_{CC+}	TLC32044I				40	mA
		TLC32044C				35	
I_{CC-}	Supply current from V_{CC-}	TLC32044I				−40	mA
		TLC32044C				−35	
I_{DD}	Supply current from V_{DD}		$f_{MSTR\ CLK}$ = 5.184 MHz			7	mA
V_{ref}	Internal reference output voltage			3		3.3	V
αV_{ref}	Temperature coefficient of internal reference voltage				250		ppm/°C
r_o	Output resistance at REF				100		kΩ

receive amplifier input

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
	A/D converter offset error (filters in)			10	70	mV
CMRR	Common-mode rejection ratio at IN+, IN−, or AUX IN+, AUX IN−	See Note 6		55		dB
r_I	Input resistance at IN+, IN− or AUX IN+, AUX IN−, REF			100		kΩ

transmit filter output

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OO}	Output offset voltage at OUT+ or OUT− (single-ended relative to ANLG GND)			15	80	mV
V_{OM}	Maximum peak output voltage swing across R_L at OUT+ or OUT− (single-ended)	R_L ≥ 300 Ω, Offset voltage = 0	±3			V
V_{OM}	Maximum peak output voltage swing between OUT+ and OUT− (differential output)	R_L ≥ 600 Ω	±6			V

†All typical values are at T_A = 25°C.

NOTE 6: The test condition is a 0-dBm, 1-kHz input signal with an 8-kHz conversion rate.

electrical characteristics over recommended operating free-air temperature range, V_{CC+} = 5 V, V_{CC-} = −5 V, V_{DD} = 5 V (unless otherwise noted)

system distortion specifications, SCF clock frequency = 288 kHz

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
Attenuation of second harmonic of A/D input signal	single-ended	V_{in} = −0.1 dB to −24 dB referred to V_{ref}, See Note 7		70		dB
	differential		62	70		
Attenuation of third and higher harmonics of A/D input signal	single-ended	V_{in} = −0.1 dB to −24 dB referred to V_{ref}, See Note 7		65		dB
	differential		57	65		
Attenuation of second harmonic of D/A input signal	single-ended	V_{in} = −0 dB to −24 dB referred to V_{ref}, See Note 7		70		dB
	differential		62	70		
Attenuation of third and higher harmonics of D/A input signal	single-ended	V_{in} = −0 dB to −24 dB referred to V_{ref}, See Note 7		65		dB
	differential		57	65		

A/D channel signal-to-distortion ratio

PARAMETER	TEST CONDITIONS (see Note 7)	A_V = 1‡		A_V = 2‡		A_V = 4‡		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
A/D channel signal-to-distortion ratio	V_{in} = −6 dB to −0.1 dB	58		>58§		>58§		dB
	V_{in} = −12 dB to −6 dB	58		58		>58§		
	V_{in} = −18 dB to −12 dB	56		58		58		
	V_{in} = −24 dB to −18 dB	50		56		58		
	V_{in} = −30 dB to −24 dB	44		50		56		
	V_{in} = −36 dB to −30 dB	38		44		50		
	V_{in} = −42 dB to −36 dB	32		38		44		
	V_{in} = −48 dB to −42 dB	26		32		38		
	V_{in} = −54 dB to −48 dB	20		26		32		

† All typical values are at T_A = 25°C.

‡ A_V is the programmable gain of the input amplifier.

§ A value >60 is over range and signal clipping occurs.

D/A channel signal-to-distortion ratio

PARAMETER	TEST CONDITIONS (see Note 7)	MIN	MAX	UNIT
D/A channel signal-to-distortion ratio	V_{in} = −6 dB to −0.1 dB	58		dB
	V_{in} = −12 dB to −6 dB	58		
	V_{in} = −18 dB to −12 dB	56		
	V_{in} = −24 dB to −18 dB	50		
	V_{in} = −30 dB to −24 dB	44		
	V_{in} = −36 dB to −30 dB	38		
	V_{in} = −42 dB to −36 dB	32		
	V_{in} = −48 dB to −42 dB	26		
	V_{in} = −54 dB −48 dB	20		

NOTE 7: The test condition is a 1-kHz input signal with an 8-kHz conversion rate (0 dB relative to V_{ref}). The load impedance for the DAC is 600 Ω.

electrical characteristics over recommended operating free-air temperature range, $V_{CC+} = 5$ V, $V_{CC-} = -5$ V, $V_{DD} = 5$ V (unless otherwise noted) (Continued)

gain and dynamic range

PARAMETER	TEST CONDITIONS	MIN	TYP[†]	MAX	UNIT
Absolute transmit gain tracking error while transmitting into 600 Ω	−48 dB to 0 dB signal range, See Note 8		±0.05	±0.15	dB
Absolute receive gain tracking error	−48 dB to 0 dB signal range, See Note 8		±0.05	±0.15	dB

power supply rejection and crosstalk attenuation

PARAMETER		TEST CONDITIONS	MIN	TYP[†]	MAX	UNIT
V_{CC+} or V_{CC-} supply voltage rejection ratio, receive channel	f = 0 to 30 kHz	Idle channel, supply signal at 200 mV p-p measured at DR (ADC output)		30		dB
	f = 30 kHz to 50 kHz			45		
V_{CC+} or V_{CC-} supply voltage rejection ratio, transmit channel (single-ended)	f = 0 to 30 kHz	Idle channel, supply signal at 200 mV p-p measured at OUT+		30		dB
	f = 30 kHz to 50 kHz			45		
Crosstalk attenuation, transmit-to-receive (single-ended)				80		dB

[†] All typical values are at $T_A = 25°C$.

NOTE 8: Gain tracking is relative to the absolute gain at 1 kHz and 0 dB (0 dB relative to V_{ref}).

delay distortion

bandpass filter transfer function, SCF f_{clock} = 288 kHz, input (IN+ − IN−) is a ±3-V sinewave† (see Note 9)

PARAMETER	TEST CONDITION	FREQUENCY RANGE	ADJUSTMENT ADDEND‡	MIN	TYP§	MAX	UNIT
Filter gain	Input signal reference is 0 dB	f ≤ 50 Hz	K1 × 0 dB	−33	−29	−25	dB
		f = 100 Hz	K1 × − 0.26 dB	−4	−2	−1	
		f = 150 Hz to 3100 Hz	K1 × 0 dB	−0.25	0	0.25	
		f = 3100 Hz to 3300 Hz	K1 × 0 dB	−0.3	0	0.3	
		f = 3300 Hz to 3650 Hz	K1 × 0 dB	−0.5	0	0.5	
		f = 3800 Hz	K1 × 2.3 dB	−5	−3	−1	
		f = 4000 Hz	K1 × 2.7 dB	−20	−17	−16	
		f ≥ 4400 Hz	K1 × 3.2 dB			−40	
		f ≥ 5000 Hz	K1 × 0 dB			−65	

low-pass filter transfer function, SCF f_{clock} = 288 kHz (see Note 9)

PARAMETER	TEST CONDITION	FREQUENCY RANGE	ADJUSTMENT ADDEND‡	MIN	TYP§	MAX	UNIT
Filter gain	Input signal reference is 0 dB	f = 0 Hz to 3100 Hz	K1 × 0 dB	−0.25	0	0.25	dB
		f = 3100 Hz to 3300 Hz	K1 × 0 dB	−0.3	0	0.3	
		f = 3300 Hz to 3650 Hz	K1 × 0 dB	−0.5	0	0.5	
		f = 3800 Hz	K1 × 2.3 dB	−5	−3	−1	
		f = 4000 Hz	K1 × 2.7 dB	−20	−17	−16	
		f ≥ 4400 Hz	K1 × 3.2 dB			−40	
		f ≥ 5000 Hz	K1 × 0 dB			−65	

serial port

PARAMETER		TEST CONDITIONS	MIN	TYP§	MAX	UNIT
V_{OH}	High-level output voltage	I_{OH} = −300 μA	2.4			V
V_{OL}	Low-level output voltage	I_{OL} = 2 mA			0.4	V
I_I	Input current				±10	μA
C_i	Input capacitance			15		pF
C_o	Output capacitance			15		pF

† See filter curves in typical characteristics.

‡ The MIN, TYP, and MAX specifications are given for a 288-kHz SCF clock frequency. A slight error in the 288-kHz SCF may result from inaccuracies in the MSTR CLK frequency, resulting from crystal frequency tolerances. If this frequency error is less than 0.25%, the ADJUSTMENT ADDEND should be added to the MIN, TYP, and MAX specifications, where K1 = 100 • [(SCF frequency − 288 kHz)/288 kHz]. For errors greater than 0.25%, see Note 10.

§ All typical values are at T_A = 25°C.

NOTES: 9. The filter gain outside of the passband is measured with respect to the gain at 1 kHz. The filter gain within the passband is measured with respect to the average gain within the passband. The passbands are 150 to 3600 Hz and 0 to 3600 Hz for the bandpass and low-pass filters respectively.

10. For switched-capacitor filter clocks at frequencies other than 288 kHz, the filter response is shifted by the ratio of switched-capacitor filter clock frequency to 288 kHz.

operating characteristics over recommended operating free-air temperature range, $V_{CC+} = 5$ V, $V_{CC-} = -5$ V, $V_{DD} = 5$ V

noise (measurement includes low-pass and bandpass switched-capacitor filters)

PARAMETER		TEST CONDITIONS	TYP†	MAX	UNIT
Transmit noise	with (sin x)/x	DX input = 00000000000000, constant input code		550	μV rms
	without (sin x)/x		325	425	μV rms
			18		dBrncO
Receive noise (see Note 11)		Inputs grounded, gain = 1	300	500	μV rms
			18		dBrncO

timing requirements

serial port recommended input signals

PARAMETER		MIN	MAX	UNIT
$t_{c(MCLK)}$	Master clock cycle time	95		ns
$t_{r(MCLK)}$	Master clock rise time		10	ns
$t_{f(MCLK)}$	Master clock fall time		10	ns
	Master clock duty cycle	25%	75%	
	$\overline{\text{RESET}}$ pulse duration (see Note 12)	800		ns
$t_{su(DX)}$	DX setup time before SCLK↓	20		ns
$t_{h(DX)}$	DX hold time after SCLK↓	$t_{c(SCLK)}/4$		ns

†All typical values are at $T_A = 25$ °C.

NOTES: 11. The noise is computed by statistically evaluating the digital output of the A/D converter.

12. $\overline{\text{RESET}}$ pulse duration is the amount of time that the reset pin is held below 0.8 V after the power supplies have reached their recommended values.

operating characteristics over recommended operating free-air temperature range, V_{CC+} = 5 V, V_{CC-} = −5 V, V_{DD} = 5 V (continued)

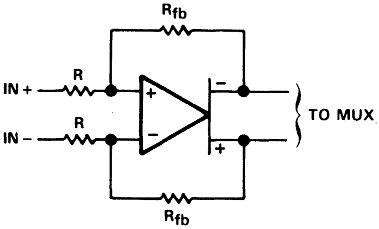
serial port — AIC output signals

PARAMETER		MIN	MAX	UNIT
$t_{c(SCLK)}$	Shift clock (SCLK) cycle time	380		ns
$t_{f(SCLK)}$	Shift clock (SCLK) fall time		50	ns
$t_{r(SCLK)}$	Shift clock (SCLK) rise time		50	ns
	Shift clock (SCLK) duty cycle	45	55	%
$t_{d(CH-FL)}$	Delay from SCLK↑ to $\overline{FSR}/\overline{FSX}$↓		90	ns
$t_{d(CH-FH)}$	Delay from SCLK↑ to $\overline{FSR}/\overline{FSX}$↑		90	ns
$t_{d(CH-DR)}$	DR valid after SCLK↑		90	ns
$t_{dw(CH-EL)}$	Delay from SCLK↑ to $\overline{EODX}/\overline{EODR}$↓ in word mode		90	ns
$t_{dw(CH-EH)}$	Delay from SCLK↑ to $\overline{EODX}/\overline{EODR}$↑ in word mode		90	ns
$t_{f(EODX)}$	$\overline{EODX}$ fall time		15	ns
$t_{f(EODR)}$	$\overline{EODR}$ fall time		15	ns
$t_{db(CH-EL)}$	Delay from SCLK↑ to $\overline{EODX}/\overline{EODR}$↓ in byte mode		100	ns
$t_{db(CH-EH)}$	Delay from SCLK↑ to $\overline{EODX}/\overline{EODR}$↑ in byte mode		100	ns

TABLE 2. GAIN CONTROL TABLE (ANALOG INPUT SIGNAL REQUIRED FOR FULL-SCALE A/D CONVERSION)

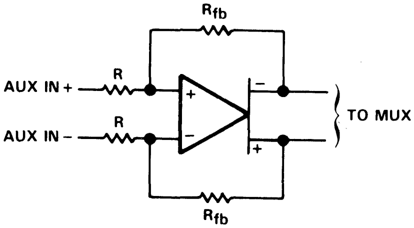
INPUT CONFIGURATIONS	CONTROL REGISTER BITS		ANALOG INPUT[†]	A/D CONVERSION RESULT
	d6	d7		
Differential configuration Analog input = IN+ − IN− = AUX IN+ − AUX IN−	1 0	1 0	±6 V	full-scale
	1	0	±3 V	full-scale
	0	1	±1.5 V	full-scale
Single-ended configuration Analog input = IN+ − ANLG GND = AUX IN+ − ANLG GND	1 0	1 0	±3 V	half-scale
	1	0	±3 V	full-scale
	0	1	±1.5 V	full-scale

[†] In this example, V_{ref} is assumed to be 3 V. In order to minimize distortion, it is recommended that the analog input not exceed 0.1 dB below full scale.



R_{fb} = R for d6 = 1, d7 = 1
d6 = 0, d7 = 0
R_{fb} = 2R for d6 = 1, d7 = 0
R_{fb} = 4R for d6 = 0, d7 = 1

FIGURE 3. IN+ AND IN− GAIN CONTROL CIRCUITRY



R_{fb} = R for d6 = 1, d7 = 1
d6 = 0, d7 = 0
R_{fb} = 2R for d6 = 1, d7 = 0
R_{fb} = 4R for d6 = 0, d7 = 1

FIGURE 4. AUX IN+ AND AUX IN− GAIN CONTROL CIRCUITRY

(sin x)/x correction section

If the designer does not wish to use the on-board second-order (sin x)/x correction filter, correction can be accomplished in digital signal processor (DSP) software. (Sin x)/x correction can be accomplished easily and efficiently in digital signal processor (DSP) software. Excellent correction accuracy can be achieved to a band edge of 3000 Hz by using a first-order digital correction filter. The results, which are shown below, are typical of the numerical correction accuracy that can be achieved for sample rates of interest. The filter requires only seven instruction cycles per sample on the TMS320 DSPs. With a 200-ns instruction cycle, nine instructions per sample represents an overhead factor of 1.4% and 1.7% for sampling rates of 8000 Hz and 9600 Hz, respectively. This correction will add a slight amount of group delay at the upper edge of the 300—3000-Hz band.

(sin x)/x roll-off for a zero-order hold function

The (sin x)/x roll-off for the AIC DAC zero-order hold function at a band-edge frequency of 3000 Hz for the various sampling rates is shown in the table below.

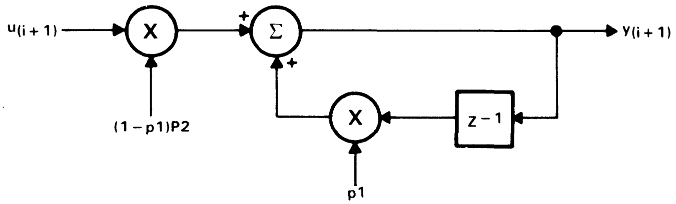
TABLE 3. (sin x)/x ROLL-OFF

f_s (Hz)	$20 \log \frac{\sin \pi f/f_s}{\pi f/f_s}$ (f = 3000 Hz) (dB)
7200	−2.64
8000	−2.11
9600	−1.44
14400	−0.63
19200	−0.35

Note that the actual AIC (sin x)/x roll-off will be slightly less than the above figures, because the AIC has less than a 100% duty cycle hold interval.

correction filter

To compensate for the (sin x)/x roll-off of the AIC, a first-order correction filter shown below, is recommended.



The difference equation for this correction filter is:

$$y_{i+1} = p2(1-p1)\,(u_{i+1}) + p1\, y_i$$

where the constant p1 determines the pole locations.

The resulting squared magnitude transfer function is:

$$|H(f)|^2 = \frac{p2^2\,(1-p1)^2}{1 - 2p1\cos(2\,\pi\, f/f_s) + p1^2}$$

correction results

Table 4 below shows the optimum p values and the corresponding correction results for 8000-Hz and 9600-Hz sampling rates.

TABLE 4

f (Hz)	ERROR (dB)
	f_s = 8000 Hz
	p1 = −0.14813
	p2 = 0.9888
		ERROR (dB)
		f_s = 9600 Hz
		p1 = −0.1307
		p2 = 0.9951

300	−0.099	−0.043
600	−0.089	−0.043
900	−0.054	0
1200	−0.002	0
1500	0.041	0
1800	0.079	0.043
2100	0.100	0.043
2400	0.091	0.043
2700	−0.043	0
3000	−0.102	−0.043

TMS320 software requirements

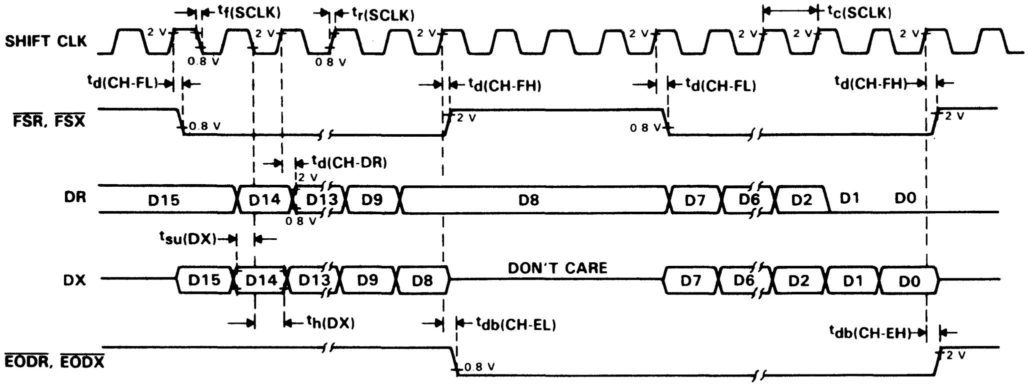
The digital correction filter equation can be written in state variable form as follows:

$$Y = k1Y + k2U$$

where k1 equals p1 (from the preceding page), k2 equals (1 − p1)p2 (from the preceding page), Y is the filter state, and U is the next I/O sample. The coefficients k1 and k2 must be represented as 16-bit integers. The SACH instruction (with the proper shift) will yield the correct result. With the assumption that the TMS320 processor page pointer and memory configuration are properly initialized, the equation can be executed in seven instructions or seven cycles with the following program:

```
ZAC
LT K2
MPY U
LTA K1
MPY Y
APAC
SACH (dma), (shift)
```

byte-mode timing



word-mode timing

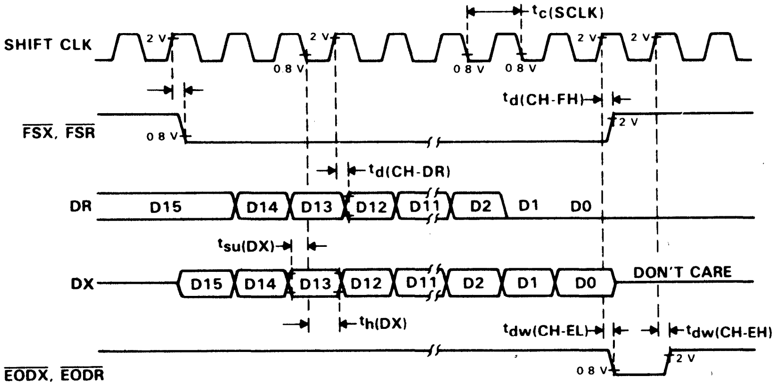


FIGURE 5. SERIAL PORT TIMING

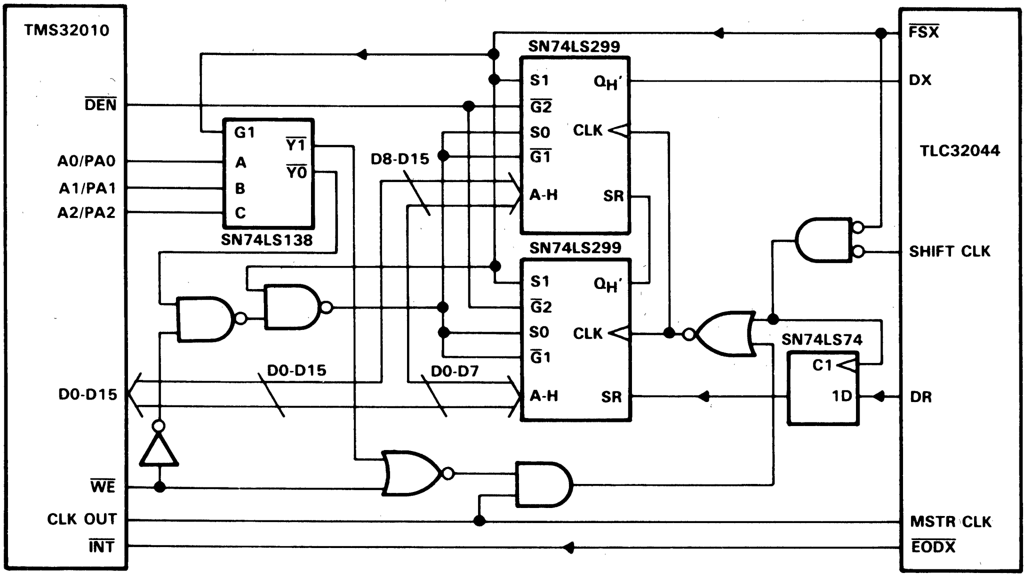


FIGURE 6. TMS32010/TMS320C15-TLC32044 INTERFACE CIRCUIT

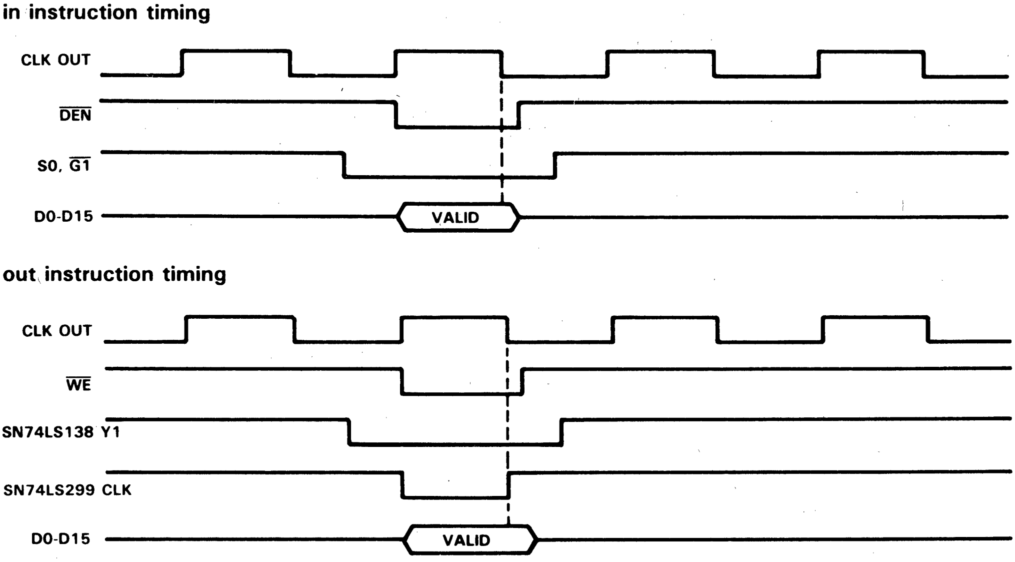


FIGURE 7. TMS32010/TMS320C15-TLC32044 INTERFACE TIMING

TYPICAL CHARACTERISTICS

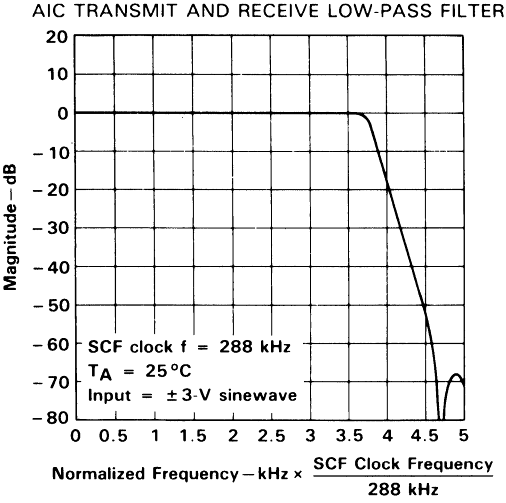


FIGURE 8

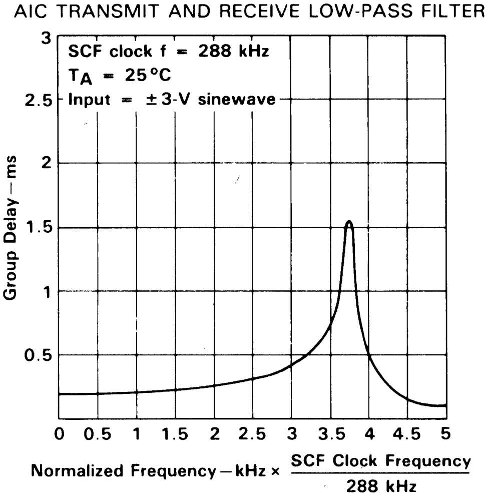


FIGURE 9

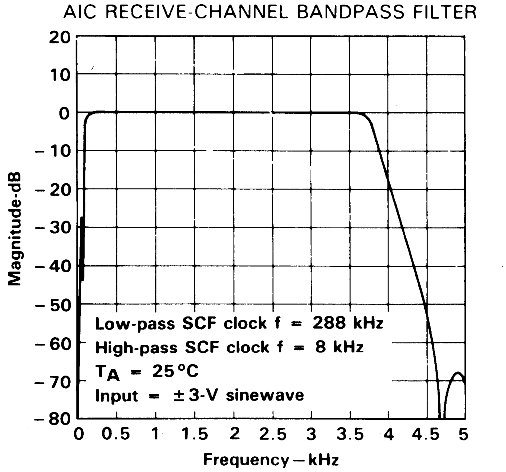


FIGURE 10

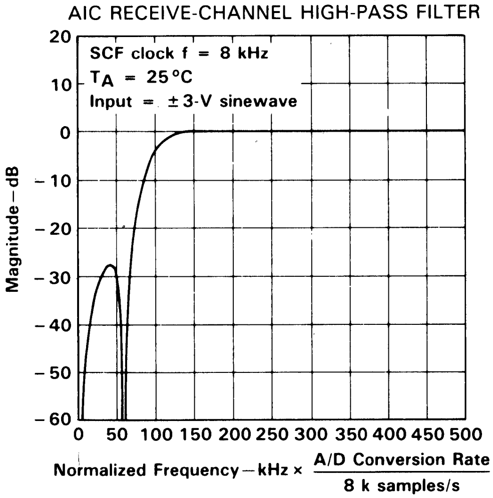


FIGURE 11

TYPICAL CHARACTERISTICS

AIC RECEIVE CHANNEL BANDPASS FILTER

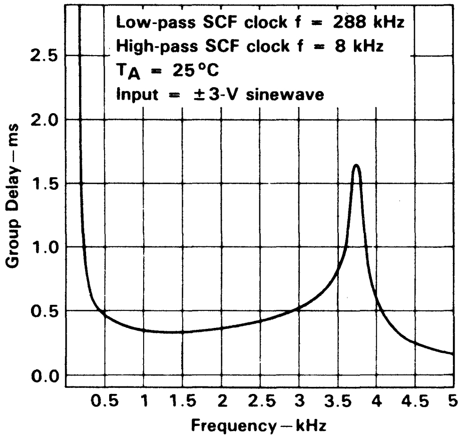


FIGURE 12

AIC (SIN X)/X CORRECTION FILTER

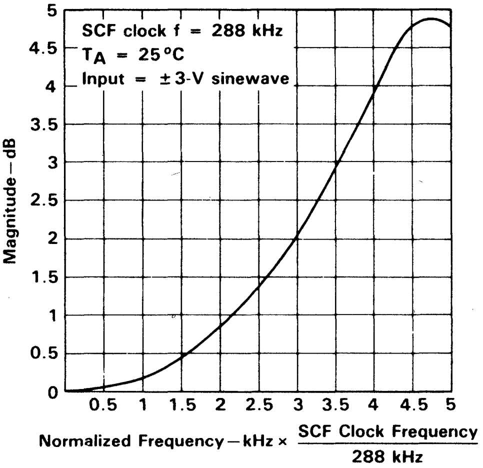


FIGURE 13

(SIN X)/X CORRECTION FILTER

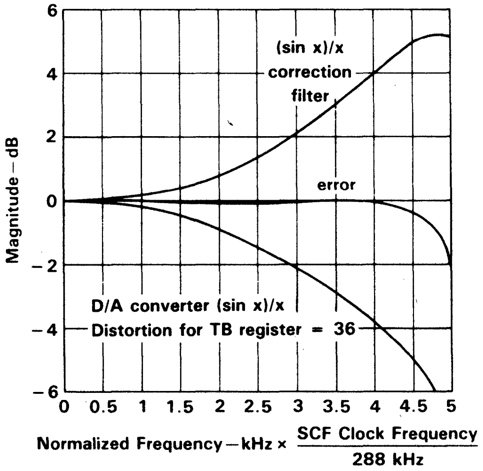


FIGURE 14

A/D SIGNAL-TO-DISTORTION RATIO
vs
INPUT SIGNAL

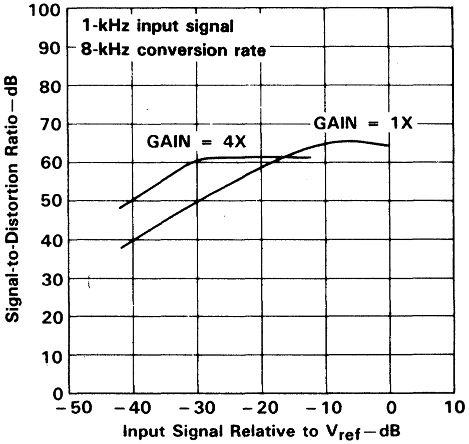


FIGURE 15

TYPICAL CHARACTERISTICS

A/D GAIN TRACKING
(GAIN RELATIVE TO GAIN AT 0 dB INPUT SIGNAL)

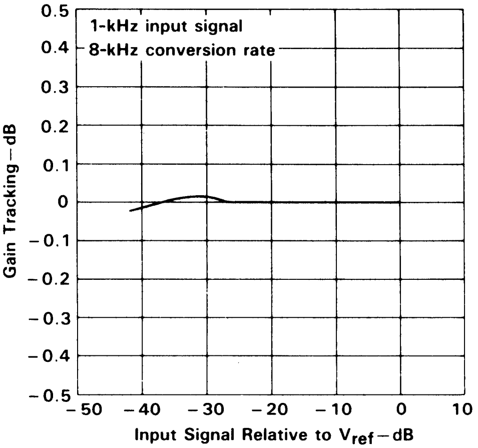


FIGURE 16

D/A CONVERTER SIGNAL-TO-DISTORTION RATIO
VS
INPUT SIGNAL LEVEL

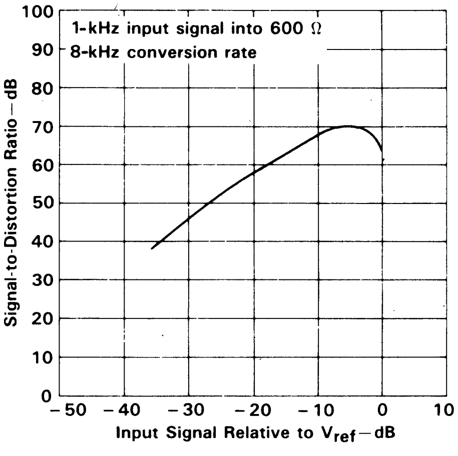


FIGURE 17

TYPICAL CHARACTERISTICS

D/A GAIN TRACKING
(GAIN RELATIVE TO GAIN
AT 0 dB INPUT SIGNAL LEVEL)

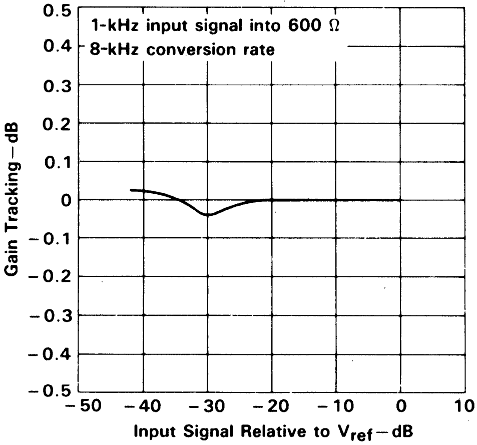


FIGURE 18

A/D SECOND HARMONIC DISTORTION
vs
INPUT SIGNAL

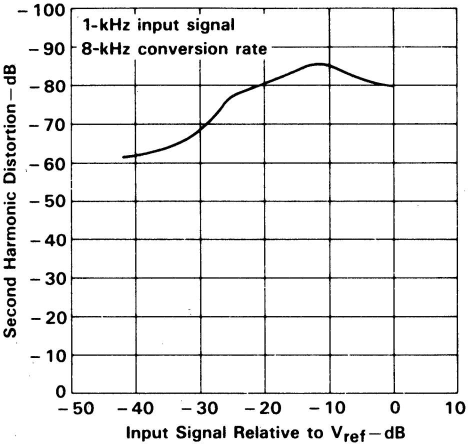


FIGURE 19

D/A SECOND HARMONIC DISTORTION
vs
INPUT SIGNAL

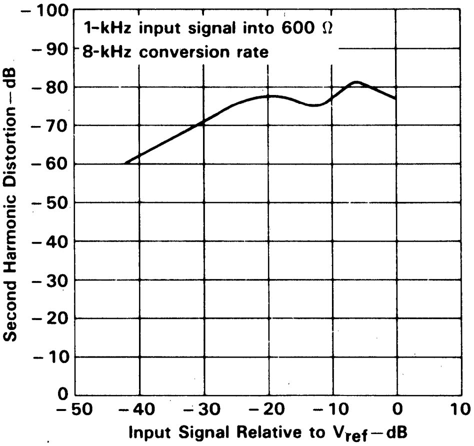


FIGURE 20

TYPICAL CHARACTERISTICS

A/D THIRD HARMONIC DISTORTION
vs
INPUT SIGNAL

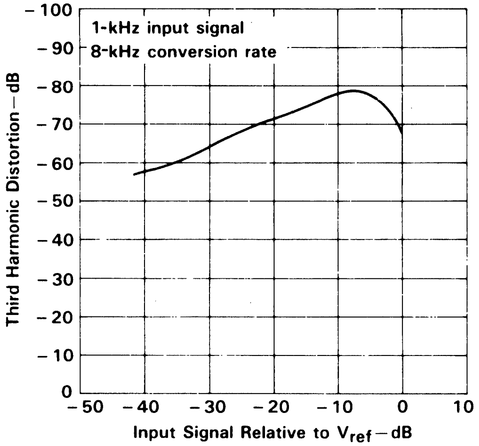


FIGURE 21

D/A THIRD HARMONIC DISTORTION
vs
INPUT SIGNAL

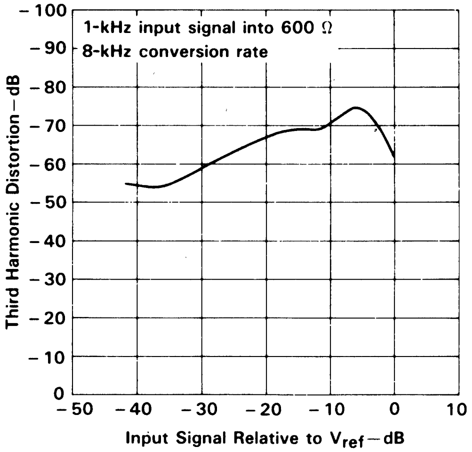
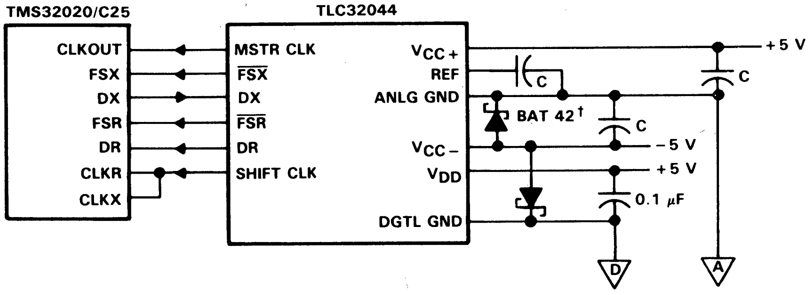


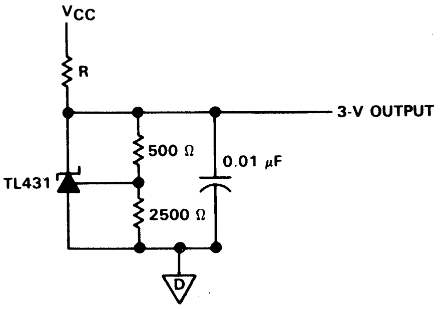
FIGURE 22

TYPICAL APPLICATION INFORMATION



C = 0.2 μF, CERAMIC

FIGURE 23. AIC INTERFACE TO THE TMS32020/C25 SHOWING DECOUPLING CAPACITORS AND SCHOTTKY DIODE†



FOR: V_{CC} = 12 V, R = 7200 Ω
V_{CC} = 10 V, R = 5600 Ω
V_{CC} = 5 V, R = 1600 Ω

FIGURE 24. EXTERNAL REFERENCE CIRCUIT FOR TLC32044

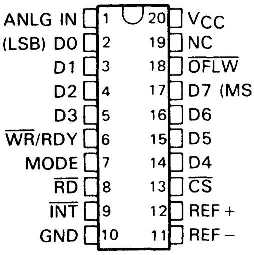
†Thomson Semiconductors

TLC0820A, TLC0820B, ADC0820B, ADC0820C
Advanced LinCMOS™ HIGH-SPEED 8-BIT ANALOG-TO-DIGITAL CONVERTERS USING MODIFIED "FLASH" TECHNIQUES

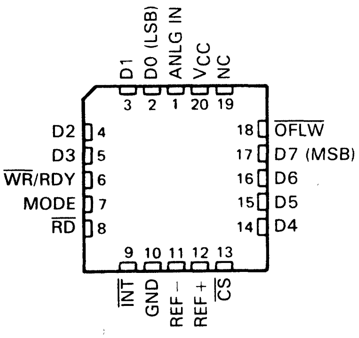
D2873, SEPTEMBER 1986—REVISED FEBRUARY 1989

- **Advanced LinCMOS™ Silicon-Gate Technology**
- **8-Bit Resolution**
- **Differential Reference Inputs**
- **Parallel Microprocessor Interface**
- **Conversion and Access Time Over Temperature Range**
  Write-Read Mode . . . 1.18 μs and 1.92 μs
  Read Mode . . . 2.5 μs Max
- **No External Clock or Oscillator Components Required**
- **On-Chip Track-and-Hold**
- **Low Power Consumption . . . 50 mW Typ**
- **Single 5-V Supply**
- **TLC0820B is Direct Replacement for National Semiconductor ADC0820B/BC and Analog Devices AD7820L/C/U; TLC0820A is Direct Replacement for National Semiconductor ADC0820C/CC and Analog Devices AD7820K/B/T**

ALL TYPES . . . DW OR N PACKAGE
TLC0820_M . . . J PACKAGE
(TOP VIEW)



TLC0820_M . . . FK PACKAGE
TLC0820_I, TLC0820_C . . . FN PACKAGE
ADC0820_CI, ADC0820_C . . . FN PACKAGE
(TOP VIEW)



NC—No internal connection

description

The TLC0820A, TLC0820B, ADC0820B, and ADC0820C are Advanced LinCMOS™ 8-bit analog-to-digital converters each consisting of two 4-bit "flash" converters, a 4-bit digital-to-analog converter, a summing (error) amplifier, control logic, and a result latch circuit. The modified "flash" technique allows low-power integrated circuitry to complete an 8-bit conversion in 1.18 μs over temperature. The on-chip track-and-hold circuit has a 100 ns sample window and allows these devices to convert continuous analog signals having slew rates of up to 100 mV/μs without external sampling components. TTL-compatible three-state output drivers and two modes of operation allow interfacing to a variety of microprocessors. Detailed information on interfacing to most popular microprocessors is readily available from the factory.

The M-suffix devices are characterized for operation over the full military temperature range of −55°C to 125°C. The I-suffix devices are characterized for operation from −40°C to 85°C. The C-suffix devices are characterized for operation from 0°C to 70°C. See Available Options.

Advanced LinCMOS is a trademark of Texas Instruments Incorporated.

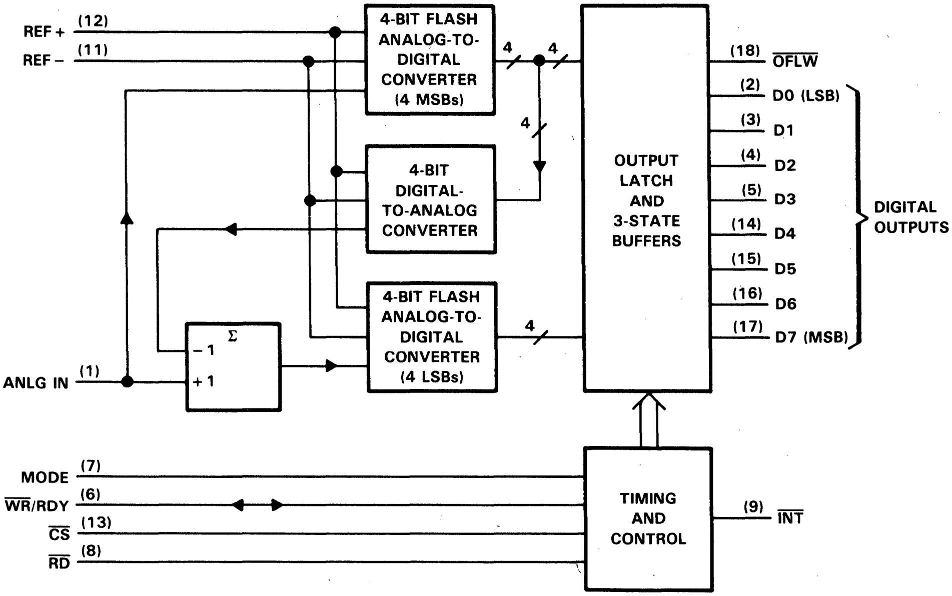
TEXAS INSTRUMENTS
POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

AVAILABLE OPTIONS

SYMBOLIZATION†		OPERATING	TOTAL
DEVICE	PACKAGE SUFFIX	TEMPERATURE RANGE	UNADJUSTED ERROR
TLC0820AC	DW, FN, N	0°C to 70°C	±1 LSB
TLC0820AI	DW, FN, N	−40°C to 85°C	±1 LSB
TLC0820AM	DW, FK, J, N	−55°C to 125°C	±1 LSB
TLC0820BC	DW, FN, N	0°C to 70°C	±0.5 LSB
TLC0820BI	DW, FN, N	−40°C to 85°C	±0.5 LSB
TLC0820BM	DW, FK, J, N	−55°C to 125°C	±0.5 LSB
ADC0820BC	DW, FN, N	0°C to 70°C	±0.5 LSB
ADC0820BCI	DW, FN, N	−40°C to 85°C	±0.5 LSB
ADC0820CC	DW, FN, N	0°C to 70°C	±1 LSB
ADC0820CCI	DW, FN, N	−40°C to 85°C	±1 LSB

†In many instances, these ICs may have both TLC0820 and ADC0820 labeling on the package.

functional block diagram



PIN		DESCRIPTION
NAME	NUMBER	
ANLG IN	1	Analog input
$\overline{\text{CS}}$	13	This input must be low in order for $\overline{\text{RD}}$ or $\overline{\text{WR}}$ to be recognized by the ADC.
D0	2	Three-state data output, bit 1 (LSB)
D1	3	Three-state data output, bit 2
D2	4	Three-state data output, bit 3
D3	5	Three-state data output, bit 4
D4	14	Three-state data output, bit 5
D5	15	Three-state data output, bit 6
D6	16	Three-state data output, bit 7
D7	17	Three-state data output, bit 8 (MSB)
GND	10	Ground
$\overline{\text{INT}}$	9	In the WRITE-READ mode, the interrupt output, $\overline{\text{INT}}$, going low indicates that the internal count-down delay time, $t_{d(int)}$, is complete and the data result is in the output latch. $t_{d(int)}$ is typically 800 ns starting after the rising edge of the $\overline{\text{WR}}$ input (see operating characteristics and Figure 3). If $\overline{\text{RD}}$ goes low prior to the end of $t_{d(int)}$, $\overline{\text{INT}}$ goes low at the end of t_{dRIL} and the conversion results are available sooner (see Figure 2). $\overline{\text{INT}}$ is reset by the rising edge of either $\overline{\text{RD}}$ or $\overline{\text{CS}}$.
MODE	7	Mode-selection input. It is internally tied to GND through a 50-μA current source, which acts like a pull-down resistor. READ mode: Occurs when this input is low. WRITE-READ mode: Occurs when this input is high.
NC	19	No internal connection
$\overline{\text{OFLW}}$	18	Normally the $\overline{\text{OFLW}}$ output is a logical high. However, if the analog input is higher than the V_{REF+}, $\overline{\text{OFLW}}$ will be low at the end of conversion. It can be used to cascade 2 or more devices to improve resolution (9 or 10-bits).
$\overline{\text{RD}}$	8	In the WRITE-READ mode with $\overline{\text{CS}}$ low, the 3-state data outputs D0 through D7 are activated when $\overline{\text{RD}}$ goes low. $\overline{\text{RD}}$ can also be used to increase the conversion speed by reading data prior to the end of the internal count-down delay time. As a result, the data transferred to the output latch is latched after the falling edge of $\overline{\text{RD}}$. In the READ mode with $\overline{\text{CS}}$ low, the conversion starts with $\overline{\text{RD}}$ going low. $\overline{\text{RD}}$ also enables the three-state data outputs upon completion of the conversion. The RDY output going into the high-impedance state and $\overline{\text{INT}}$ going low indicates completion of the conversion.
REF −	11	This input voltage is placed on the bottom of the resistor ladder.
REF +	12	This input voltage is placed on the top of the resistor ladder.
V_{CC}	20	Power supply voltage
$\overline{\text{WR}}$/RDY	6	In the WRITE-READ mode with $\overline{\text{CS}}$ low, the conversion is started on the falling edge of the $\overline{\text{WR}}$ input signal. The result of the conversion is strobed into the output latch after the internal count-down delay time, $t_{d(int)}$, provided that the $\overline{\text{RD}}$ input does not go low prior to this time. $t_{d(int)}$ is approximately 800 ns. In the READ mode, RDY (an open-drain output) will go low after the falling edge of $\overline{\text{CS}}$, and will go into the high-impedance state when the conversion is strobed into the output latch. It is used to simplify the interface to a microprocessor system.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

	TLC0820_M	TLC0820_I ADC0820_CI	TLC0820_C ADC0820_C	UNIT
Supply voltage, V_{CC} (see Note 1)	10	10	10	V
Input voltage range, all inputs (see Note 1)	−0.2 to V_{CC}+0.2	−0.2 to V_{CC}+0.2	−0.2 to V_{CC}+0.2	V
Output voltage range, all outputs (see Note 1)	−0.2 to V_{CC}+0.2	−0.2 to V_{CC}+0.2	−0.2 to V_{CC}+0.2	V
Operating free-air temperature range	−55 to 125	−40 to 85	0 to 70	°C
Storage temperature range	−65 to 150	−65 to 150	−65 to 150	°C
Case temperature for 60 seconds: FK package	260			°C
Case temperature for 10 seconds: FN package		260	260	°C
Lead temperature 1,6 mm (1/16 inch) from case for 60 seconds: J package	300			°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds: DW or N package	260	260	260	°C

NOTE 1: All voltages are with respect to network ground terminal, pin 10.

recommended operating conditions

			TLC0820_M			TLC0820_I ADC0820_CI			TLC0820_C ADC0820_C			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}			4.5	5	8	4.5	5	8	4.5	5	8	V
Analog input voltage			−0.1		V_{CC}+0.1	−0.1		V_{CC}+0.1	−0.1		V_{CC}+0.1	V
Positive reference voltage, V_{REF+}			V_{REF-}		V_{CC}	V_{REF-}		V_{CC}	V_{REF-}		V_{CC}	V
Negative reference voltage, V_{REF-}			GND		V_{REF+}	GND		V_{REF+}	GND		V_{REF+}	V
High-level input voltage, V_{IH}	V_{CC} = 4.75 V to 5.25 V	$\overline{CS}$, $\overline{WR}$/RDY, $\overline{RD}$	2			2			2			V
		MODE	3.5			3.5			3.5			
Low-level input voltage, V_{IL}	V_{CC} = 4.75 V to 5.25 V	$\overline{CS}$, $\overline{WR}$/RDY, $\overline{RD}$			0.8			0.8			0.8	V
		MODE			1.5			1.5			1.5	
Delay to next conversion, $t_{d(NC)}$ (see Figures 1, 2, 3, and 4)			500			500			500			ns
Delay time from $\overline{WR}$ to $\overline{RD}$ in write-read mode, t_{dWR} (see Figure 2)			0.4			0.4			0.4			μs
Write-pulse duration in write-read mode, t_{wW} (see Figures 2, 3, and 4)			0.5		50	0.5		50	0.5		50	μs
Operating free-air temperature, T_A			−55		125	−40		85	0		70	°C

electrical characteristics at specified operating free-air temperature, V_{CC} = 5 V (unless otherwise noted)

PARAMETER		TEST CONDITONS		MIN	TYP†	MAX	UNIT
V_{OH} High-level output voltage	Any D, $\overline{INT}$, or $\overline{OFLW}$	V_{CC} = 4.75 V, I_{OH} = −360 μA	Full range	2.4			V
		V_{CC} = 4.75 V, I_{OH} = −10 μA	Full range	4.5			
			25°C	4.6			
V_{OL} Low-level output voltage	Any D, $\overline{OFLW}$, $\overline{INT}$, or $\overline{WR}$/RDY	V_{CC} = 5.25 V, I_{OL} = 1.6 mA	Full range			0.4	V
			25°C			0.34	
I_{IH} High-level input current	$\overline{CS}$ or $\overline{RD}$	V_{IH} = 5 V	Full range		0.005	1	μA
	$\overline{WR}$/RDY		Full range			3	
			25°C		0.1	0.3	
	MODE		Full range			200	
			25°C		50	170	
I_{IL} Low-level input current	$\overline{CS}$, $\overline{WR}$/RDY, $\overline{RD}$, or MODE	V_{IL} = 0	Full range		−0.005	−1	μA
I_{OZ} Off-state (high-impedance state) output current	Any D or $\overline{WR}$/RDY	V_O = 5 V	Full range			3	μA
			25°C		0.1	0.3	
		V_O = 0	Full range			−3	
			25°C		−0.1	−0.3	
I_I Analog input current		$\overline{CS}$ at 5 V, V_I = 5 V	Full range			3	μA
			25°C			0.3	
		$\overline{CS}$ at 5 V, V_I = 0	Full range			−3	
			25°C			−0.3	
I_{OS} Short-circuit output current	Any D, $\overline{OFLW}$, $\overline{INT}$, or $\overline{WR}$/RDY	V_O = 5 V	Full range	7			mA
			25°C	8.4	14		
	Any D or $\overline{OFLW}$	V_O = 0	Full range	−6			
			25°C	−7.2	−12		
	$\overline{INT}$		Full range	−4.5			
			25°C	−5.3	−9		
R_{ref} Reference resistance			Full range	1.25		6	kΩ
			25°C	1.4	2.3	5.3	
I_{CC} Supply current		$\overline{CS}$, $\overline{WR}$/RDY, and $\overline{RD}$ at 0 V	Full range			15	mA
			25°C		7.5	13	
C_i Input capacitance	Any digital		Full range		5		pF
	ANLG IN				45		
C_o Output capacitance	Any digital		Full range			5	pF

†All typical values are at T_A = 25°C.

operating characteristics, V_{CC} = 5 V, V_{REF+} = 5 V, V_{REF-} = 0, t_r = t_f = 20 ns, T_A = 25°C (unless otherwise noted)

PARAMETER		TEST CONDITIONS		TLC0820B ADC0820B			TLC0820A ADC0820C			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
k_{SVS}	Supply voltage sensitivity	V_{CC} = 5 V ± 5%, T_A = MIN to MAX			±1/16	±1/4		±1/16	±1/4	LSB
	Total unadjusted error[†]	MODE pin at 0 V, T_A = MIN to MAX				1/2			1	LSB
t_{convR}	Read mode conversion time	MODE pin at 0 V, See Figure 1			1.6	2.5		1.6	2.5	μs
$t_{d(int)}$	Internal count-down delay time	MODE pin at 5 V, C_L = 50 pF, See Figures 3 and 4			800	1300		800	1300	ns
t_{aR}	Access time from $\overline{RD}$↓	MODE pin at 0 V, See Figure 1			t_{convR} +20	t_{convR} +50		t_{convR} +20	t_{convR} +50	ns
t_{aR1}	Access time from $\overline{RD}$↓	MODE pin at 5 V, $t_{dWR} < t_{d(int)}$, See Figure 2	C_L = 15 pF		190	280		190	280	ns
			C_L = 100 pF		210	320		210	320	
t_{aR2}	Access time from $\overline{RD}$↓	MODE pin at 5 V, $t_{dWR} > t_{d(int)}$ See Figure 3	C_L = 15 pF		70	120		70	120	ns
			C_L = 100 pF		90	150		90	150	
t_{aINT}	Access time from $\overline{INT}$↓	MODE pin at 5 V, See Figure 4			20	50		20	50	ns
t_{dis}	Disable time from $\overline{RD}$↑	R_L = 1 kΩ, C_L = 10 pF, See Figures 1, 2, 3, and 5			70	95		70	95	ns
t_{dRDY}	Delay time from $\overline{CS}$↓ to RDY↓	MODE pin at 0 V, C_L = 50 pF, See Figure 1			50	100		50	100	ns
t_{dRIH}	Delay time from $\overline{RD}$↑ to $\overline{INT}$↑	C_L = 50 pF, See Figures 1, 2, and 3			125	225		125	225	ns
t_{dRIL}	Delay time from $\overline{RD}$↓ to $\overline{INT}$↓	MODE pin at 5 V, $t_{dWR} < t_{d(int)}$, See Figure 2			200	290		200	290	ns
t_{dWIH}	Delay time from $\overline{WR}$↑ to $\overline{INT}$↑	MODE pin at 5 V, C_L = 50 pF, See Figure 4			175	270		175	270	ns
	Slew rate tracking				0.1			0.1		V/μs

[†] Total unadjusted error includes offset, full-scale, and linearity errors.

PARAMETER MEASUREMENT INFORMATION

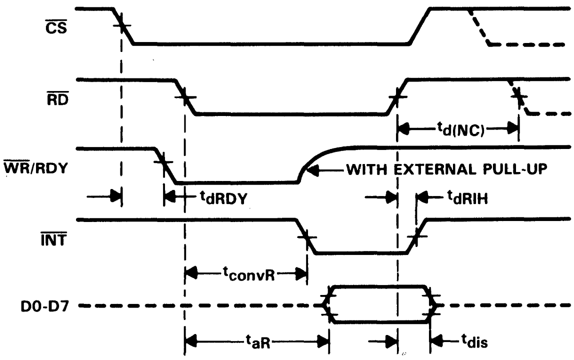


FIGURE 1. READ MODE WAVEFORMS (MODE PIN LOW)

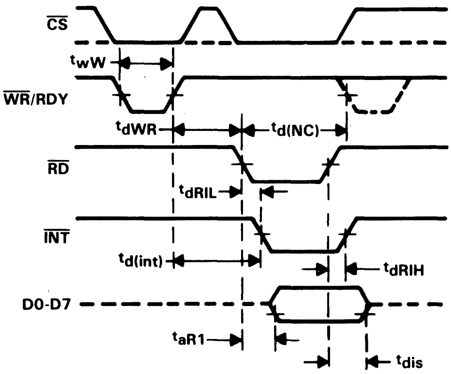


FIGURE 2. WRITE-READ MODE WAVEFORMS
[MODE PIN HIGH AND $t_{dWR} < t_{d(int)}$]

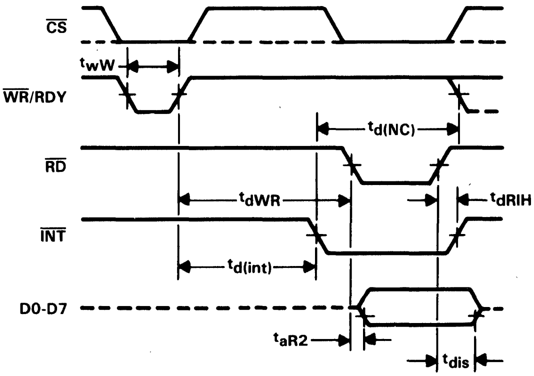


FIGURE 3. WRITE-READ WAVEFORMS
[MODE PIN HIGH AND $t_{dWR} > t_{d(int)}$]

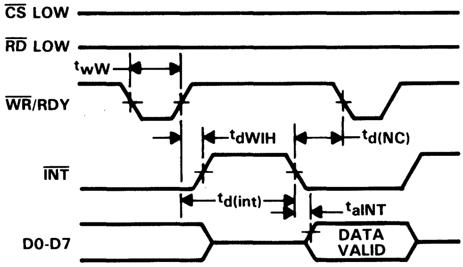
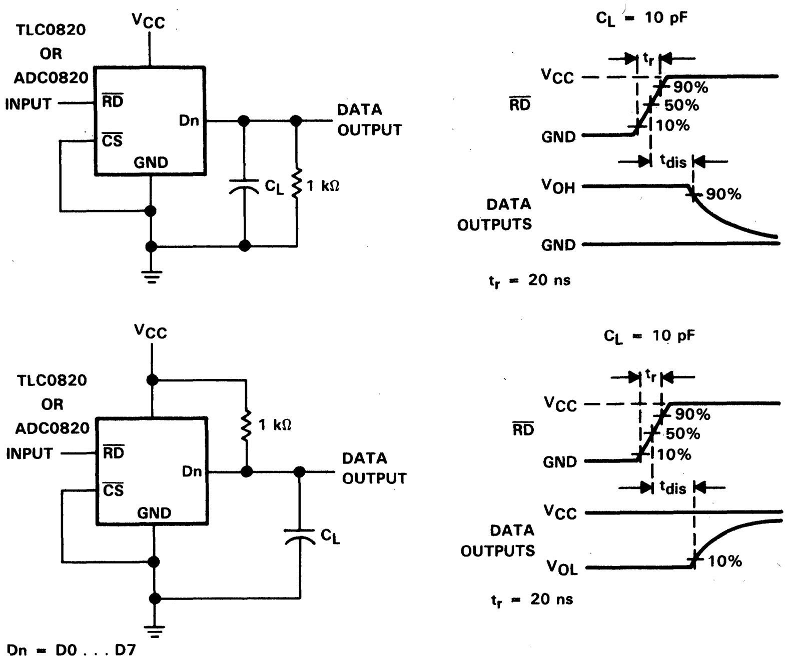


FIGURE 4. WRITE-READ MODE WAVEFORMS
(STAND-ALONE OPERATION, MODE PIN HIGH, AND RD LOW)

PARAMETER MEASUREMENT INFORMATION



TEST CIRCUIT

VOLTAGE WAVEFORMS

FIGURE 5. TEST CIRCUIT AND VOLTAGE WAVEFORMS

PRINCIPLES OF OPERATION

The TLC0820A, TLC0820B, ADC0820B and ADC0820C each employ a combination of "sampled-data" comparator techniques and "flash" techniques common to many high-speed converters. Two 4-bit "flash" analog-to-digital conversions are used to give a full 8-bit output.

The recommended analog input voltage range for conversion is −0.1 V to V_{CC}+0.1 V. Analog input signals that are less than V_{REF-} + ½ LSB or greater than V_{REF+} − ½ LSB convert to 00000000 or 11111111 respectively. The reference inputs are fully differential with common-mode limits defined by the supply rails. The reference input values define the full-scale range of the analog input. This allows the gain of the ADC to be varied for ratiometric conversion by changing the V_{REF+} and V_{REF-} voltages.

The device operates in two modes, read (only) and write-read, which are selected by the MODE pin (pin 7). The converter is set to the read (only) mode when pin 7 is low. In the read mode, the $\overline{\text{WR}}$/RDY pin is used as an output and is referred to as the "ready" pin. In this mode, a low on the "ready" pin while $\overline{\text{CS}}$ is low indicates that the device is busy. Conversion starts on the falling edge of $\overline{\text{RD}}$ and is completed no more than 2.5 μs later when $\overline{\text{INT}}$ falls and the "ready" pin returns to a high-impedance state. Data outputs also change from high-impedance to active states at this time. After the data is read, $\overline{\text{RD}}$ is taken high, $\overline{\text{INT}}$ returns high, and the data outputs return to their high-impedance states.

The converter is set to the write-read mode when pin 7 is high and $\overline{\text{WR}}$/RDY is referred to as the "write" pin. Taking $\overline{\text{CS}}$ and the "write" pin low selects the converter and initiates measurement of the input signal. Approximately 600 ns after the "write" pin returns high, the conversion is completed. Conversion starts on the rising edge of $\overline{\text{WR}}$/RDY in the write-read mode.

The high-order 4-bit "flash" ADC measures the input by means of 16 comparators operating simultaneously. A high precision 4-bit DAC then generates a discrete analog voltage from the result of that conversion. After a time delay, a second bank of comparators does a low-order conversion on the analog difference between the input level and the high-order DAC output. The results from each of these conversions enter an 8-bit latch and are output to the three-state buffers on the falling edge of $\overline{\text{RD}}$.

TYPICAL APPLICATION DATA

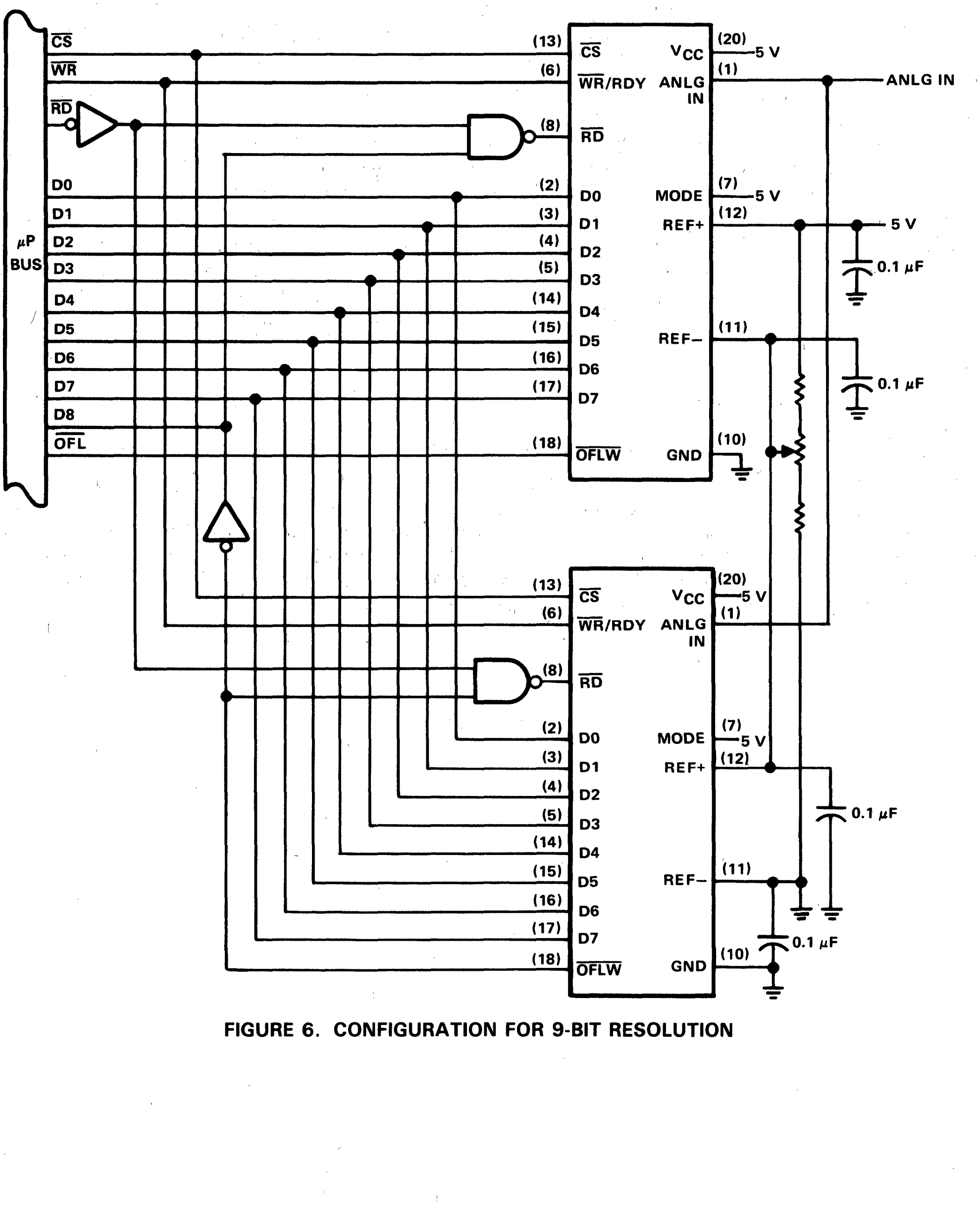


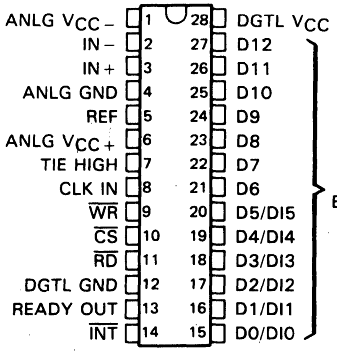
FIGURE 6. CONFIGURATION FOR 9-BIT RESOLUTION

TLC1225A, TLC1225B
SELF-CALIBRATING 12-BIT-PLUS-SIGN UNIPOLAR OR BIPOLAR ANALOG-TO-DIGITAL CONVERTERS

D2982, FEBRUARY 1987—REVISED JANUARY 1989

- **Advanced LinCMOS™ Technology**
- **Self-Calibration Eliminates Expensive Trimming at Factory and Offset Adjustment in the Field**
- **12-Bit Plus Sign Bipolar or 12-Bit Unipolar**
- **±1/2 and ±1 LSB Linearity Error in Unipolar Configuration**
- **10 µs Conversion Time (clock = 2.6 MHz)**
- **Compatible with All Microprocessors**
- **True Differential Analog Voltage Inputs**
- **0 to 5 V Analog Voltage Range with Single 5-V Supply (Unipolar Configuration)**
- **−5 V to 5 V Analog Voltage Range with ±5-V Supplies (Bipolar Configuration)**
- **Low Power . . . 25 mW Maximum**

J OR N DUAL-IN-LINE PACKAGE
(TOP VIEW)



description

The TLC1225A and TLC1225B converters are manufactured with Texas Instruments highly efficient Advanced LinCMOS™ technology. Either of the TLC1225A or TLC1225B CMOS analog-to-digital converters can be operated as a unipolar or bipolar converter. A unipolar input(0 to 5 V) can be accommodated with a single 5-V supply; a bipolar input (−5 V to 5 V) requires the addition of a 5-V negative supply. Conversion is performed via the successive-approximation method. The TLC1225A and TLC1225B output the converted data in a parallel word and interfaces directly to a 16-bit data bus. Negative numbers are given in the two's complement data format. All digital signals are fully TTL and CMOS compatible.

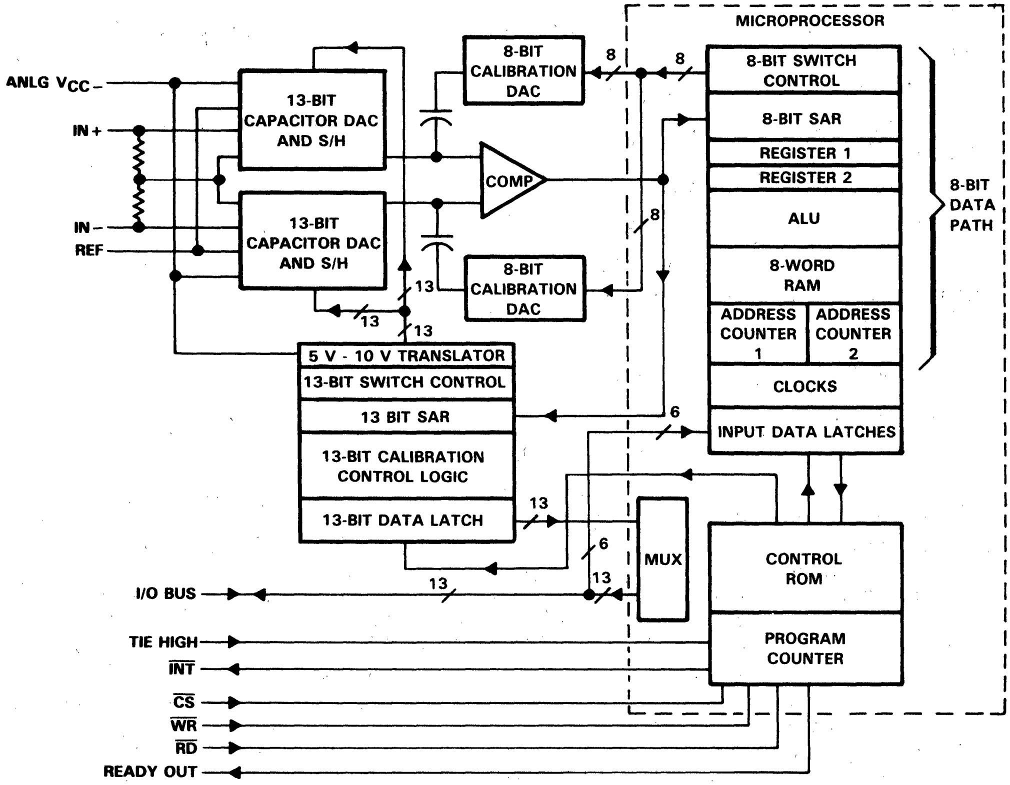
These converters utilize a self-calibration technique by which seven of the internal capacitors in the capacitive ladder of the A/D conversion circuitry can be automatically calibrated. The internal capacitors are calibrated during a nonconversion, capacitor-calibrate cycle in which all seven of the internal capacitors are calibrated at the same time. A conversion requires only 10 µs (2.6 MHz clock) after the nonconversion, capacitor-calibrating cycle has been completed. The calibration or conversion cycle may be initiated at any time by issuing the proper command word to the data bus. The self-calibrating technique eliminates the need for expensive trimming of thin-film resistors at the factory and provide excellent performance at low cost.

The TLC1225AM and TLC1225BM are characterized for operation over the full military temperature range of −55°C to 125°C. The TLC1225AI and TLC1225BI are characterized for operation from −40°C to 85°C.

Advanced LinCMOS™ is a trademark of Texas Instruments Incorporated

PRODUCT PREVIEW documents contain information on products in the formative or design phase of development. Characteristic data and other specifications are design goals. Texas Instruments reserves the right to change or discontinue these products without notice.

functional block diagram



operation description

calibration of comparator offset

The following actions are performed to calibrate the comparator offset:

1. The IN+ and IN− inputs are internally shorted together in order that the comparator input is zero. A course comparator offset calibration is performed by storing the offset voltages of the interconnecting comparator stages on the coupling capacitors that connect these stages. Refer to Figure 1. The storage of offset voltages is accomplished by closing all switches and then opening switches A and A′, then switches B and B′, and then C and C′. This process continues until all interconnecting stages of the comparator are calibrated. After this action, some of the comparator offset still remains uncalibrated.

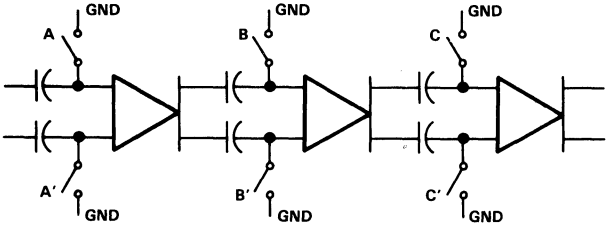


FIGURE 1

2. An A/D conversion is done on the remaining offset with the 8-bit calibration DACs and 8-bit SAR and the result is stored in the RAM.

capacitor calibration of the ADC's capacitive ladder

The following actions are performed to calibrate capacitors in the 13-bit DACs that comprise the ADC's capacitive ladder:

1. The IN+ and IN− inputs are internally disconnected from the 13-bit capacitive DACs.
2. The most significant bit (MSB) capacitor is tied to REF, while the rest of the ladder capacitors are tied to GND. The A/D conversion result for the remaining comparator offset, obtained in Step 2 above, is retrieved from the RAM and is input to the 8-bit DACs.
3. Step 1 of the Calibration of Comparator Offset sequence is performed. The 8-bit DAC input is returned to zero and the remaining comparator offset is then subtracted. Thus, the comparator offset is completely corrected.
4. Now the MSB capacitor is tied to GND, while the rest of the ladder capacitors, C_X, are tied to REF. An MSB capacitor voltage error (see Figure 2) on the comparator output will occur if the MSB capacitor does not equal the sum of the other capacitors in the capacitive ladder. This error voltage is converted to an 8-bit word from which a capacitor error is computed and stored in the RAM.
5. The capacitor voltage error for the next most significant capacitor is calibrated by keeping the MSB capacitor grounded and then performing the above Steps 1-4 while using the next most significant capacitor in lieu of the MSB capacitor. The seven most significant capacitors can be calibrated in this manner.

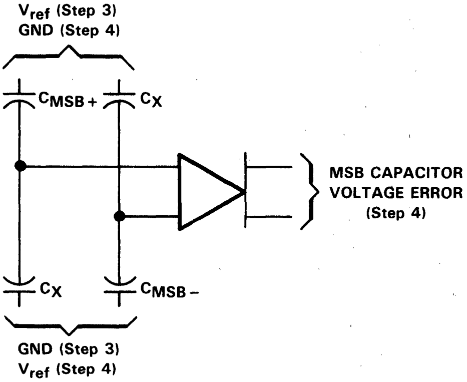


FIGURE 2

analog-to-digital conversion

The following steps are performed in the analog-to-digital conversion process:

1. Step 1 of the Calibration of Comparator Offset Sequence is performed. The A/D conversion result for the remaining comparator offset, which was obtained in Step 2 of the Calibration of Comparator Offset, is retrieved from the RAM and is input to the 8-bit DACs. Thus the comparator offset is completely corrected.
2. IN+ and IN− are sampled onto the 13-bit capacitive ladders.
3. The 13-bit analog-to-digital conversion is performed. As the successive-approximation conversion proceeds successively through the seven most significant capacitors, the error for each of these capacitors is recovered from the RAM and accumulated in a register. This register controls the 8-bit DACs so the total accumulated error for these capacitors is subtracted out during the conversion process.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage (ANLG V_{CC+} and DGTL V_{CC}) (see Note 1)	15 V
Supply voltage, ANLG V_{CC-}	−15 V
Control and Clock input voltage range	−0.3 V to +15 V
Analog input (IN+, IN−) voltage range,	
V_{I+} and V_{I-}	ANLG V_{CC-} −0.3 V to ANLG V_{CC+} +0.3 V
Reference voltage range, V_{ref}	−0.3 V to ANLG V_{CC+} +0.3 V
Pin 7 voltage range, V_{OS}	−0.3 V to ANLG V_{CC+} +0.3 V
Output voltage range	−0.3 V to DGTL V_{CC} +0.3 V
Input current (per pin)	±5 mA
Input current (per package)	±20 mA
Operating free-air temperature range:	
TLC1225AM, TL1225BM	−55°C to 125°C
TLC1225AI, TLC1225BI	−40°C to 85°C
Storage temperature range	−65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from the case for 60 seconds: J package	300°C
Lead temperature 1,6 mm (1/16 inch) from the case for 10 seconds: N package	260°C

NOTE 1: All analog voltages are referred to ANLG GND and all digital voltages are referred to DGTL GND.

recommended operating conditions

		MIN	MAX	UNIT
Supply voltage	ANLG V_{CC+}	4.5	6	V
	ANLG V_{CC-}	−5.5	ANLG GND	
	DGTL V_{CC}	4.5	6	
High-level input voltage, V_{IH}, all digital inputs except CLK IN (V_{CC} = 4.75 V to 5.25 V)		2		V
Low-level input voltage, V_{IL}, all digital inputs except CLK IN (V_{CC} = 4.75 V to 5.25 V)			0.8	V
Analog input voltage, V_{I+}, V_{I-}	Bipolar range	ANLG V_{CC-} − 0.05	ANLG V_{CC+} + 0.05	V
	Unipolar range	ANLG GND − 0.05	ANLG V_{CC+} + 0.05	
Pin 7 (TIE HIGH)		2		V
Clock input frequency, f_{clock}		0.3	2.6	MHz
Clock duty cycle		40%	60%	
Pulse duration, $\overline{CS}$ and $\overline{WR}$ both low, t_w ($\overline{CS} \cdot \overline{WR}$)		50		ns
Setup time before $\overline{WR}$↑ or $\overline{CS}$↑, t_{su}			50	ns
Hold time after $\overline{WR}$↑ or $\overline{CS}$↑, t_h			50	ns
Operating free-air temperature, T_A	TLC1225AM, TLC1225BM	−55	125	°C
	TLC1225AI, TLC1225BI	−40	85	

electrical characteristics over recommended operating free-air temperature range, ANLG V_{CC+} = DGTL V_{CC} = V_{ref} = 5 V, ANLG V_{CC-} = −5 V (for bipolar input range), ANLG V_{CC-} = ANLG GND (for unipolar input range) (unless otherwise noted) (see Note 2)

PARAMETER		TEST CONDITIONS		MIN	MAX	UNIT
V_{OH}	High-level output voltage	DGTL V_{CC} = 4.75 V	I_O = −1.8 mA	2.4		V
			I_O = −50 µA	4.5		
V_{OL}	Low-level output voltage	DGTL V_{CC} = 4.75 V,	I_O = 8 mA		0.4	V
V_{T+}	Clock positive-going threshold voltage			2.7	3.5	V
V_{T-}	Clock negative-going threshold voltage			1.4	2.1	V
V_{hys}	Clock input hysteresis	V_{T+} min − V_{T-} max		0.6		V
		V_{T+} max − V_{T-} min			2.1	
r_{ref}	Input resistance, REF terminal			1	10	MΩ
I_{IH}	High-level input current	V_I = 5 V			1	µA
I_{IL}	Low-level input current	V_I = 0			−1	µA
I_{OZ}	High-impedance-state output leakage current	V_O = 0			−3	µA
		V_O = 5 V			3	
I_O	Output current	V_O = 0			−6	mA
		V_O = 5 V			8	
DGTL I_{CC}	Supply current from DGTL V_{CC}	f_{clk} = 2.6 MHz,	$\overline{CS}$ high		3	mA
ANLG I_{CC+}	Supply current from ANLG V_{CC+}	f_{clk} = 2.6 MHz,	$\overline{CS}$ high		3	mA
ANLG I_{CC-}	Supply current from ANLG V_{CC-}	f_{clk} = 2.6 MHz,	$\overline{CS}$ high		−3	mA

NOTE 2: Bipolar input range is defined as: V_{I+} = −5.05 V to 5.05 V, V_{I-} = −5.05 V to 5.05 V, and $|V_{I+} - V_{I-}| \leq 5.05$ V. The unipolar input voltage range is defined as: V_{I+} = −0.05 V to 5.05 V, V_{I-} = −0.05 V to 5.05 V, and $|V_{I+} - V_{I-}| \leq 5.05$ V.

operating characteristics over recommended operating free-air temperature range, ANLG V_{CC+} = DGTL V_{CC} = V_{ref} = 5 V, ANLG V_{CC-} = −5 V (for bipolar input range), ANLG V_{CC-} = ANLG GND (for unipolar input range), f_{clock} = 2.6 MHz (unless otherwise noted) (see Note 2)

PARAMETER			TEST CONDITIONS		MIN	MAX	UNIT
E_L	Linearity error		Unipolar input range	TLC1225A		±1	LSB
				TLC1225B		±0.5	
			Bipolar input range	TLC1225A		±2	
				TLC1225B		±1.5	
	Zero error					±0.5	LSB
	Adjusted positive and negative full-scale error (see Note 3)		Unipolar input range			±1	LSB
	Adjusted positive and negative full-scale error (see Note 4)		Bipolar input range			±1	LSB
	Temperature coefficient of gain					15	ppm/°C
	Temperature coefficient of offset point					1.5	ppm/°C
k_{SVS}	Supply voltage sensitivity	Zero error	ANLG V_{CC+} = 5 V ± 5%, ANLG V_{CC-} = −5 V ± 5%, DGTL V_{CC} = 5 V ± 5%			±0.75	LSB
		Positive and negative full-scale error				±0.75	
		Linearity error				±0.25	
t_c	Conversion time (1/f_{clk})					27	clock cycles
t_a	Access time (delay from falling edge of $\overline{\mathrm{CS}}\cdot\overline{\mathrm{RD}}$ to data output)		C_L = 100 pF			110	ns
t_{dis}	Disable time, output (delay from rising edge of $\overline{\mathrm{RD}}$ to high-impedance state		R_L = 10 kΩ,	C_L = 10 pF		60	ns
			R_L = 2 kΩ,	C_L = 100 pF		60	
$t_{d(READY)}$	$\overline{\mathrm{RD}}$ or $\overline{\mathrm{WR}}$ to READY OUT delay					140	ns
$t_{d(INT)}$	$\overline{\mathrm{RD}}$ or $\overline{\mathrm{WR}}$ to reset of $\overline{\mathrm{INT}}$ delay					400	ns

NOTES: 2. Bipolar input range is defined as: V_{I+} = −5.05 V to 5.05 V, V_{I-} = −5.05 V to 5.05 V, and $|V_{I+} - V_{I-}| \leq$ 5.05 V. The unipolar input voltage range is defined as: V_{I+} = −0.05 V to 5.05 V, V_{I-} = −0.05 V to 5.05 V, and $|V_{I+} - V_{I-}| \leq$ 5.05 V.

3. See the Positive and Negative Full-Scale Adjustment section, Unipolar Inputs.

4. See the Positive and Negative Full-Scale Adjustment section, Bipolar Inputs.

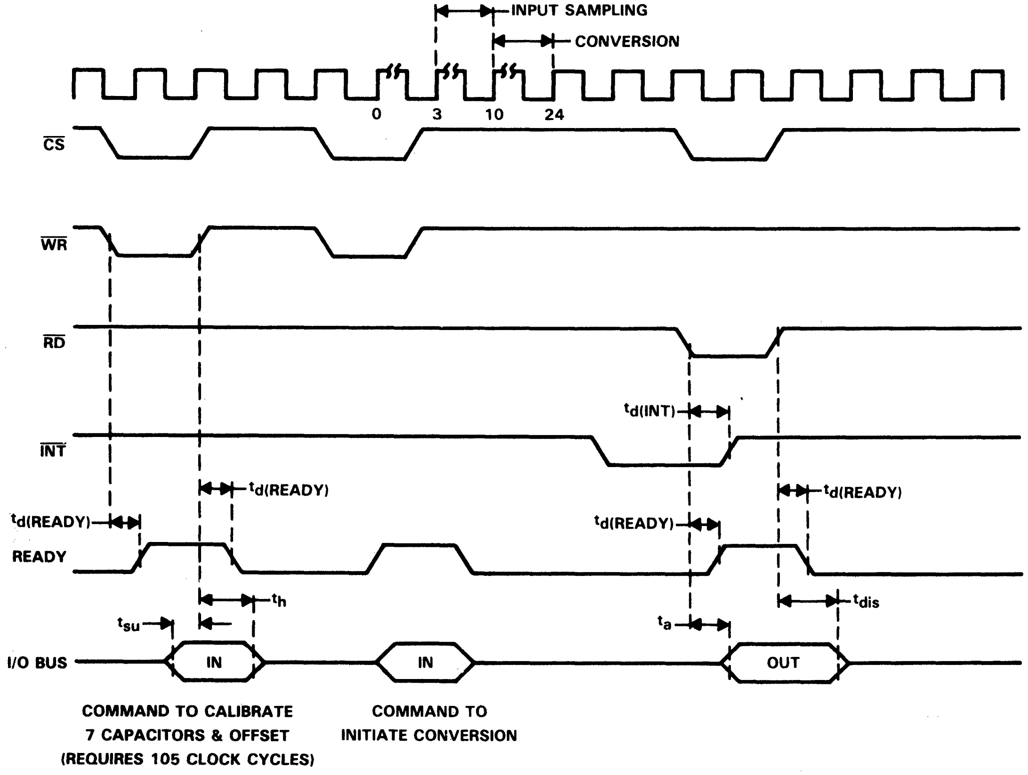


FIGURE 3. TIMING DIAGRAM

PARAMETER MEASUREMENT INFORMATION

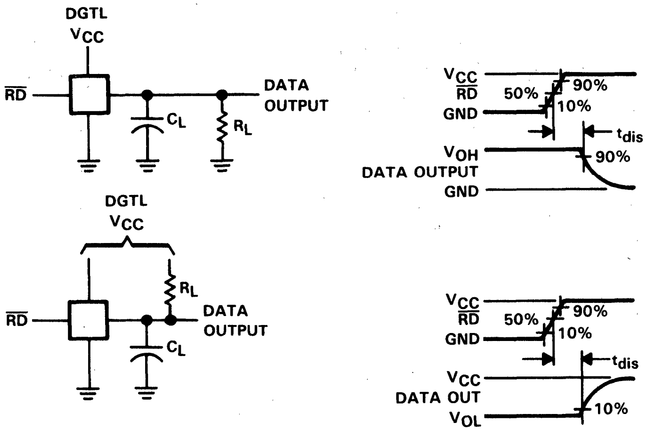


FIGURE 4. LOAD CIRCUITS AND WAVEFORMS

PRINCIPLES OF OPERATION

power-up calibration sequence

Power-Up calibration is not automatic and calibration is initiated by writing control words to the six least significant bits of the data bus. If addressed or initiated, conversion can begin after the first clock cycle. However, full A/D conversion accuracy is not established until after internal capacitor calibration.

conversion start sequence

The writing of the conversion command word to the six least significant bits of the data bus, when either $\overline{CS}$ or $\overline{WR}$ goes high, initiates the conversion sequence.

analog sampling sequence

Sampling of the input signal occurs during clock cycles 3 thru 10 of the conversion sequence.

completed A/D conversion

When $\overline{INT}$ goes low, conversion is complete and the A/D result can be read. A new conversion can begin immediately. The A/D conversion is complete at the end of clock cycle 27 of the conversion sequence.

aborting a conversion in process and beginning a new conversion

If a conversion is initiated while a conversion sequence is in process, the ongoing conversion will be aborted and a new conversion sequence will begin.

reading the conversion result

When both $\overline{CS}$ and $\overline{RD}$ go low, all 13 bits of conversion data are output to the I/O bus. The format of the output is extended sign with 2's complement, right justified data. For both unipolar and bipolar cases, the sign bit D12 is low if $V_{I+} - V_{I-}$ is positive and high if $V_{I+} - V_{I-}$ is negative.

general

reset INT

When reading the conversion data, the falling edge of the first low-going combination of $\overline{CS}$ and $\overline{RD}$ will reset $\overline{INT}$. The falling edge of the low-going combination of $\overline{CS}$ and $\overline{WR}$ will also reset $\overline{INT}$.

ready out

For high-speed microprocessors, READY OUT allows the TLC1225 to insert a wait state in the microprocessor's read or write cycle.

reference voltage (V_{ref})

This voltage defines the range for $|V_{I+} - V_{I-}|$. When $|V_{I+} - V_{I-}|$ equals V_{ref}, the highest conversion data value results. When $|V_{I+} - V_{I-}|$ equals 0, the conversion data value is zero. Thus, for a given input, the conversion data changes ratiometrically with changes in V_{ref}.

TIE HIGH

This pin is a digital input and should be tied high.

calibration and conversion considerations

Calibration of the internal capacitor and A/D conversion are two separate actions. Each action is independently initiated. A calibration command that calibrates all seven internal capacitors is normally issued before conversion. A conversion command then initiates the A/D conversion. Subsequent conversions can be performed by issuing additional conversion commands. The calibration and conversion commands are totally independent from one another and can be initiated in any order. Calibration and conversion commands require 105 and 27 clock cycles, respectively.

The calibrate and conversion commands are initiated by writing control words on the six least significant bits of the data bus. These control words are written into the IC when either $\overline{CS}$ or $\overline{WR}$ goes high. The initiation of these commands is illustrated in the Timing Diagram. The bit patterns for the commands are shown in Table 1.

TABLE 1. CONVERSION COMMANDS

COMMAND	$\overline{CS}$ + $\overline{WR}$	I/O BUS						REQUIRED NUMBER
		DI5	DI4	DI3	DI2	DI1	DI0	OF CLOCK CYCLES
Conversion	↑	H	L	X	X	X	L	27
Calibrate†	↑	L	X	L	L	L	L	105

†Calibration is lost when clock is stopped.

analog inputs

differential inputs provide common-mode rejection

The differential inputs reduce common-mode noise. Common-mode noise is noise common to both IN+ and IN− inputs, such as 60-Hz noise. There is no time interval between the sampling of the IN+ and IN− so these inputs are truly differential. Thus, no conversion errors result from a time interval between the sampling of the IN+ and IN− inputs.

input bypass capacitors

Input bypass capacitors may be used for noise filtering. However, the charge on these bypass capacitors will be depleted during the input sampling sequence when the internal sampling capacitors are charged. Note that the charging of the bypass capacitors through the differential source resistances must keep pace with the charge depletion of the bypass capacitors during the input sampling sequence. Higher source resistances reduce the amount of charging current for the bypass capacitors. Also, note that fast, successive conversion will have the greatest charge depletion effect on the bypass capacitors. Therefore, the above phenomenon becomes more significant as source resistances and the converssion rate (i.e., higher clock frequency and conversion initiation rate) increase.

In addition, if the above phenomenon prevents the bypass capacitors from fully charging between conversions, voltage drops across the source resistances will result due to the ongoing bypass capacitor charging currents. The voltage drops will cause a conversion error. Also, the voltage drops increase with higher $| V_{I+} - V_{I-} |$ values, higher source resistances, and lower charge on the bypass capacitors (i.e., faster conversion rate).

For low-source-resistance applications ($R_{source} < 100\ \Omega$), a 0.001-µF bypass capacitor at the inputs will prevent pickup due to the series lead inductance of a long wire. A 100-Ω resistor can be placed between the capacitor and the output of an operational amplifier to isolate the capacitor from the operational amplifier.

input leads

The input leads should be kept as short as possible, since the coupling of noise and digital clock signals to the inputs can cause errors.

power supply considerations

Noise spikes on the V_{CC} lines can cause conversion error. Low-inductance tantalum capacitors (> 1 µF) with short leads should be used to bypass ANLG V_{CC} and DGTL V_{CC}. A separate regulator for the TLC1225A or TLC1225B and other analog circuitry will greatly reduce digital noise on the supply line.

positive and negative full-scale adjustment

unipolar inputs

Apply a differential input voltage that is 0.5 LSB below the desired analog full-scale voltage (V_{FS}) and adjust the magnitude of the REF input so that the output code is just changing from 0 1111 1111 1110 to 0 1111 1111 1111. If this transition is desired for a different input voltage, the reference voltage can be adjusted accordingly.

bipolar inputs

First, follow the procedure for the unipolar case.

Second, apply a differential input voltage so that the digital output code is just changing from 1 0000 0000 0001 to 1 0000 0000 0000. Call this actual differential voltage V_X. The ideal differential voltage for this transition is:

$$-V_{FS} + \frac{V_{FS}}{8192} \tag{1}$$

The difference between the actual and ideal differential voltages is:

$$\text{Delta} = V_X - \left(-V_{FS} + \frac{V_{FS}}{8192}\right) \tag{2}$$

Then apply a differential input voltage of:

$$V_X - \frac{\text{Delta}}{2} \tag{3}$$

and adjust V_{ref} so the digital output code is just changing from 1 0000 0000 0001 to 1 0000 0000 0000. This procedure produces positive and negative full-scale transitions with symmetrical minimum error.

TYPICAL APPLICATIONS

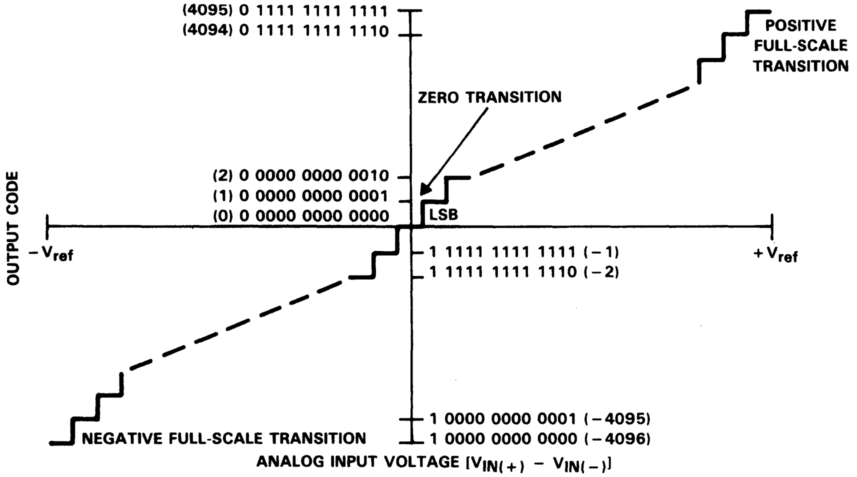
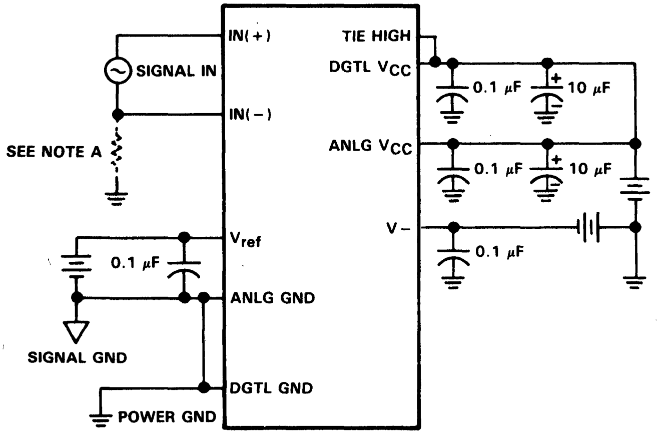


FIGURE 5. TRANSFER CHARACTERISTIC



NOTES: A. The analog input must have some current return path to ANALOG GND.
B. Bypass capacitor leads must be as short as possible.

FIGURE 6. ANALOG CONSIDERATIONS

TYPICAL APPLICATIONS (Continued)

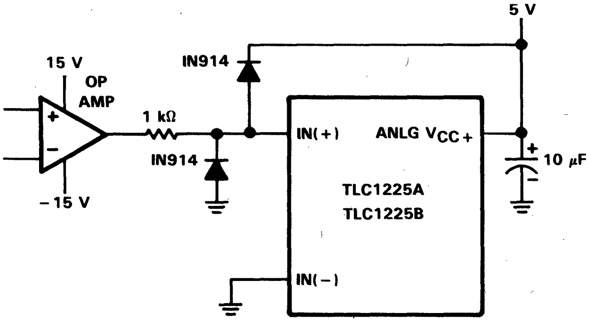
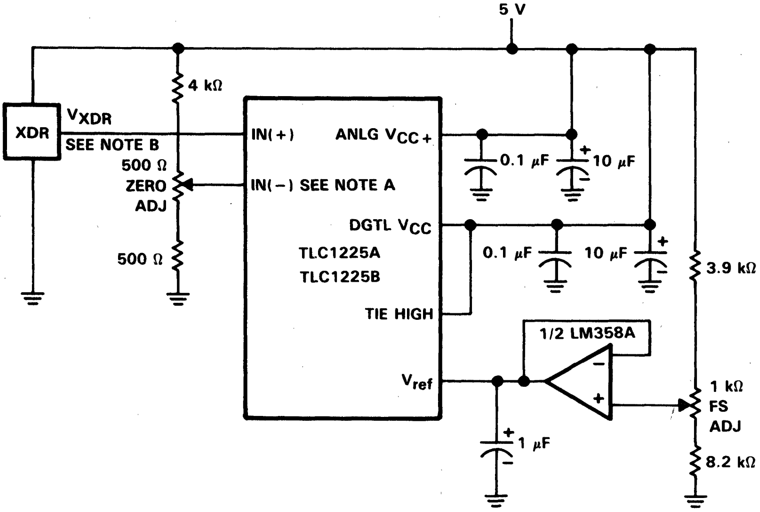


FIGURE 7. INPUT PROTECTION



NOTES: A. $V_{I-} = 0.15 \times \text{ANLG } V_{CC+}$.
B. 15% of ANALOG $V_{CC} \leq V_{XDR} \leq$ 85% of ANALOG V_{CC}.

FIGURE 8. OPERATING WITH RATIOMETRIC TRANSDUCERS

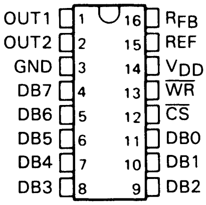
TLC7524
Advanced LinCMOS™ 8-BIT MULTIPLYING DIGITAL-TO-ANALOG CONVERTER

D3008, SEPTEMBER 1986—REVISED OCTOBER 1988

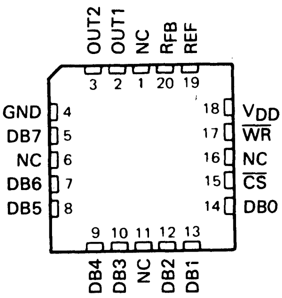
- **Advanced LinCMOS™ Silicon-Gate Technology**
- **Easily Interfaced to Microprocessors**
- **On-Chip Data Latches**
- **Monotonic over the Entire A/D Conversion Range**
- **Segmented High-Order Bits Ensure Low-Glitch Output**
- **Designed to be Interchangeable with Analog Devices AD7524, PMI PM-7524, and Micro Power Systems MP7524**
- **Fast Control Signaling for Digital Signal Processor Applications Including Interface with TMS320**

KEY PERFORMANCE SPECIFICATIONS	
Resolution	8 Bits
Linearity error	½ LSB Max
Power dissipation at V_{DD} = 5 V	5 mW Max
Settling time	100 ns Max
Propagation delay	80 ns Max

D OR N PACKAGE
(TOP VIEW)



FN PACKAGE
(TOP VIEW)



NC—No internal connection

description

The TLC7524 is an Advanced LinCMOS™ 8-bit digital-to-analog converter (DAC) designed for easy interface to most popular microprocessors.

The TLC7524 is an 8-bit multiplying DAC with input latches and with a load cycle similar to the "write" cycle of a random access memory. Segmenting the high-order bits minimizes glitches during changes in the most-significant bits, which produce the highest glitch impulse. The TLC7524 provides accuracy to ½ LSB without the need for thin-film resistors or laser trimming, while dissipating less than 5 milliwatts typically.

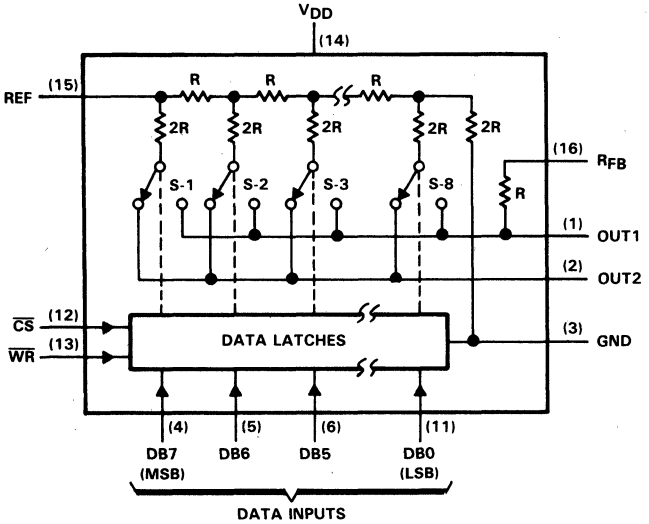
Featuring operation from a 5-V to 15-V single supply, the TLC7524 interfaces easily to most microprocessor buses or output ports. Excellent multiplying (2 or 4 quadrant) makes the TLC7524 an ideal choice for many microprocessor-controlled gain-setting and signal-control applications.

The TLC7524I is characterized for operation from −25°C to 85°C, and the TLC7524C is characterized for operation from 0°C to 70°C.

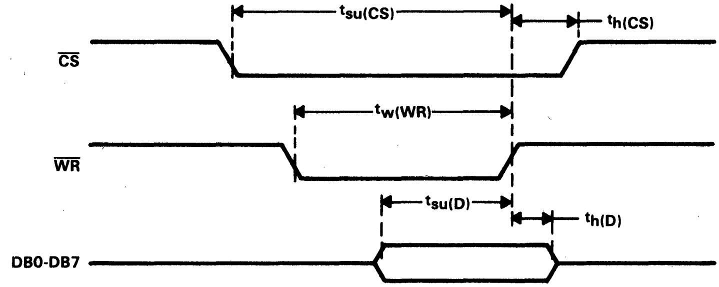
Advanced LinCMOS is a trademark of Texas Instruments Incorporated.

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

functional block diagram



operating sequence



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{DD}		−0.3 V to 16.5 V
Digital input voltage, V_I		−0.3 V to V_{DD}+0.3 V
Reference voltage, V_{ref}		±25 V
Peak digital input current, I_I		10 µA
Operating free-air temperature range:	TLC7524I	−25°C to 85°C
	TLC7524C	0°C to 70°C
Storage temperature range		−65°C to 150°C
Case temperature for 10 seconds: FN package		260°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds: D or N package		260°C

recommended operating conditions

		V_{DD} = 5 V			V_{DD} = 15 V			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{DD}		4.75	5	5.25	14.5	15	15.5	V
Reference voltage, V_{ref}			±10			±10		V
High-level input voltage, V_{IH}		2.4			13.5			V
Low-level input voltage, V_{IL}				0.8			1.5	V
CS setup time, $t_{su(CS)}$		40			40			ns
CS hold time, $t_{h(CS)}$		0			0			ns
Data bus input setup time, $t_{su(D)}$		25			25			ns
Data bus input hold time, $t_{h(D)}$		10			10			ns
Pulse duration, $\overline{\text{WR}}$ low, $t_{w(WR)}$		40			40			ns
Operating free-air temperature, T_A	TLC7524I	−25		85	−25		85	°C
	TLC7524C	0		70	0		70	

electrical characteristics over recommended operating free-air temperature range, V_{ref} = ±10 V, OUT1 and OUT2 at GND (unless otherwise noted)

PARAMETER			TEST CONDITIONS	V_{DD} = 5 V			V_{DD} = 15 V			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
I_{IH}	High-level input current		V_I = V_{DD}			10			10	µA
I_{IL}	Low-level input current		V_I = 0			−10			−10	µA
I_{lkg}	Output leakage current	OUT1	DB0-DB7 at 0 V, $\overline{\text{WR}}$, $\overline{\text{CS}}$ at 0 V, V_{ref} = ±10 V			±400			±200	nA
		OUT2	DB0-DB7 at V_{DD}, $\overline{\text{WR}}$, $\overline{\text{CS}}$ at 0 V, V_{ref} = ±10 V			±400			±200	
I_{DD}	Supply current	Quiescent	DB0-DB7 at V_{IH}min or V_{IL}max			1			2	mA
		Standby	DB0-DB7 at 0 V or V_{DD}			500			500	µA
k_{SVS}	Supply voltage sensitivity, Δgain/ΔV_{DD}		ΔV_{DD} = ±10%		0.01	0.16		0.005	0.04	%FSR/%
C_i	Input capacitance, DB0-DB7, $\overline{\text{WR}}$, $\overline{\text{CS}}$		V_I = 0			5			5	pF
C_o	Output capacitance	OUT1	DB0-DB7 at 0 V, $\overline{\text{WR}}$ and $\overline{\text{CS}}$ at 0 V			30			30	pF
		OUT2				120			120	
C_o	Output capacitance	OUT1	DB0-DB7 at V_{DD}, $\overline{\text{WR}}$ and $\overline{\text{CS}}$ at 0 V			120			120	pF
		OUT2				30			30	
	Reference input impedance (Pin 15 to GND)			5		20	5		20	kΩ

operating characteristics over recommended operating free-air temperature range, V_{ref} = ±10 V, OUT1 and OUT2 at GND (unless otherwise noted)

PARAMETER	TEST CONDITIONS	V_{DD} = 5 V			V_{DD} = 15 V			UNIT
		MIN	TYP	MAX	MIN	TYP†	MAX	
Linearity error				±0.5			±0.5	LSB
Gain error	See Note 1			±2.5			±2.5	LSB
Settling time (to ½ LSB)	See Note 2			100			100	ns
Propagation delay from digital input to 90% of final analog output current	See Note 2			80			80	ns
Feedthrough at OUT1 or OUT2	V_{ref} = ±10 V (100-kHz sinewave) $\overline{\text{WR}}$ and $\overline{\text{CS}}$ at 0 V, DB0-DB7 at 0 V			0.5			0.5	%FSR
Temperature coefficient of gain	T_A = 25°C to MAX		±0.004			±0.001		%FSR/°C

NOTES: 1. Gain error is measured using the internal feedback resistor. Nominal Full Scale Range (FSR) = V_{ref} − 1 LSB.
2. OUT1 load = 100 Ω, C_{ext} = 13 pF, $\overline{\text{WR}}$ at 0 V, $\overline{\text{CS}}$ at 0 V, DB0-DB7 at 0 V to V_{DD} or V_{DD} to 0 V.

principles of operation

The TLC7524 is an 8-bit multiplying D/A converter consisting of an inverted R-2R ladder, analog switches, and data input latches. Binary weighted currents are switched between the OUT1 and OUT2 bus lines, thus maintaining a constant current in each ladder leg independent of the switch state. The high-order bits are decoded and these decoded bits, through a modification in the R-2R ladder, control three equally weighted current sources. Most applications only require the addition of an external operational amplifier and a voltage reference.

The equivalent circuit for all digital inputs low is seen in Figure 1. With all digital inputs low, the entire reference current, I_{ref}, is switched to OUT2. The current source I/256 represents the constant current flowing through the termination resistor of the R-2R ladder, while the current source I_{lkg} represents leakage currents to the substrate. The capacitances appearing at OUT1 and OUT2 are dependent upon the digital input code. With all digital inputs high, the off-state switch capacitance (30 pF maximum) appears at OUT2 and the on-state switch capacitance (120 pF maximum) appears at OUT1. With all digital inputs low, the situation is reversed as shown in Figure 1. Analysis of the circuit for all digital inputs high is similar to Figure 1; however, in this case, I_{ref} would be switched to OUT1.

Interfacing the TLC7524 D/A converter to a microprocessor is accomplished via the data bus and the $\overline{\text{CS}}$ and $\overline{\text{WR}}$ control signals. When $\overline{\text{CS}}$ and $\overline{\text{WR}}$ are both low, the TLC7524 analog output responds to the data activity on the DB0-DB7 data bus inputs. In this mode, the input latches are transparent and input data directly affects the analog output. When either the $\overline{\text{CS}}$ signal or $\overline{\text{WR}}$ signal goes high, the data on the DB0-DB7 inputs are latched until the $\overline{\text{CS}}$ and $\overline{\text{WR}}$ signals go low again. When $\overline{\text{CS}}$ is high, the data inputs are disabled regardless of the state of the $\overline{\text{WR}}$ signal.

The TLC7524 is capable of performing 2-quadrant or full 4-quadrant multiplication. Circuit configurations for 2-quadrant or 4-quadrant multiplication are shown in Figures 2 and 3. Input coding for unipolar and bipolar operation are summarized in Tables 1 and 2, respectively.

principles of operation (continued)

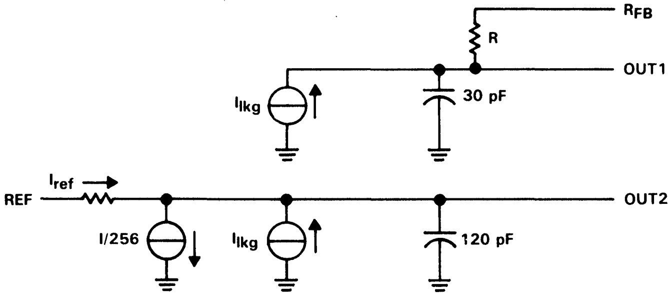


FIGURE 1. TLC7524 EQUIVALENT CIRCUIT WITH ALL DIGITAL INPUTS LOW

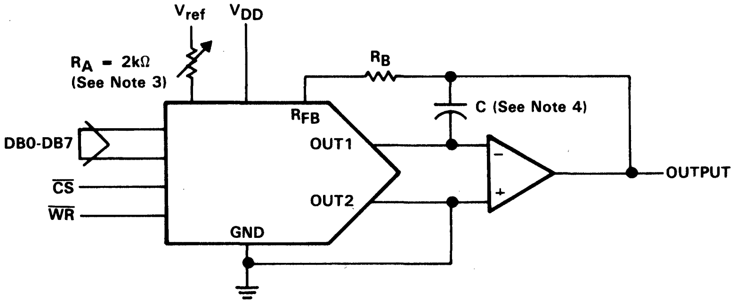


FIGURE 2. UNIPOLAR OPERATION (2-QUADRANT MULTIPLICATION)

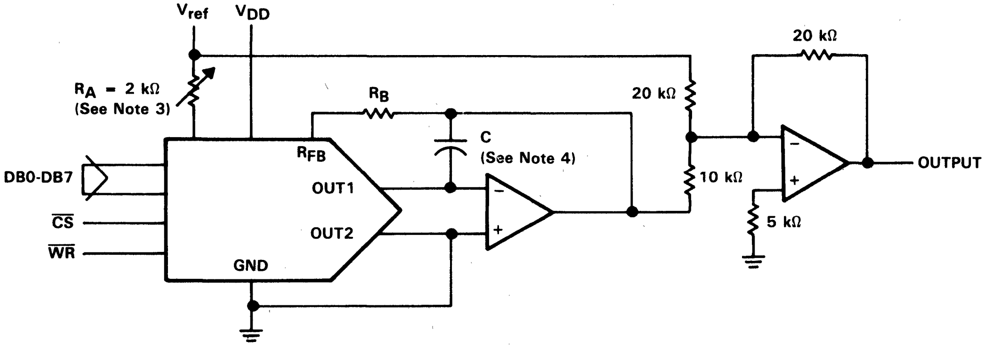


FIGURE 3. BIPOLAR OPERATION (4-QUADRANT OPERATION)

NOTES: 3. R_A and R_B used only if gain adjustment is required.
4. C phase compensation (10-15 pF) is required when using high-speed amplifiers to prevent ringing or oscillation.

principles of operation (continued)

TABLE 1. UNIPOLAR BINARY CODE

DIGITAL INPUT (SEE NOTE 5) MSB LSB	ANALOG OUTPUT
11111111	$-V_{ref}$ (255/256)
10000001	$-V_{ref}$ (129/256)
10000000	$-V_{ref}$ (128/256) = $-V_{ref}/2$
01111111	$-V_{ref}$ (127/256)
00000001	$-V_{ref}$ (1/256)
00000000	0

TABLE 2. BIPOLAR (OFFSET BINARY) CODE

DIGITAL INPUT (SEE NOTE 6) MSB LSB	ANALOG OUTPUT
11111111	V_{ref} (127/128)
10000001	V_{ref} (1/128)
10000000	0
01111111	$-V_{ref}$ (1/128)
00000001	$-V_{ref}$ (127/128)
00000000	$-V_{ref}$

NOTES: 5. LSB = 1/256 (V_{ref}).
6. LSB = 1/128 (V_{ref}).

microprocessor interfaces

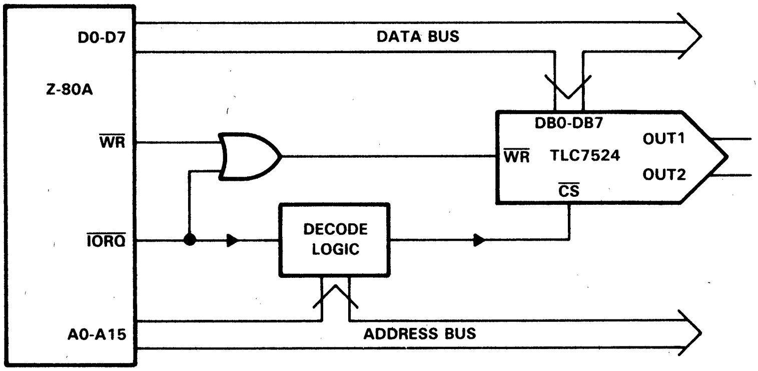


FIGURE 4. TLC7524—Z-80A INTERFACE

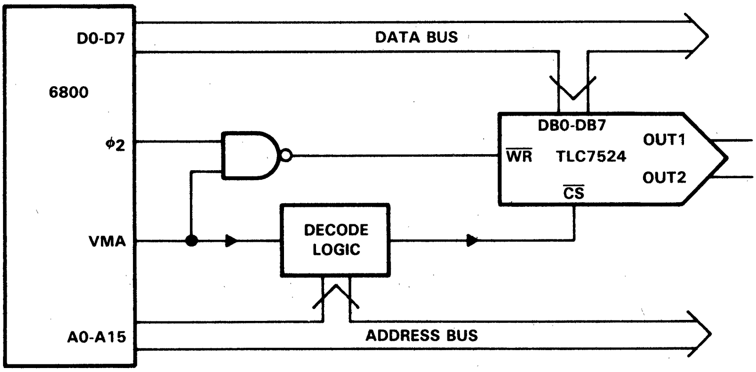


FIGURE 5. TLC7524—6800 INTERFACE

microprocessor interfaces (continued)

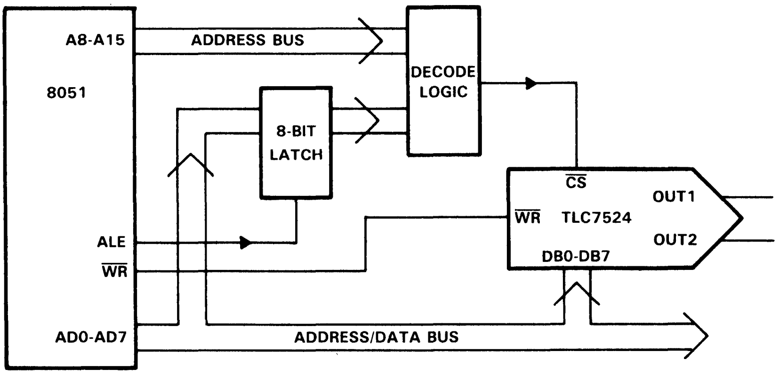


FIGURE 6. TLC7524—8051 INTERFACE

TYPICAL APPLICATION DATA

voltage-mode operation

It is possible to operate the TLC7524 current multiplying D/A converter in a voltage mode. In the voltage mode, a fixed voltage is placed on the current output pin. The analog output voltage is then available at the reference voltage pin. Figure 7 is an example of a current multiplying D/A, which is operated in voltage mode.

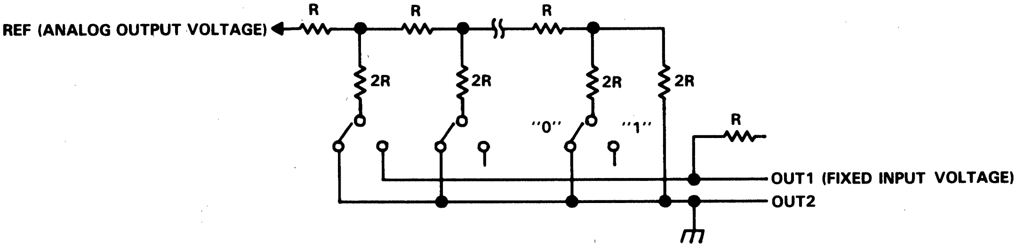


FIGURE 7. VOLTAGE MODE OPERATION

The relationship between the fixed input voltage and the analog output voltage is given by the following equation:

$$V_O = V_I \; (D/256)$$

where

V_O = analog output voltage
V_I = fixed input voltage
D = digital input code converted to decimal

In voltage-mode operation, the TLC7524 will meet the following specification:

PARAMETER	TEST CONDITIONS	MIN	MAX	UNIT
Linearity error at REF	V_{DD} = 5 V, OUT1 = 2.5 V, OUT2 at GND, T_A = 0°C to 70°C		1	LSB

F.2 TI Sockets

The sockets produced by Texas Instruments are designed for high-density packaging needs. The production sockets and burn-in/test sockets for DIP and PLCC packages, described in the following pages, are compatible with TMS320C1x devices.

For additional information about TI sockets, contact the nearest TI sales office or contact:

Texas Instruments Incorporated
Connector Systems Dept, MS 14-3
Attleboro, MA 02703
(617) 699-5242/5269
Telex: 92-7708

IC SOCKETS
DUAL-IN-LINE

PERFORMANCE SPECIFICATIONS

Mechanical

Accommodates IC leads 0.011 ± 0.003 in by 0.018 ± 0.003
Recommended PCB thickness range: 0.062 in to 0.092 in
Recommended PCB hole size range: 0.032 in to 0.042 in
Recommended hole grid pattern: 0.100 in ± 0.003 in each direction
Vibration: 15 G, 10-2000 Hz per MIL-STD 1344A, Method 2005.1 Test Condition III.
Shock: 100 G, sawtooth waveform, 2 shocks each direction per MIL-STD 202, Method 213, Test Condition I
Durability: 5 cycles, 10 mΩ max contact resistance change per MIL-STD 1344, Method 2016
Solderability: per MIL-STD 202, Method 208
Insertion force (C7X and C86): 16 oz (454 g) per pin max
Insertion force (C50): 12 oz per pin max
Withdrawal force: (40 g) per pin min

Electrical

Contact rating: 1.0 A per contact
Contact resistance: 20 mΩ max initial
Insulation resistance: 1000 MΩ at 500 V dc per MIL-STD 1344, Method 3003
Dielectric withstanding voltage: 1000 V ac rms per MIL-STD 1344, Method 3001.1
Capacitance: 1.0 pF max per MIL-STD 202, Method 305

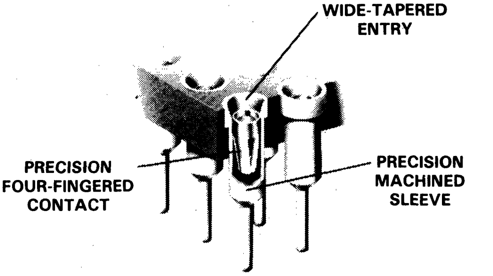
Environmental

Operating temperature: −55°C to 125°C, gold; −40°C to 100°C, tin
Corrosive atmosphere: 10 mΩ max contact resistance change when exposed to 22% ammonium sulfide for 4 hours
Gas tight: 10 mΩ max contact resistance change when exposed to nitric acid vapor for 1 hour
Temperature soak: 10 mΩ max contact resistance change when exposed to 105°C temperature for 48 hours
Shelf life: 12 months min

Materials (C7X, C50, and C86)

Body — PBT polyester U/L 94 VO rating
C7X & C50 Contacts — Outer sleeve: brass
Clip: BECU or PHBR
Contact finish — clip 30 µin gold over 50 µin nickel or 50 µin tin/lead over 50 µin nickel
Specified by Part Number — sleeve 10 µin gold over 50 µin nickel or 50 µin tin/lead over 50 µin nickel
C86 Contacts — Phosphor bronze base metal
C86 Contact-finish — Tin plate 200 µin over copper flash

C7X SERIES — SCREW MACHINE



C7X SERIES
PART NUMBER SYSTEM

C7X (X) XX – X X

X — Variations
Solder Tail
9 — Pin length 0.105/0.150
Wire Wrap
3 — Pin length 0.510

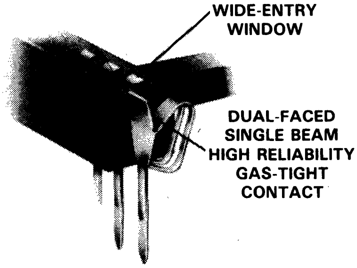
X — Plating (Sleeve/Clip)
0 — Gold/Gold
5 — Tin/Gold

XX — Number of Positions

(X) — S — Single-in-line package (where applicable)

C7X — Screw Machine Socket
1 — wire wrap
2 — solder tail

C86 SERIES — STAMPED AND FORMED



C86 SERIES
PART NUMBER SYSTEM

C 86 XX – 01

01 — Variation
01 — Standard product

XX — Number of positions

86 — Tin Dual Face Wipe Single Beam

C — TI Socket Series

IC SOCKETS
BURN-IN/TEST DIP

PERFORMANCE SPECIFICATIONS

Mechanical

Accommodates IC leads 0.011 in by 0.018 in NOM
Recommended PCB thickness range: 0.062 in to 0.092 in
Recommended PCB hold size range: 0.032 in to 0.042 in
Durability: 10K cycles — CM Series, 5K cycles — CP/CQ
Solderability: per MIL-STD 202, Method 208

Electrical

Contact rating: 1.0 A per contact
Contact resistance: 20 mΩ max initial
Insulation resistance: 1000 MΩ at 500 V dc
Dielectric withstanding voltage: 1000 V ac rms
Capacitance: 1.0 pF max per MIL-STD 202, Method 305

Environmental

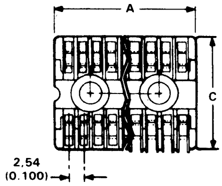
Operating temperature: −65 °C to 170 °C — CP/CM Series, −65 °C to 150 °C — CQ Series
Humidity: 10 mΩ max contact resistance
Temperature Soak: 10 mΩ max contact resistance change

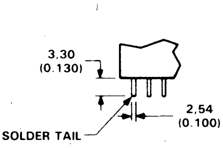
MATERIALS

Body — PPS (polyphenylen sulfide) glass filled U/L 94 VO
Contacts — Higher performance copper nickel alloy
Plating:† 4 μin of gold min over 100 μin of nickel min

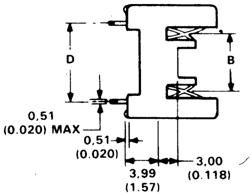
†For additional plating options consult the factory

BURN-IN/TEST DIP SOCKETS

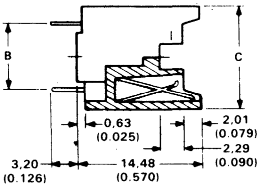




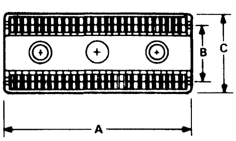
CQ37 SERIES

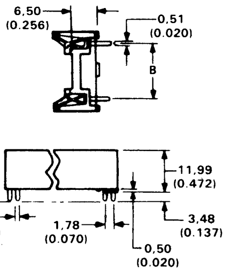


CP37 SERIES

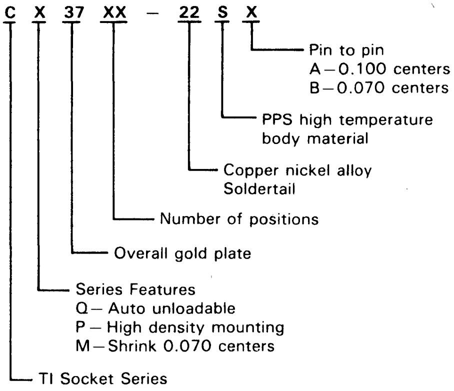


CM37 SERIES





PART NUMBER SYSTEM



CQ37 SERIES

Number of Positions	A ±0.01 Length	D ±0.02	C ±0.01 Width	B ±0.01 Contact
14 16 18 20	20,32 (0.800) 22,35 (0.880) 24,89 (0.980) 27,43 (1.080)	12,70 (0.500)	15,24 (0.600)	7,62 (0.300)
24 28 40 42	32,51 (1,280) 37,59 (1.480) 52,83 (2.080) 55,37 (2.180)	19,05 (0.750)	22,86 (0.900)	15,24 (0.600)

CP37 SERIES

Number of Positions	A max Length	B ±0.02	C max Width
8 14 16 18 20	11,68 (0.460) 17,78 (0.700) 20,32 (0.800) 22,86 (0.900) 25,40 (1.000)	7,62 (0.300)	12,70 (0.500)
24 28 40	30,48 (1.200) 35,56 (1.400) 50,80 (2.000)	15,24 (0.600)	20,32 (0.800)

CM37 SERIES

Number of Positions	A ±0.016 Length	B ±0.02	C ±0.016 Width
28	27,18 (1.070)	10,67 (0.420)	17,20 (0.677)
40 42 54	37,85 (1.490) 39,62 (1.560) 50,29 (1.980)	16,51 (0.650)	23,11 (0.910)
64	59,18 (2.330)	20,32 (0.800)	26,92 (1.060)

Dimensions in parentheses are inches
Contact factory for detailed information

IC SOCKETS
PLASTIC LEADED CHIP CARRIER

PERFORMANCE SPECIFICATIONS

Mechanical

Recommended PCB thickness range: 0.062 in to 0.092 in
Recommended PCB hole size range: 0.032 in to 0.042 in
Vibration: 15 G
Shock: 100 G
Solderability: Per MIL-STD 202, Method 208
Insertion force: 0.59 lbs per position
Withdrawal force: 0.25 lbs per position
Normal force: 200 g min, 450 g typ
Wipe: 0.075 in min
Durability: 5 cycles min
Contact retention: 1.5 lbs min

Electrical

Current carrying capacity: 1 A
Insulation resistance: 5000 MΩ min
Dielectric withstanding voltage: 1000 V ac rms min
Capacitance: 1.0 pF max

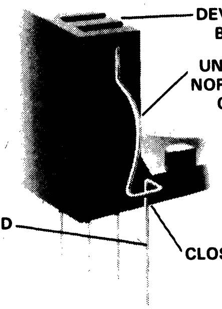
Environmental

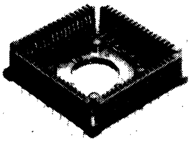
Operating temperature:
Operating: – 40°C to 85°C
Storage: –40°C to 95°C
Temperature cycling with humidity: will conform to final EIA specifications
Shelf life: 1 year min

MATERIALS

Body — Ryton R-4 (40% glass) U/L 94-VO rating
Contacts — CDA 510 spring temper
Contact finish — 90/10 tin (200 μin –400 μin) over 40 μin copper

Contact factory for detailed information



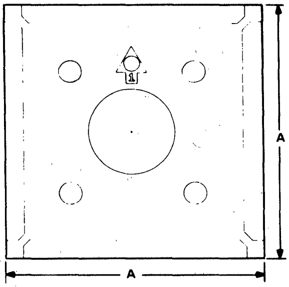


PART NUMBER SYSTEM

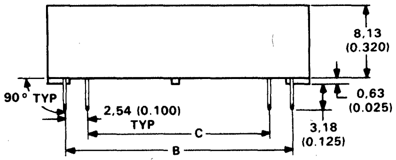
CPR PH XXX – X – X – 0

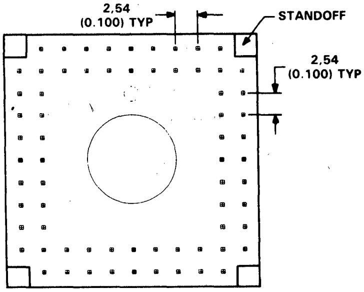
- Contact surface 1 — tin lead plating
- Contact spacing 1 — 0.050 in
- Number of pos (044, 052, 068, 084)
- Plated thru hole, solder tail
- TI socket Series
  Plastic leaded chip carrier

PLASTIC LEADER CHIP CARRIER CPR SERIES



Device guide barriers not shown





Pos	A	B	C
44	21,43 (0.844	17,78 (0.700)	12,70 (0.500)
52	23,98 (0.944)	20,32 (0.800)	15,24 (0.600)
68	29,06 (1.144)	25,40 (1.000)	20,32 (0.800)
84	34,14 (1.344)	30,48 (1.200)	25,40 (1.000)

Extraction tool available, consult factory.

IC SOCKETS
PLCC BURN-IN/TEST

PRODUCT FEATURES

Can be loaded by top actuated insertion or press-in insertion, either manually or automatically
High reliability due to high pressure contact point
Open body and high stand-off design provide high efficiency in heat dissipation
High durability up to 10,000 cycles
Compact design

PERFORMANCE SPECIFICATIONS

Mechanical

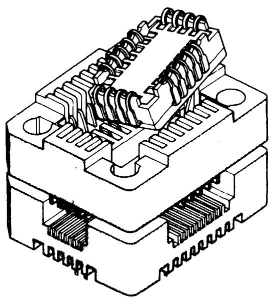
Durability: 10,000 cycles
Operating Temperature: 180°C max

Electrical

Contact rating: 1.0 A per contact
Contact resistance: 30 mΩ max
Insulation resistance: 1000 MΩ min
Dielectric withstanding voltage: 500 V ac rms min

MATERIALS

Body — ultem glass filled (U/L 94 VO)
Contact — copper alloy
Plating — overall gold plate



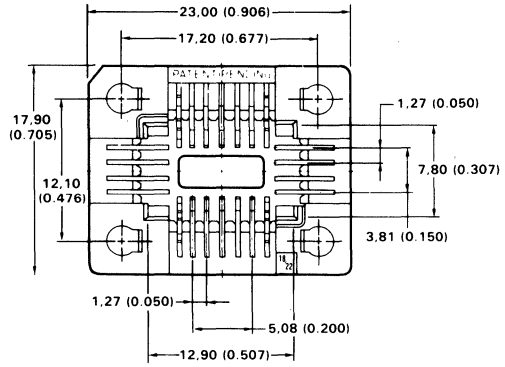
PART NUMBER SYSTEM

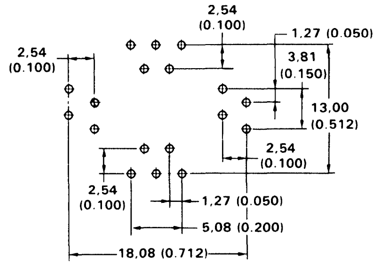
CPJ AA33A – XXX B

XXX — Number of positions

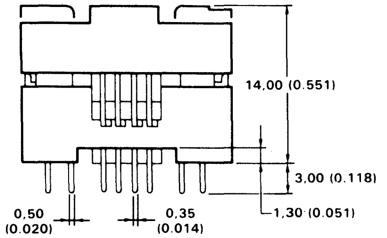
CPJ — TI series socket

PLCC BURN-IN/TEST SOCKETS CPJ SERIES





SIZES: 18 PIN
22 PIN



Dimensions in parentheses are inches
Contact factory for detailed information

F.3 Crystals

This section lists the commonly used crystal frequencies, crystal specification requirements, and the names of suitable vendors.

Table F-1 lists the commonly used crystal frequencies and the devices with which they can be used.

Table F-1. Commonly Used Crystal Frequencies

FREQUENCY	DEVICE
14 MHz	TMS32010-14, TMS320C10-14
18.432 MHz	TMS32010/C10, TMS320C15/E15, TMS320C17/E17
20 MHz	TMS32010/C10, TMS320C15/E15, TMS320C17/E17
20.48 MHz	TMS32010/C10, TMS320C15/E15, TMS320C17/E17
25.6 MHz	TMS32010/C10, TMS320C15/E15, TMS320C17/E17

A crystal connected across X1 and X2/CLKIN on the TMS320 processor enables the internal oscillator, as shown in Figure F-1. The frequency of CLKOUT is one-fourth the crystal fundamental frequency. Crystal specification requirements are listed below.

Load capacitance = 20 pF
Series resistance = 30 ohm
Power dissipation = 1 mW
Parallel resonant
14-MHz and 20-MHz crystals use fundamental mode.
25-MHz operation may require third-overtone crystal.

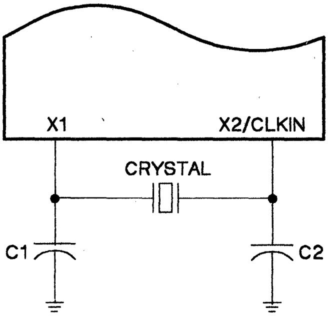


Figure F-1. Crystal Connection

Appendix F - Crystals

Vendors of crystals suitable for use with TMS320 devices are listed below.

RXD, Inc.
Norfolk, NB
(800) 228-8108

CTS Knight, Inc.
Contact the local distributor

N.E.L. Frequency Controls, Inc.
Burlington, WI
(414) 763-3591

Appendix G

Programming the TMS320E15/E17 EPROM Cell

This appendix presents the TMS320E15/E17 EPROM cells which are featured in the *First-Generation Digital Signal Processors* data sheet. Both devices, TMS320E15/E17, include one 4K x 16-bit EPROM which is implemented from a standard EPROM cell. This expands their capabilities in the areas of prototyping, early field testing, and production. When used with either 4K-word masked-ROM TMS320C15/C17, the appropriate TMS320E15/E17 yields a more cost-effective production as a result of more migration paths for data.

EPROM adaptor sockets are available which provide pin-to-pin conversion for programming the TMS320E15/E17. One adaptor socket (part number RTC/PGM320A-06) is shown in Figure G-1 and is capable of converting a 40-pin DIP device into an equivalent 28-pin device. Another socket (part number RTC/PGM320C-06), not shown, permits a 44- to 28-pin conversion.

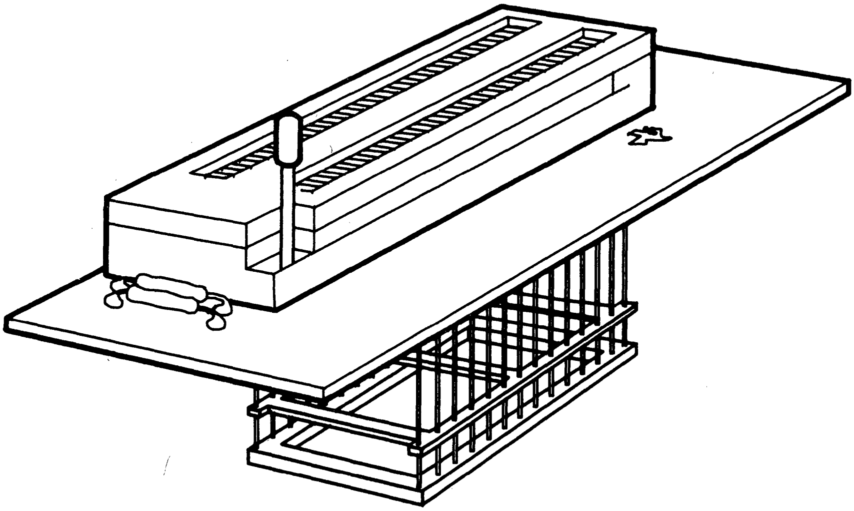


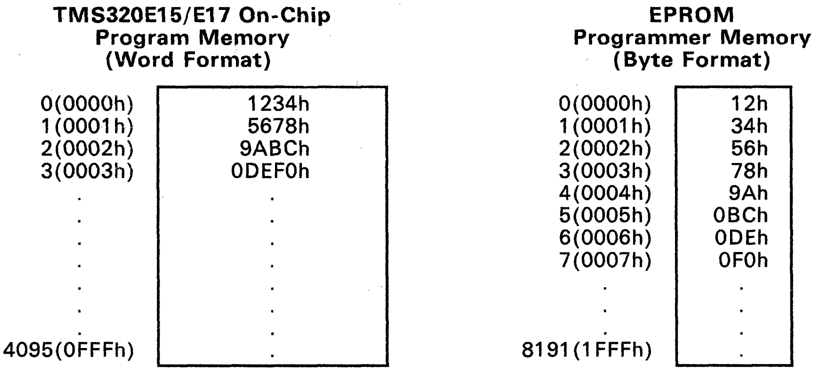
Figure G-1. EPROM Adaptor Socket

Key features of the EPROM cell include standard programming and verification. The EPROM cell also includes a code protection feature that allows code to be protected against copyright violations. The protection feature can be used to protect reading the EPROM contents. This appendix describes erasure, FAST programming and verification, and EPROM protection and verification.

G.1 FAST Programming and Verification

Both TMS320E15/E17 EPROM cells are similar to the TMS27C64 8K x 8-bit EPROM. Their memories can be erased by using an ultraviolet light source and electrically programmed by using the same family and device codes. The TMS320E15/E17, like the TMS27C64, devices operate from a 5-V supply for reading and a 12.5-V supply for programming. All programming signals are TTL level. For programming outside the system, existing EPROM programmers can be used. Locations may be systematically or randomly programmed as a singular or blocked address. When programmed in a block format, each byte of data is separately loaded into the EPROM cell with the high byte preceding the low byte. The manufacturing process is largely responsible for their dissimilarity. Due to HVCMOS technology, the TMS27C64 has a read-only memory; the memories of TMS320E15/E17 have both reading and writing capabilities. The TMS27C64 is pin-to-pin compatible with all 28-pin ROMs and EPROMs.

The TMS320E15/E17 uses 13 address lines to address the 4K-word memory in byte format (8K-byte memory). In word format, the most-significant byte of each word is assigned an even address while the least-significant byte is assigned an odd address in the byte format. Programming information must be downloaded into the EPROM programmer memory in a high-byte to low-byte order for proper programming (see Figure G-2).



TMS320E15/E17 On-Chip Program Memory (Word Format)	
0(0000h)	1234h
1(0001h)	5678h
2(0002h)	9ABCh
3(0003h)	0DEF0h
.	.
.	.
.	.
4095(0FFFh)	.

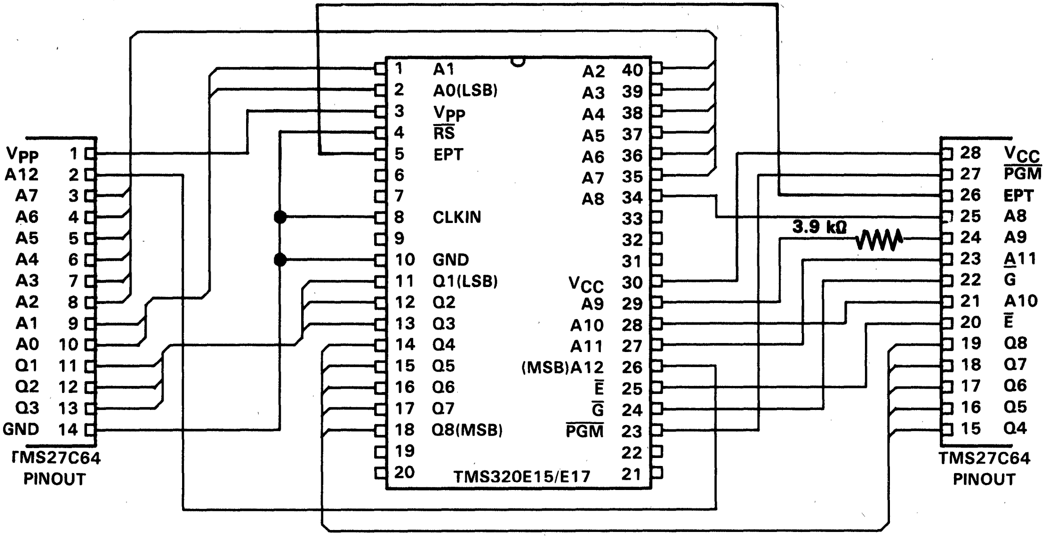
EPROM Programmer Memory (Byte Format)	
0(0000h)	12h
1(0001h)	34h
2(0002h)	56h
3(0003h)	78h
4(0004h)	9Ah
5(0005h)	0BCh
6(0006h)	0DEh
7(0007h)	0F0h
.	.
.	.
8191(1FFFh)	.

Figure G-2. EPROM Programming Data Format

Figure G-3 shows the wiring diagram for programming the TMS320E15/E17 by using the TMS27C64 in its 28-pin output form. The illustration furnishes a table for each pin nomenclature on the TMS27C64 with a description of that pin. Programming the code into either EPROM device should be done in the serial mode.

Caution:

Although acceptable by some EPROM programmers, the signature mode cannot be used on any TMS320E1x device. The signature mode will input a high-level voltage (12.5 Vdc) onto pin A9. Since the TMS320E1x EPROM cell is not designed for high voltage, the cell will be damaged. To prevent an accidental application of voltage, Texas Instruments has inserted a 3.9 kΩ resistor between pin A9 of the TI programmer socket and the programmer itself.



PIN NOMENCLATURE

NAME	I/O	DEFINITION
A12(MSB)-A0(LSB)	I	On-chip EPROM programming address lines
CLKIN	I	Clock oscillator input
E	I	EPROM chip select
EPT	I	EPROM test mode select
G	I	EPROM read/verify select
GND	I	Ground
PGM	I	EPROM write/program select
Q8(MSB)-Q1(LSB)	I/O	Data lines for byte-wide programming of on-chip 8K bytes of EPROM
RS	I	Reset for initializing the device
V_{CC}	I	5-V power supply
V_{PP}	I	12.5-V power supply

Figure G-3. TMS320E15/E17 EPROM Conversion to TMS27C64 EPROM Pinout

Table G-1 shows the programming levels required for programming, verifying, and reading the EPROM cell. Following the table, individual paragraphs describe the function of each programming level.

Table G-1. TMS320E15/E17 Programming Mode Levels

SIGNAL NAME†	TMS320E15/ E17 DIP PIN	TMS27C64 DIP PIN	PROGRAM	PROGRAM VERIFY	PROGRAM INHIBIT	READ	OUTPUT DISABLE
$\overline{E}$	25	20	V_{IL}	V_{IL}	V_{IH}	V_{IL}	V_{IL}
$\overline{G}$	24	22	V_{IH}	$\overline{PULSE}$	X	$\overline{PULSE}$	V_{IH}
$\overline{PGM}$	23	27	$\overline{PULSE}$	V_{IH}	X	V_{IH}	V_{IH}
V_{PP}	3	1	V_{PP}	V_{PP}	V_{PP}	V_{CC}	V_{CC}
V_{CC}	30	28	$V_{CC}+1$	$V_{CC}+1$	$V_{CC}+1$	V_{CC}	V_{CC}
V_{SS}	10	14	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
CLKIN	8	14	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
$\overline{RS}$	4	14	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
EPT	5	26	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
Q8-Q1	18-11	19-15,13-11	D_{IN}	Q_{OUT}	HI-Z	Q_{OUT}	HI-Z
A12-A10	26-28	2,23,21	ADDR	ADDR	X	ADDR	X
A9-A7	29,34,35	24,25,3	ADDR	ADDR	X	ADDR	X
A6	36	4	ADDR	ADDR	X	ADDR	X
A5	37	5	ADDR	ADDR	X	ADDR	X
A4	38	6	ADDR	ADDR	X	ADDR	X
A3-A0	39,40,1,2	7-10	ADDR	ADDR	X	ADDR	X

LEGEND:
† = in accordance with TMS27C64.
V_{IH} = TTL high level; V_{IL} = TTL low level; ADDR = byte address bit
$V_{PP} = 12.5 \pm 0.25$ V; $V_{CC} = 5 \pm 0.25$ V; X = don't care
$\overline{PULSE}$ = low-going TTL level pulse; D_{IN} = byte to be programmed at ADDR
Q_{OUT} = byte stored at ADDR.

Erasure

Before programming, the memory must be erased by exposing high-intensity ultraviolet through its transparent lid. Note that normal ambient light contains the correct wavelength for erasure. Therefore, the window should be covered with an opaque label after programming either the TMS320E15/E17. The recommended minimum exposure dose (UV-intensity × exposure-time) is 15 watt-seconds per square centimeter. If located about 2.5 centimeters above the transparent lid, a typical 12 milliwatt per square centimeter, filterless UV lamp will erase the device in 21 minutes. After erasing the memory, all bits are in a high state.

FAST Programming

After erasing, all memory bits in the cell are a logic one. Logic zeroes must be programmed into their desired locations. The FAST Programming algorithm, shown in Figure G-4, is normally used to program the entire EPROM contents, although individual locations may be programmed separately. A programmed logic zero can only be erased by ultraviolet light. Data is presented in parallel (eight bits) on pins Q8-Q1. Once addresses and data are stable, $\overline{\text{PGM}}$ is pulsed. The programming mode is achieved when V_{PP} = 12.5 V, $\overline{\text{PGM}}$ = V_{IL}, V_{CC} = 6.0 V, $\overline{\text{G}}$ = V_{IH}, and $\overline{\text{E}}$ = V_{IL}. More than one TMS320E15/E17 can be programmed when the devices are connected in parallel. Locations can be programmed in any order.

FAST Programming uses two types of programming pulses: prime and final. The length of the prime pulse is 1 ms. After each prime pulse, the byte being programmed is verified. If correct data is read, the final programming pulse is applied; if correct data is not read, an additional 1-ms prime pulse is applied up to a maximum of 15 times. The final programming pulse is 4 ms times the number of prime programming pulses applied. This sequence of programming and verification is performed at V_{CC} = 6.0 V, and V_{PP} = 12.5 V. When the full FAST Programming routine is complete, all bits are verified with V_{CC} = V_{PP} = 5 V.

Program Verify

Programmed bits may be verified with V_{PP} = 12.5 V when $\overline{\text{G}}$ = V_{IL}, $\overline{\text{E}}$ = V_{IL}, and $\overline{\text{PGM}}$ = V_{IH}. Figure G-5 shows the timing for the program and verify operation for the FAST programs.

Program Inhibit

Programming may be inhibited by maintaining a high level input on the $\overline{\text{E}}$ pin or $\overline{\text{PGM}}$ pin.

Read

The EPROM contents may be read independent of the programming cycle, provided the RBIT (ROM protect bit) has not been programmed. The read is accomplished by setting $\overline{\text{E}}$ to zero and pulsing $\overline{\text{G}}$ low. The contents of the EPROM location selected by the value on the address inputs appear on Q8-Q1.

Output Disable

During the EPROM programming process, the EPROM data outputs may be disabled, if desired, by establishing the output disable state. This state is selected by setting the $\overline{\text{G}}$ and $\overline{\text{E}}$ pins high. While output disable is selected, Q8-Q1 are placed in the high-impedance state.

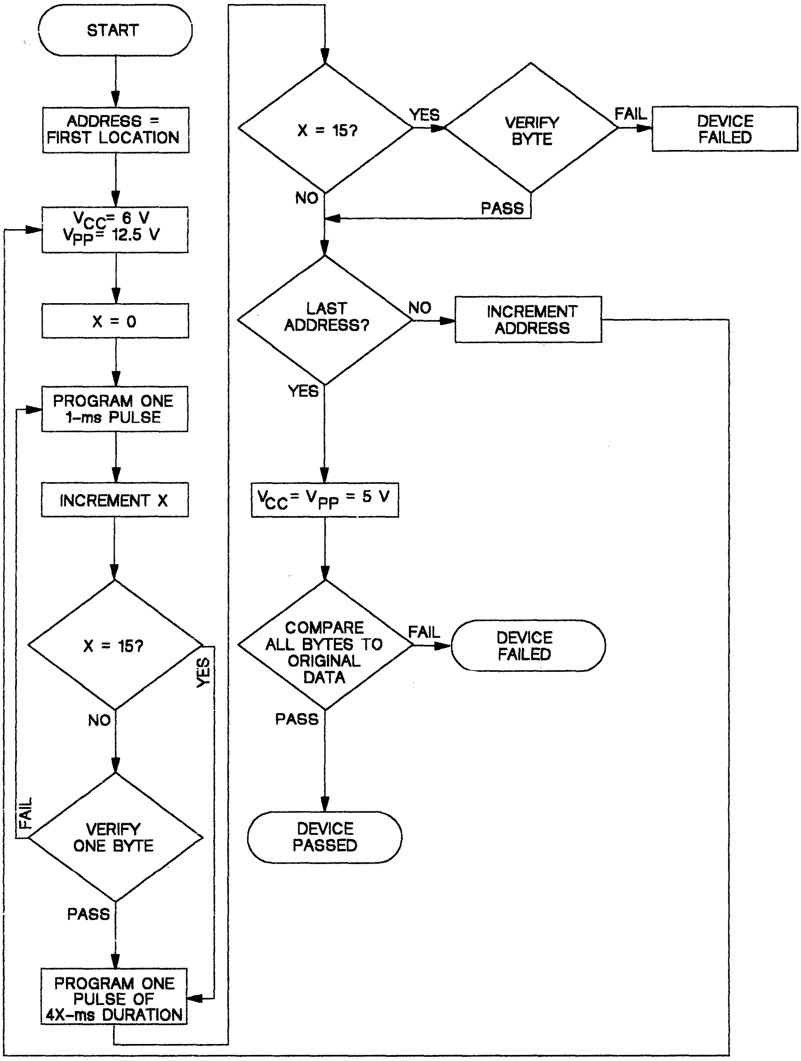


Figure G-4. FAST Programming Flowchart

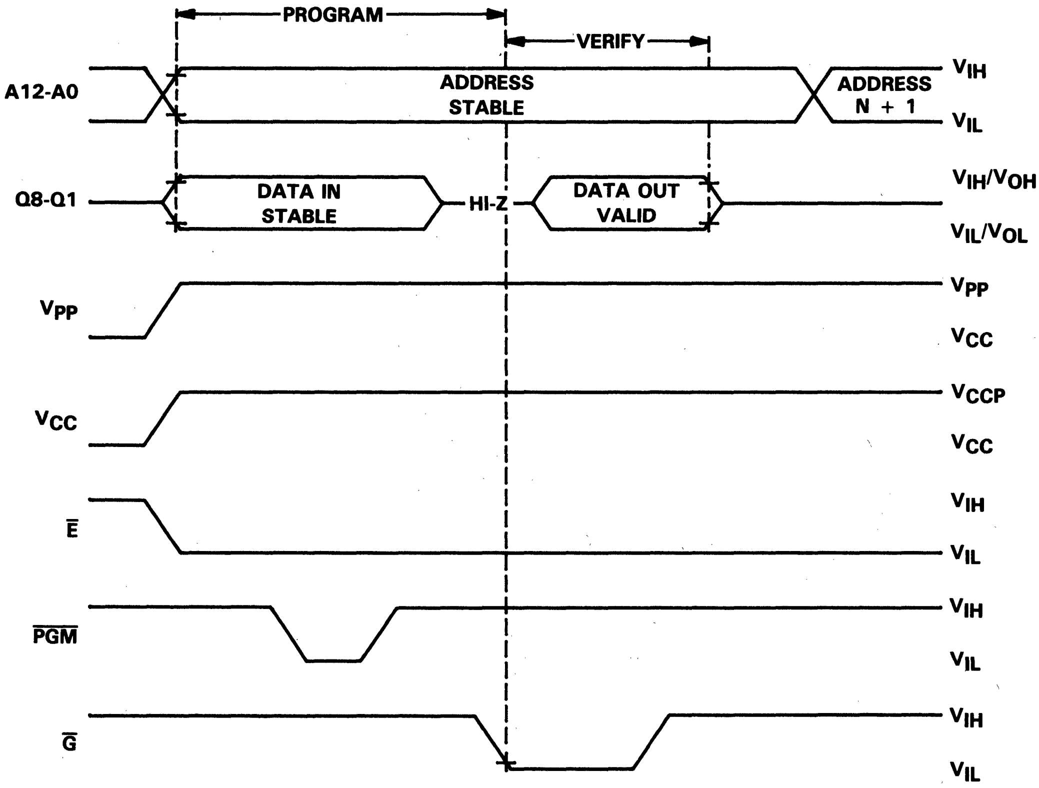


Figure G-5. Fast Programming Timing

G.2 EPROM Protection and Verification

This section describes the code protection feature included in the EPROM cell, which protects code against copyright violations. Table G-2 shows the programming levels required for protecting the EPROM and verifying the protection. Following the table, individual paragraphs describe the protect and verify functions.

Table G-2. TMS320E15/E17 EPROM Protect and Protect Verify Mode Levels

SIGNAL NAME†	TMS320E15/E17 DIP PIN	TMS27C64 DIP PIN	EPROM PROTECT	PROTECT VERIFY
$\overline{E}$	25	20	V_{IH}	V_{IL}
$\overline{G}$	24	22	V_{IH}	V_{IL}
$\overline{PGM}$	23	27	V_{IH}	V_{IH}
V_{PP}	3	1	V_{PP}	$V_{CC}+1$
V_{CC}	30	28	$V_{CC}+1$	$V_{CC}+1$
V_{SS}	10	14	V_{SS}	V_{SS}
CLKIN	8	14	V_{SS}	V_{SS}
$\overline{RS}$	4	14	V_{SS}	V_{SS}
EPT	5	26	V_{PP}	V_{PP}
Q8-Q1	18-11	19-15,13-11	Q8=$\overline{PULSE}$	Q8=RBIT
A12-A10	26-28	2,23,21	X	X
A9-A7	29,34,35	24,25,3	X	X
A6	36	4	X	V_{IL}
A5	37	5	X	X
A4	38	6	V_{IH}	X
A3-A0	39,40,1,2	7-10	X	X

LEGEND:
† = in accordance with TMS27C64.
V_{IH} = TTL high level; V_{IL} = low-level TTL, V_{CC} = 5 ± 0.25 V
V_{PP} = 12.5 ± 0.25 V; X = don't care
$\overline{PULSE}$ = low-going TTL level pulse; RBIT = ROM protect bit

EPROM Protection

The EPROM protection facility is used to completely disable reading of the EPROM contents to guarantee security of proprietary algorithms. This facility is implemented through a unique EPROM cell called the RBIT (ROM protect bit) cell. Once the contents to be protected are programmed into the EPROM, the RBIT is programmed, disabling access to the EPROM contents and disabling the microprocessor mode on the device. Once programmed, the RBIT can only be cleared by erasing the entire EPROM array with ultraviolet light, thereby maintaining security of the proprietary algorithm. Programming the RBIT is accomplished using the EPROM protection cycle, which consists of setting the $\overline{E}$, $\overline{G}$, $\overline{PGM}$, and A4 pins high, V_{PP} and EPT to 12.5 ± 0.25 V, and pulsing Q8 low. The complete sequence of operations involved in programming the RBIT is shown in the flowchart of Figure G-6. The required setups in the figure are detailed in Table G-2.

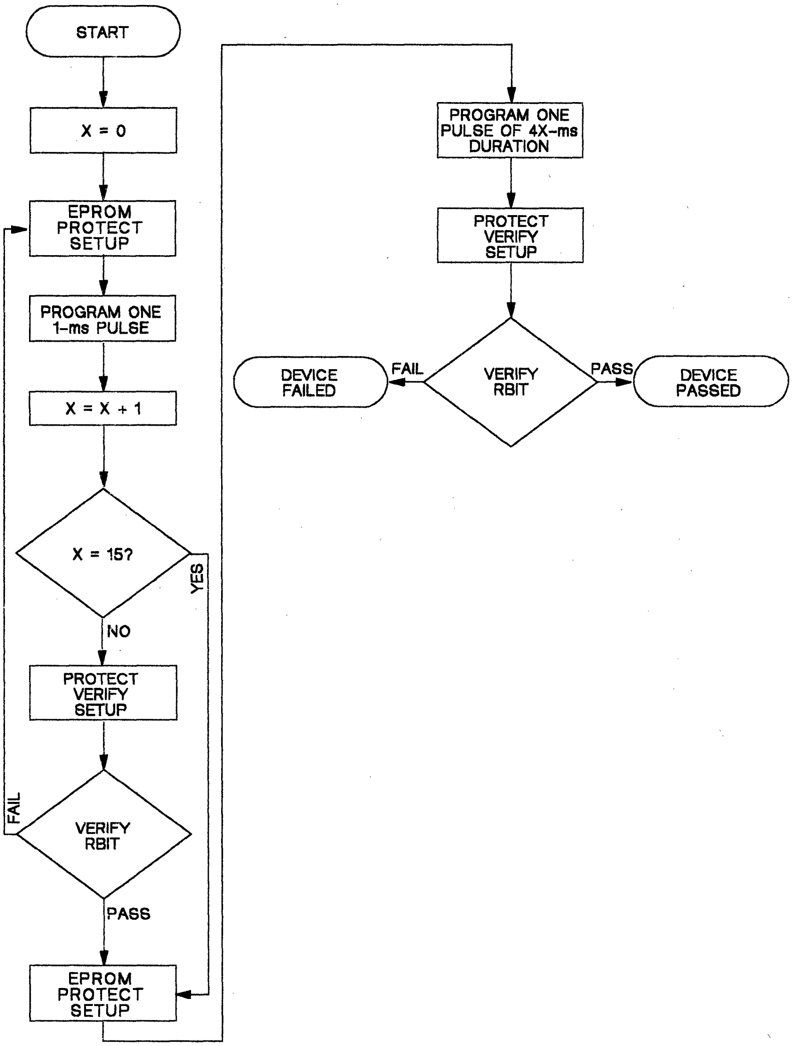
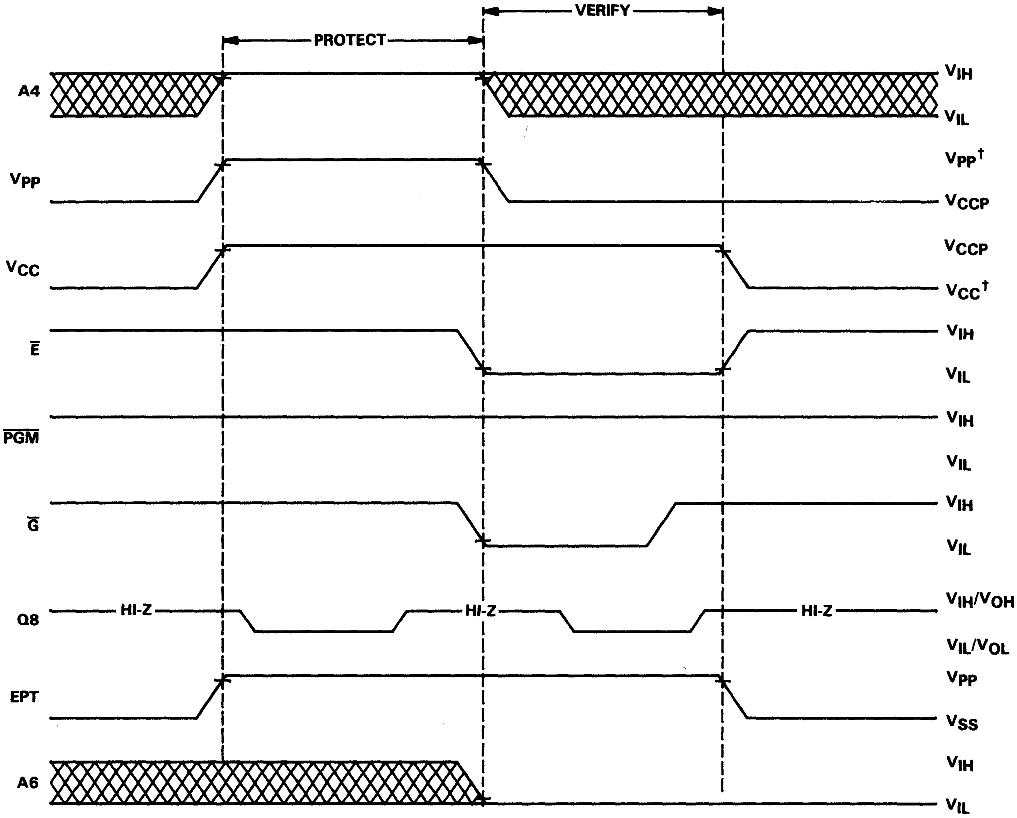


Figure G-6. EPROM Protection Flowchart

Protect Verify

Protect verify is used following the EPROM protection to verify correct programming of the RBIT (see Figure G-6). When using protect verify, Q8 outputs the state of the RBIT. When RBIT = 1, the EPROM is unprotected; when RBIT = 0, the EPROM is protected. The EPROM protection and verify timings are shown in Figure G-7.



†V_{PP} = 12.5 V and V_{CC} = 6.0 V for Fast Programming.

Figure G-7. EPROM Protection Timing

Index

TI Worldwide Sales Offices

ALABAMA: Huntsville: 500 Wynn Drive, Suite 514, Huntsville, AL 35805, (205) 837-7530.

ARIZONA: Phoenix: 8825 N. 23rd Ave., Phoenix, AZ 85021, (602) 995-1007;**TUCSON:** 818 W. Miracle Mile, Suite 43, Tucson, AZ 85705, (602) 292-2640.

CALIFORNIA: Irvine: 17891 Cartwright Dr., Irvine, CA 92714, (714) 660-1200; **Roseville:** 1 Sierra Gate Plaza, Roseville, CA 95678, (916) 786-9208; **San Diego:** 4333 View Ridge Ave., Suite 100, San Diego, CA 92123, (619) 278-9601; **Santa Clara:** 5353 Betsy Ross Dr., Santa Clara, CA 95054, (408) 980-9000; **Torrance:** 690 Knox St., Torrance, CA 90502, (213) 217-7010; **Woodland Hills:** 21220 Erwin St., Woodland Hills, CA 91367, (818) 704-7759.

COLORADO: Aurora: 1400 S. Potomac Ave., Suite 101, Aurora, CO 80012, (303) 368-8000.

CONNECTICUT: Wallingford: 9 Barnes Industrial Park Rd., Barnes Industrial Park, Wallingford, CT 06492, (203) 269-0074.

FLORIDA: Altamonte Springs: 370 S. North Lake Blvd, Altamonte Springs, FL 32701, (305) 260-2116; **Ft. Lauderdale:** 2950 N.W. 62nd St., Ft. Lauderdale, FL 33309, (305) 973-8502; **Tampa:** 4803 George Rd., Suite 390, Tampa, FL 33634, (813) 885-7411.

GEORGIA: Norcross: 5515 Spalding Drive, Norcross, GA 30092, (404) 662-7900

ILLINOIS: Arlington Heights: 515 W. Algonquin, Arlington Heights, IL 60005, (312) 640-2925.

INDIANA: Ft. Wayne: 2020 Inwood Dr., Ft. Wayne, IN 46815, (219) 424-5174; **Carmel:** 550 Congressional Dr., Carmel, IN 46032, (317) 573-6400.

IOWA: Cedar Rapids: 373 Collins Rd. NE, Suite 201, Cedar Rapids, IA 52402, (319) 395-9550.

KANSAS: Overland Park: 7300 College Blvd., Lighton Plaza, Overland Park, KS 66210, (913) 451-4511.

MARYLAND: Columbia: 8815 Centre Park Dr., Columbia MD 21045, (301) 964-2003.

MASSACHUSETTS: Waltham: 950 Winter St., Waltham, MA 02154, (617) 895-9100.

MICHIGAN: Farmington Hills: 33737 W. 12 Mile Rd., Farmington Hills, MI 48018, (313) 553-1569. **Grand Rapids:** 3075 Orchard Vista Dr. S.E., Grand Rapids, MI 49506, (616) 957-4200.

MINNESOTA: Eden Prairie: 11000 W. 78th St., Eden Prairie, MN 55344 (612) 828-9300.

MISSOURI: St. Louis: 11816 Borman Drive, St. Louis, MO 63146, (314) 569-7600.

NEW JERSEY: Iselin: 485E U.S. Route 1 South, Parkway Towers, Iselin, NJ 08830 (201) 750-1050.

NEW MEXICO: Albuquerque: 2820-D Broadbent Pkwy NE, Albuquerque, NM 87107, (505) 345-2555.

NEW YORK: East Syracuse: 6365 Collamer Dr., East Syracuse, NY 13057, (315) 463-9291; **Melville:** 1895 Walt Whitman Rd., P.O. Box 2936, Melville, NY 11747, (516) 454-6600; **Pittsford:** 2851 Clover St., Pittsford, NY 14534, (716) 385-6770; **Poughkeepsie:** 385 South Rd., Poughkeepsie, NY 12601, (914) 473-2900.

NORTH CAROLINA: Charlotte: 8 Woodlawn Green, Woodlawn Rd., Charlotte, NC 28210, (704) 527-0933; **Raleigh:** 2809 Highwoods Blvd., Suite 100, Raleigh, NC 27625, (919) 876-2725.

OHIO: Beachwood: 23775 Commerce Park Rd., Beachwood, OH 44122, (216) 464-6100; **Beavercreek:** 4200 Colonel Glenn Hwy., Beavercreek, OH 45431, (513) 427-6200.

OREGON: Beaverton: 6700 SW 105th St., Suite 110, Beaverton, OR 97005, (503) 643-6758.

PENNSYLVANIA: Blue Bell: 670 Sentry Pkwy, Blue Bell, PA 19422, (215) 825-9500.

PUERTO RICO: Hato Rey: Mercantil Plaza Bldg., Suite 505, Hato Rey, PR 00918, (809) 753-8700.

TENNESSEE: Johnson City: Erwin Hwy, P.O. Drawer 1255, Johnson City, TN 37605 (615) 461-2192.

TEXAS: Austin: 12501 Research Blvd., Austin, TX 78759, (512) 250-7655; **Richardson:** 1001 E. Campbell Rd., Richardson, TX 75081, (214) 680-5082; **Houston:** 9100 Southwest Frwy., Suite 250, Houston, TX 77074, (713) 778-6592; **San Antonio:** 1000 Central Parkway South, San Antonio, TX 78232, (512) 496-1779.

UTAH: Murray: 5201 South Green St., Suite 200, Murray, UT 84123, (801) 266-8972.

WASHINGTON: Redmond: 5010 148th NE, Bldg B, Suite 107, Redmond, WA 98052, (206) 881-3080.

WISCONSIN: Brookfield: 450 N. Sunny Slope, Suite 150, Brookfield, WI 53005, (414) 782-2899.

CANADA: Nepean: 301 Moodie Drive, Mallorn Center, Nepean, Ontario, Canada, K2H9C4, (613) 726-1970. **Richmond Hill:** 280 Centre St. E., Richmond Hill L4C1B1, Ontario, Canada (416) 884-9181; **St. Laurent:** Ville St. Laurent Quebec, 9460 Trans Canada Hwy., St. Laurent, Quebec, Canada H4S1R7, (514) 336-1860.

ARGENTINA: Texas Instruments Argentina Viamonte 1119, 1053 Capital Federal, Buenos Aires, Argentina, 541/748-3699

AUSTRALIA (& NEW ZEALAND): Texas Instruments Australia Ltd.: 6-10 Talavera Rd., North Ryde (Sydney), New South Wales, Australia 2113, 2 + 887-1122; 5th Floor, 418 St. Kilda Road, Melbourne, Victoria, Australia 3004, 3 + 267-4677; 171 Philip Highway, Elizabeth, South Australia 5112, 8 + 255-2066.

AUSTRIA: Texas Instruments Ges.m.b.H.: Industriestrabe B/16, A-2345 Brunn/Gebirge, 2236-846210.

BELGIUM: Texas Instruments N.V. Belgium S.A.: 11, Avenue Jules Bondetlaan 11, 1140 Brussels, Belgium, (02) 242-3080.

BRAZIL: Texas Instruments Electronicos do Brasil Ltda.: Rua Paes Leme, 524-7 Andar Pinheiros, 05424 Sao Paulo, Brazil, 0815-6166.

DENMARK: Texas Instruments A/S, Mairelundvej 46E, 2730 Herlev, Denmark, 2 - 91 74 00.

FINLAND: Texas Instruments Finland OY: Ahertajantie 3, P.O. Box 81, ESPOO, Finland, (90) 0-461-422.

FRANCE: Texas Instruments France: Paris Office, BP 67 8-10 Avenue Morane-Saulnier, 78141 Velizy-Villacoublay cedex (1) 30 70 1003.

GERMANY (Fed. Republic of Germany): Texas Instruments Deutschland GmbH: Haggertystrasse 1, 8050 Freising, 8161 + 80-4591; Kurfuerstendamm 195/196, 1000 Berlin 15, 30 + 882-7365; III, Hagen 43/Kibbelstrasse, .19, 4300 Essen, 201-24250; Kirchhorsterstrasse 2, 3000 Hannover 51, 511 + 648021; Maybachstrabe 11, 7302 Ostfildern 2-Nelingen, 711 + 34030.

TEXAS
INSTRUMENTS

HONG KONG: Texas Instruments Hong Kong Ltd., 8th Floor, World Shipping Ctr., 7 Canton Rd., Kowloon, Hong Kong, (852) 3-7351223.

IRELAND: Texas Instruments (Ireland) Limited: 7/8 Harcourt Street, Stillorgan, County Dublin, Eire, 1 781677.

ITALY: Texas Instruments Italia S.p.A. Divisione Semiconduttori: Viale Europa, 40, 20093 Cologne Monzese (Milano), (02) 253001; Via Castello della Magliana, 38, 00148 Roma, (06) 5222651; Via Amendola, 17, 40100 Bologna, (051) 554004.

JAPAN: Tokyo Marketing/Sales (Headquarters): Texas Instruments Japan Ltd., MS Shibaura Bldg., 9F, 4-13-23 Shibaura, Minato-ku, Tokyo 108, Japan, 03-769-8700. Texas Instruments Japan Ltd.: Nissho-Iwai Bldg. 5F, 30 Imabashi 3-chome, Higashi-ku, Osaka 541, Japan, 06-294-1881; Daini Toyota West Bldg. 7F, 10-27 Meieki 4-chome, Nakamura-ku, Nagoya 450, 052-583-8691; Daiichi Seimei Bldg. 6F, 3-10 Oyama-cho, Kanazawa 920, Ishikawa-ken, 0762-23-5471; Daiichi Olympic Tachikawa Bldg. 6F, 1-25-12 Akebono-cho, Tachikawa 190, Tokyo, 0425-27-6426; Matsumoto Showa Bldg. 6F, 2-11 Fukashi 1-chome, Matsumoto 390, Nagano-ken, 0263-33-1060; Yokohama Nishiguchi KN Bldg. 6F, 2-8-4 Kita-Saiwai-cho, Nishi-ku, Yokohama 220, 045-322-6741; Nihon Seimei Kyoto Yasaka Bldg. 5F, 843-2 Higashi Shiokohjidori, Nishinotoh-in Higashi-iru, Shiokouji, Shimogyo-ku, Kyoto 600, 075-341-7713; 2597-1, Aza Harudai, Oaza Yasaka, Kitsuki 873, Oita-ken, 09786-3-3211; Miho Plant, 2350 Kihara Miho-mura, Inashiki-gun 300-04, Ibaragi-ken, 0298-85-2541.

KOREA: Texas Instruments Korea Ltd., 28th Fl., Trade Tower, #159, Samsung-Dong, Kangnam-ku, Seoul, Korea 2 + 551-2810.

MEXICO: Texas Instruments de Mexico S.A.: Alfonso Reyes—115, Col. Hipodromo Condesa, Mexico, D.F., Mexico 06120, 525/525-3860.

MIDDLE EAST: Texas Instruments: No. 13, 1st Floor Mannai Bldg., Diplomatic Area, P.O. Box 26335, Manama Bahrain, Arabian Gulf, 973 + 274681.

NETHERLANDS: Texas Instruments Holland B.V., 19 Hogehilweg, 1100 AZ Amsterdam—Zuidoost, Holland 20 + 5602911.

NORWAY: Texas Instruments Norway A/S: PB106, Refstad 0585, Oslo 5, Norway, (2) 155090.

PEOPLES REPUBLIC OF CHINA: Texas Instruments China Inc., Beijing Representative Office, 7-05 Citic Bldg., 19 Jianguomenwai Dajje, Beijing, China, (861) 5002255, Ext. 3750.

PHILIPPINES: Texas Instruments Asia Ltd.: 14th Floor, Ba- Lepanto Bldg., Paseo de Roxas, Makati, Metro Manila, Philippines, 817-60-31.

PORTUGAL: Texas Instruments Equipamento Electronico (Portugal), Lda.: Rua Eng. Frederico Ulrich, 2650 Moreira Da Maia, 4470 Maia, Portugal, 2-948-1003.

SINGAPORE (+ INDIA, INDONESIA, MALAYSIA, THAILAND): Texas Instruments Singapore (PTE) Ltd., Asia Pacific Division, 101 Thompson Rd. #23-01, United Square, Singapore 1130, 350-8100.

SPAIN: Texas Instruments Espana, S.A.: C/Jose Lazaro Galdiano No. 6, Madrid 28036, 1/458.14.58.

SWEDEN: Texas Instruments International Trade Corporation (Sverigefilialen): S-164-93, Stockholm, Sweden, 8 - 752-5800.

SWITZERLAND: Texas Instruments, Inc., Reidstrasse 6, CH-8953 Dietikon (Zuerich) Switzerland, 1-740 2220.

TAIWAN: Texas Instruments Supply Co., 9th Floor Bank Tower, 205 Tun Hwa N. Rd., Taipei, Taiwan, Republic of China, 2 + 713-9311.

UNITED KINGDOM: Texas Instruments Limited: Manton Lane, Bedford, MK41 7PA, England, 0234 270111.

TI Sales Offices

ALABAMA: Huntsville (205) 837-7530.

ARIZONA: Phoenix (602) 995-1007;
Tucson (602) 292-2640.

CALIFORNIA: Irvine (714) 660-1200;
Roseville (916) 786-9208;
San Diego (619) 278-9601;
Santa Clara (408) 980-9000;
Torrance (213) 217-7010;
Woodland Hills (818) 704-7759.

COLORADO: Aurora (303) 368-8000.

CONNECTICUT: Wallingford (203) 269-0074.

FLORIDA: Altamonte Springs (305) 260-2116;
Ft. Lauderdale (305) 973-8502;
Tampa (813) 885-7411.

GEORGIA: Norcross (404) 662-7900.

ILLINOIS: Arlington Heights (312) 640-2925.

INDIANA: Carmel (317) 573-6400;
Ft. Wayne (219) 424-5174.

IOWA: Cedar Rapids (319) 395-9550.

KANSAS: Overland Park (913) 451-4511.

MARYLAND: Columbia (301) 964-2003.

MASSACHUSETTS: Waltham (617) 895-9100.

MICHIGAN: Farmington Hills (313) 553-1569;
Grand Rapids (616) 957-4200.

MINNESOTA: Eden Prairie (612) 828-9300.

MISSOURI: St. Louis (314) 569-7600.

NEW JERSEY: Iselin (201) 750-1050.

NEW MEXICO: Albuquerque (505) 345-2555.

NEW YORK: East Syracuse (315) 463-9291;
Melville (516) 454-6600;
Pittsford (716) 385-6770;
Poughkeepsie (914) 473-2900.

NORTH CAROLINA: Charlotte (704) 527-0933;
Raleigh (919) 876-2725.

OHIO: Beachwood (216) 464-6100;
Beaver Creek (513) 427-6200.

OREGON: Beaverton (503) 643-6758.

PENNSYLVANIA: Blue Bell (215) 825-9500.

PUERTO RICO: Hato Rey (809) 753-8700.

TENNESSEE: Johnson City (615) 461-2192.

TEXAS: Austin (512) 250-7655;
Houston (713) 778-6592;
Richardson (214) 680-5082;
San Antonio (512) 496-1779.

UTAH: Murray (801) 266-8972.

WASHINGTON: Redmond (206) 881-3080.

WISCONSIN: Brookfield (414) 782-2899.

CANADA: Nepean, Ontario (613) 726-1970;
Richmond Hill, Ontario (416) 884-9181;
St. Laurent, Quebec (514) 336-1860.

TI Regional Technology Centers

CALIFORNIA: Irvine (714) 660-8105;
Santa Clara (408) 748-2220;

GEORGIA: Norcross (404) 662-7945.

ILLINOIS Arlington Heights (312) 640-2909.

MASSACHUSETTS: Waltham (617) 895-9196.

TEXAS: Richardson (214) 680-5066.

CANADA: Nepean, Ontario (613) 726-1970.

TI Distributors

TI AUTHORIZED DISTRIBUTORS
Arrow/Kierulff Electronics Group
Arrow (Canada)
Future Electronics (Canada)
GRS Electronics Co., Inc.
Hall-Mark Electronics
Marshall Industries
Newark Electronics
Schweber Electronics
Time Electronics
Wyle Laboratories
Zeus Components

– OBSOLETE PRODUCT ONLY –
Rochester Electronics, Inc.
Newburyport, Massachusetts
(508) 462-9332

ALABAMA: Arrow/Kierulff (205) 837-6955;
Hall-Mark (205) 837-8700; Marshall (205) 881-9235;
Schweber (205) 895-0480.

ARIZONA: Arrow/Kierulff (602) 437-0750;
Hall-Mark (602) 437-1200; Marshall (602) 496-0290;
Schweber (602) 431-0030; Wyle (602) 866-2888.

CALIFORNIA: Los Angeles/Orange County:
Arrow/Kierulff (818) 701-7500, (714) 838-5422;
Hall-Mark (818) 773-4500, (714) 669-4100;
Marshall (818) 407-0101, (818) 459-5500,
(714) 458-5395; Schweber (818) 880-9686;
(714) 863-0200, (213) 320-8090; Wyle (818) 880-9000,
(714) 863-9953; Zeus (714) 921-9000; (818) 889-3838;
Sacramento: Hall-Mark (916) 624-9781;
Marshall (916) 635-9700; Schweber (916) 364-0222;
Wyle (916) 638-5282;
San Diego: Arrow/Kierulff (619) 565-4800;
Hall-Mark (619) 268-1201; Marshall (619) 578-9600;
Schweber (619) 450-0454; Wyle (619) 565-9171;
San Francisco Bay Area: Arrow/Kierulff (408) 745-6600,
Hall-Mark (408) 432-0900; Marshall (408) 942-4600;
Schweber (408) 432-7171; Wyle (408) 727-2500;
Zeus (408) 998-5121.

COLORADO: Arrow/Kierulff (303) 790-4444;
Hall-Mark (303) 790-1662; Marshall (303) 451-8383;
Schweber (303) 799-0258; Wyle (303) 457-9953.

CONNETICUT: Arrow/Kierulff (203) 265-7741;
Hall-Mark (203) 271-2844; Marshall (203) 265-3822;
Schweber (203) 264-4700.

FLORIDA: Ft. Lauderdale:
Arrow/Kierulff (305) 429-8200; Hall-Mark (305) 971-9280;
Marshall (305) 977-4880; Schweber (305) 977-7511;
Orlando: Arrow/Kierulff (407) 323-0252;
Hall-Mark (407) 830-5855; Marshall (407) 767-8585;
Schweber (407) 331-7555; Zeus (407) 365-3000;
Tampa: Hall-Mark (813) 530-4543;
Marshall (813) 576-1399; Schweber (813) 541-5100.

GEORGIA: Arrow/Kierulff (404) 449-8252;
Hall-Mark (404) 447-8000; Marshall (404) 923-5750;
Schweber (404) 449-9170.

ILLINOIS: Arrow/Kierulff (312) 250-0500;
Hall-Mark (312) 860-3800; Marshall (312) 490-0155;
Newark (312) 784-5100; Schweber (312) 364-3750.

INDIANA: Indianapolis: Arrow/Kierulff (317) 243-9353;
Hall-Mark (317) 872-8875; Marshall (317) 297-0483;
Schweber (317) 843-1050.

IOWA: Arrow/Kierulff (319) 395-7230;
Schweber (319) 373-1417.

KANSAS: Kansas City: Arrow/Kierulff (913) 541-9542;
Hall-Mark (913) 888-4747; Marshall (913) 492-3121;
Schweber (913) 492-2922.

TEXAS
INSTRUMENTS

MARYLAND: Arrow/Kierulff (301) 995-6002;
Hall-Mark (301) 988-9800; Marshall (301) 235-9464;
Schweber (301) 840-5900; Zeus (301) 997-1118.

MASSACHUSETTS Arrow/Kierulff (508) 658-0900;
Hall-Mark (508) 667-0902; Marshall (508) 658-0810;
Schweber (617) 275-5100; Time (617) 532-6200;
Wyle (617) 273-7300; Zeus (617) 863-8800.

MICHIGAN: Detroit: Arrow/Kierulff (313) 462-2290;
Hall-Mark (313) 462-1205; Marshall (313) 525-5850;
Newark (313) 967-0600; Schweber (313) 525-8100;
Grand Rapids: Arrow/Kierulff (616) 243-0912.

MINNESOTA: Arrow/Kierulff (612) 830-1800;
Hall-Mark (612) 941-2600; Marshall (612) 559-2211;
Schweber (612) 941-5280.

MISSOURI: St. Louis: Arrow/Kierulff (314) 567-6888;
Hall-Mark (314) 291-5350; Marshall (314) 291-4650;
Schweber (314) 739-0526.

NEW HAMPSHIRE: Arrow/Kierulff (603) 668-6968;
Schweber (603) 625-2250.

NEW JERSEY: Arrow/Kierulff (201) 538-0900,
(609) 596-8000; GRS Electronics (609) 964-8560;
Hall-Mark (201) 575-4415, (201) 882-9773,
(609) 235-1900; Marshall (201) 882-0320,
(609) 234-9100; Schweber (201) 227-7880.

NEW MEXICO: Arrow/Kierulff (505) 243-4566.

NEW YORK: Long Island:
Arrow/Kierulff (516) 231-1009; Hall-Mark (516) 737-0600;
Marshall (516) 273-2424; Schweber (516) 334-7474;
Zeus (914) 937-7400;
Rochester: Arrow/Kierulff (716) 427-0300;
Hall-Mark (716) 425-3300; Marshall (716) 235-7620;
Schweber (716) 424-2222;
Syracuse: Marshall (607) 798-1611.

NORTH CAROLINA: Arrow/Kierulff (919) 876-3132,
(919) 725-8711; Hall-Mark (919) 872-0712;
Marshall (919) 878-9882; Schweber (919) 876-0000.

OHIO: Cleveland: Arrow/Kierulff (216) 248-3990;
Hall-Mark (216) 349-4632; Marshall (216) 248-1788;
Schweber (216) 464-2970;
Columbus: Hall-Mark (614) 888-3313;
Dayton: Arrow/Kierulff (513) 435-5563;
Marshall (513) 898-4480; Schweber (513) 439-1800.

OKLAHOMA: Arrow/Kierulff (918) 252-7537;
Schweber (918) 622-8003.

OREGON: Arrow/Kierulff (503) 645-6456;
Marshall (503) 644-5050; Wyle (503) 640-6000.

PENNSYLVANIA: Arrow/Kierulff (412) 856-7000,
(215) 928-1800; GRS Electronics (215) 922-7037;
Marshall (412) 963-0441; Schweber (215) 441-0600,
(412) 963-6804.

TEXAS: Austin: Arrow/Kierulff (512) 835-4180;
Hall-Mark (512) 258-8848; Marshall (512) 837-1991;
Schweber (512) 339-0088; Wyle (512) 834-9957;
Dallas: Arrow/Kierulff (214) 380-6464;
Hall-Mark (214) 553-4300; Marshall (214) 233-5200;
Schweber (214) 661-5010; Wyle (214) 235-9953;
Zeus (214) 783-7010;
El Paso: Marshall (915) 593-0706;
Houston: Arrow/Kierulff (713) 530-4700;
Hall-Mark (713) 781-6100; Marshall (713) 895-9200;
Schweber (713) 784-3600; Wyle (713) 879-9953.

UTAH: Arrow/Kierulff (801) 973-6913;
Hall-Mark (801) 972-1008; Marshall (801) 485-1551;
Wyle (801) 974-9953.

WASHINGTON: Arrow/Kierulff (206) 575-4420;
Marshall (206) 486-5747; Wyle (206) 881-1150.

WISCONSIN: Arrow/Kierulff (414) 792-0150;
Hall-Mark (414) 797-7844; Marshall (414) 797-8400;
Schweber (414) 784-9020.

CANADA: Calgary: Future (403) 235-5325;
Edmonton: Future (403) 438-2858;
Montreal: Arrow Canada (514) 735-5511;
Future (514) 694-7710;
Ottawa: Arrow Canada (613) 226-6903;
Future (613) 820-8313;
Quebec City: Arrow Canada (418) 871-7500;
Toronto: Arrow Canada (416) 672-7769;
Future (416) 638-4771; Marshall (416) 674-2161;
Vancouver: Arrow Canada (604) 291-2986;
Future (604) 294-1166.

Customer Response Center

TOLL FREE: (800) 232-3200

OUTSIDE USA: (214) 995-6611
(8:00 a.m. – 5:00 p.m. CST)

TEXAS
INSTRUMENTS

Printed in U.S.A., March 1989
1604933-9702

SPRU013B