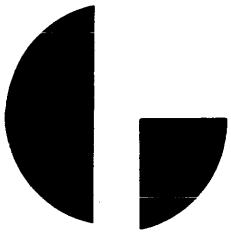


DATA GENERAL
CORPORATION
Southboro,
Massachusetts 01772
(617) 485-9100

TECHNICAL MANUAL

FOR THE

NOVA 800

VOLUME I

015-000004-00

LIST OF EFFECTIVE PAGES

Page No.	Issue	Page No.	Issue
Title	Original	4-1 thru 4-50	Original
A	Original	5-1 thru 5-13	Original
i thru v	Original	6-1 thru 6-85	Original
1-1 thru 1-29	Original	7-1 thru 7-44*	Original
2-1 thru 2-13	Original	A-1 thru A-44	Original
3-1 thru 3-7	Original	B-1 thru B-57	Original

* Section 7 contains 44 pages minimum. Additional drawings are added to this section by each selected option.

FOREWORD

Data General Corporation (DGC) has prepared this manual for use by DGC personnel and customers as a guide to the proper installation, operation, and maintenance of DGC equipment and software. The drawings and specifications contained herein are the property of DGC and shall neither be reproduced in whole or in part without DGC prior written approval nor be implied to grant any license to make, use, or sell equipment manufactured in accordance herewith.

Users of this manual are cautioned that the specifications contained herein are subject to change by DGC at any time without prior notice. Users should refer to the latest revision before using this manual in connection with the operation of DGC equipment. DGC is not responsible for errors which may appear in the drawings and specifications in this manual.

Any comments concerning this publication should be forwarded:

Publications Dept.
Data General Corp.
Route 9
Southboro, Mass. 01772

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

GENERAL DESCRIPTION

1-1 INTRODUCTION

This manual contains a detailed technical presentation of the installation, operation, and maintenance procedures for the Nova 800 Computer. The Nova 800 Computer, as described in this document, consists of the Nova 800 central processor with one or more 4K core memory assemblies. The Basic I/O Control assembly is also described in this manual. The Basic I/O Control can be configured to control three basic types of peripheral equipment, specifically the Teletype, the Paper Tape Reader, and the Paper Tape Punch. Both the memory and the Basic I/O Control with its various equipment configurations are options which may be purchased independent of the Nova 800 central processor. This manual, however, addresses its presentation to the classical definition of a Computer and describes the central processor with 4K of core memory and a Teletype I/O facility. It should be noted that Technical Manuals for each optional peripheral device is produced by the original manufacturer and is shipped under separate cover with the corresponding equipment. This accompanying documentation should be thoroughly reviewed immediately after the peripheral device is received and prior to installation.

This manual is intended to complement Data General Reference Manual, "How to Use the Nova Computers". Operation and Programming information is provided in the Reference Manual and will not be repeated here except where necessary for expositional continuity. It is recommended that all potential users of this publication become familiar with the Reference Manual prior to reviewing this Technical Manual.

The detailed technical descriptions presented in this manual assumes the user of this document has a thorough knowledge of the operation of TTL μLogic circuitry and the fundamentals of digital Computer operations. The contents of this manual are divided into three individual volumes with each volume assembled and shipped under separate cover. Volume I contains Sections I through V, Appendix A, and Appendix B. Volume II contains the Illustrated Parts List. Volume III contains all of the engineering reference drawings. All three volumes are integrated into the all-inclusive Master Documentation Package for this

equipment at the time of shipment. The input/output pin definitions of the various types of integrated circuit (IC) packages constituting the Nova 800 and Basic I/O Control logic are summarized in Appendix A of this manual. The illustrated parts list, presented in Section VI of this manual, can be used as a cross reference source which indexes the reference designator for the IC part (appearing on the logic diagram) with the corresponding manufacturer's part number. Once the IC manufacturer's part number has been retrieved from Section VI it can be used to reference the corresponding IC logic symbol and pin definitions listed in Appendix A.

1-2 GENERAL FUNCTIONAL DESCRIPTION

The Nova 800 is a general purpose Computer with a 16-bit word length. The memory cycle time for the Nova 800 is 800 nanoseconds. The Nova 800 contains four hardware accumulators which are used for temporary data storage and data manipulation during the execution of all arithmetic and logic class (ALC) instructions. The four accumulators also perform double duty as part of the Input/Output system. Data exchanges between the interface logic of the external device and the Nova 800 processor are performed under program control using the four accumulators.

The data paths within the Nova 800 Processor are 16 bits wide and features a bi-directional (I/O) Data bus which not only transmits data between the Central Processor Units (CPU) and the external I/O devices, but also allows external I/O devices to communicate directly with memory, bypassing the CPU. The speed of Data Channel operations between memory and external I/O high speed devices is greatly increased because of this direct exchange feature.

The Nova 800 is also available in an enclosure with two central processor board assemblies and 15 additional spare assembly slots, or ten more spare slots than the basic enclosure. This extra large enclosure is available under Data General option 8202. The other Nova 800 enclosure configuration available is the "Table Top" cabinet (under option 8205). The technical data provided within this publication is applicable to all three Nova 800 enclosure configurations.

Nova 800 input/output instructions perform data transfers to and from peripheral equipment. The I/O instruction format allows 64 device code definitions, of which

(octal) code 0 is not used and (octal) 77 is reserved for special functions. A 16-level programmed priority interrupt facilitates handling 16 different device speed classes within the interrupt control structure. Interrupts are enabled or disabled by a processor word, of which each bit position exercises disabling control over (the interrupt logic) the devices assigned to that bit position. In terms of Interrupt timing, the time a device must wait depends on the number of devices capable of producing interrupts, the length of service routines for devices of higher priority, and whether the data channels are in use. Excluding the execution of indirect memory reference instructions the maximum interrupt waiting time is approximately 4.6 μsecc. (without Multiply-Divide). The instruction times for the Nova 800 are listed in Table 1-1.

1-2.1 Computer Organization

The particulars covered in this paragraph are addressed primarily to the unique features of the Nova 800 architecture rather than to the fundamental operations characteristic of all general purpose Computers. Figure 1-1 is a general block diagram of the Nova 800 Computer. This diagram serves two important purposes; of which the first shows the function relationships of the major Nova 800 logic sections, the second identifies the physical locations of the major logic section (relative to the Printed Circuit board assemblies). Each printed circuit board assembly is shown as a dash line enclosing the logic sections physically mounted on that particular board. For example, the CPU-1 PCB assembly contains four major logic sections; the Arithmetic and Logic Unit (ALU) Control section, the Timing section, the Instruction Register (IR) & (Major) States section, and the Input/Output (I/O) section. The functional data paths for parallel (word) data is depicted on the drawing by the heavy flow lines. A line with an arrowhead termination on both ends denotes a bi-directional data path. The function of each one of the major Nova 800 data paths is briefly described in the following discussion.

The Nova 800 is organized around three data busses, each of which is 16 bits wide. The three busses are identified as the MEM (Memory) bus, the MBO (Memory Buffer Output) bus, and the I/O (Input/Output) bus. The primary function of the MEM bus is to carry data from the memories to CPU 1 & 2. This data may be either instructions or operands. Only one memory is allowed to use the MEM bus at a time, that being the memory which is

Table 1-1. Nova 800 Instruction Execution Times (in Microseconds)

Instruction	Execution Time
LDA	1.6*
STA	1.6*
ISZ, DSZ	1.8*
JMP	.8
JSR	.8
Indirect addressing add	.8
Base register addressing add	0
Autoindexing add	.2
COM, NEG, MOV, INC	.8**
ADC, SUB, ADD, AND	.8**
IO input (except INTA)	2.2***
NIO	2.2***
IO output	2.2***
IO skips	1.4**
INTA	2.2
MUL	8.8
DIV	8.8
Unsuccessful	1.6
Interrupt	1.6
Latency	
With multiply-divide	10.6
Without multiply-divide	4.6
Data Channel	
Input	2.0
Output	2.0
Increment	2.2
Latency	3.6
High speed channel	
Input	.8
Output	.8/1.0◇
Increment	1.0/1.2◇
Latency	
With IO	3.6
Without IO	2.0

*Times are for core; for read-only subtract .2 except subtract .4 for LDA, STA, ISZ, DSZ if reference is to read-only memory.

**If skip occurs add .2 microseconds.

***For IO Control Functions S, C or P add .6 microseconds.

◇When two numbers are given, the one at the left of the slash is the time for an isolated transfer, the one at the right is the minimum time between consecutive transfers.

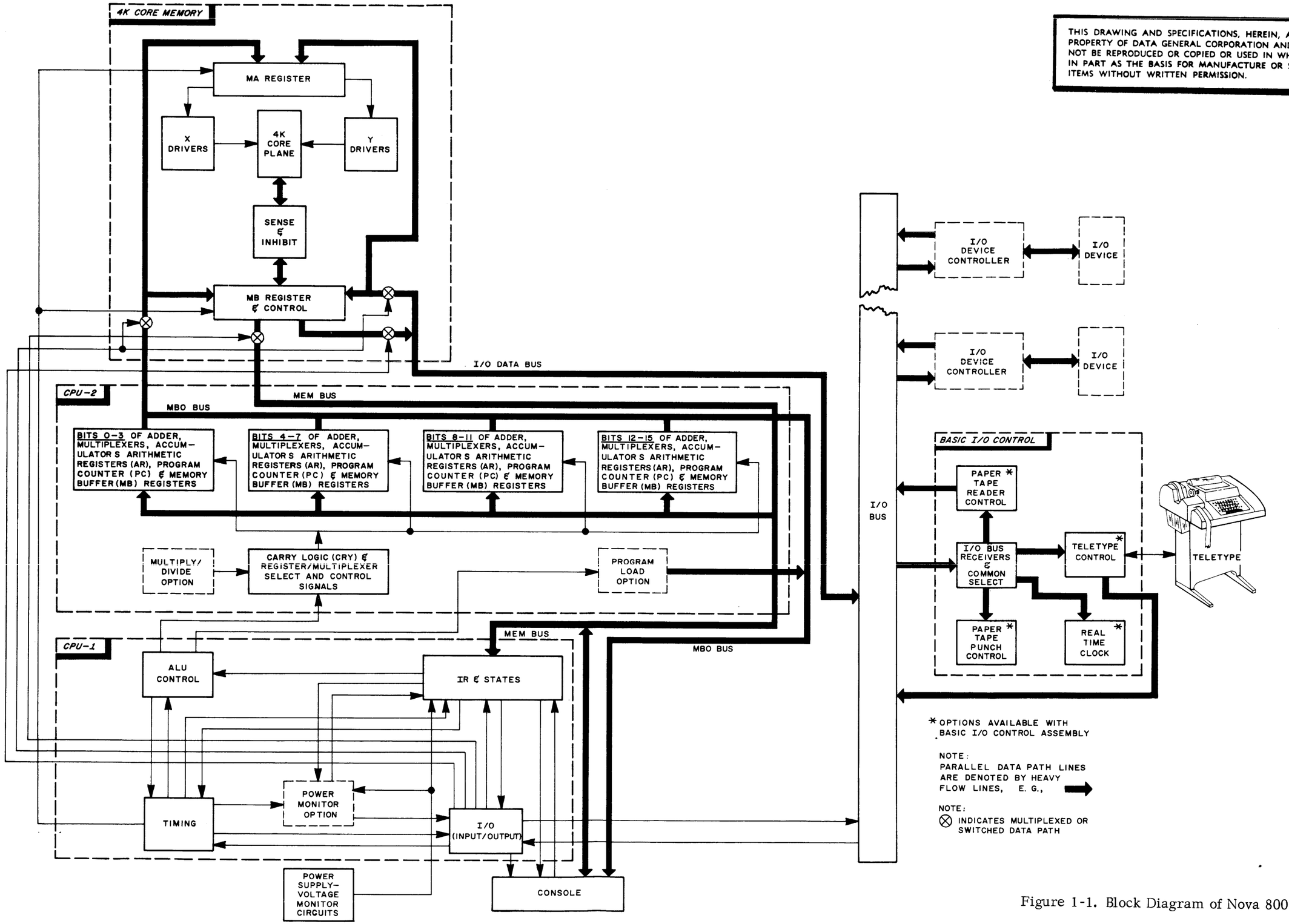


Figure 1-1. Block Diagram of Nova 800

selected. The MEM bus is also used during programmed I/O input operations. Data from an I/O device controller is loaded into the MB register of the selected memory, and transferred from the MB to CPU 1 & 2 via the MEM bus. The MEM bus also transmits data from the console to the CPU. Console data may be generated either by the console data switches, as during a READS, start, examine, etc., or by the console function switches. In the latter case, the console function to be performed is encoded into bits placed on the MEM bus, from which it is loaded into the instruction register (CPU-1) and executed as a single instruction. As data is read from the Console onto the MEM bus the selected memory is prevented from also placing data on the bus by an INH TRANS signal generated by the I/O section on CPU-1.

The MBO bus is the data path for transmitting CPU data to either memory or I/O device controllers. During a programmed I/O output operation, data is first transmitted to the MB (Memory Buffer) of the selected memory via the MBO bus. The information is then transferred from the MB out onto the I/O bus. Hence, the MBO bus is not physically connected directly to any I/O device Controller, but transmits output data through the MB. In the sequence of memory operations, the MBO bus transmits the address of the desired memory location to the selected memory MA register for loading. After the address data has been loaded into the selected MA register, the MBO bus will carry the data to be loaded into that address if the instruction requires a memory modification, as in a STA, or a console deposit operation. The MBO bus is also used in conjunction with control logic functions (PL or PI) to generate a zero address function during the last cycle of a Program Load sequence, or during the cycle preceding a PI (program interrupt cycle.)

The I/O bus differs from the other two bus paths in that it is a bi-directional path between memory and I/O device controllers. The I/O bus communicates directly with memory and no logical connection is made to the CPU's (although the bus is terminated at CPU-2). The data flow for programmed I/O operations is the same as described previously, since the I/O bus receives data from the MB register of the selected memory. During Data Channel (DCH) operations data flow is between the I/O device Controller and memory, with the CPU's completely bypassed.

As shown on Figure 1-1, the three following major Printed Circuit Board (PCB) assemblies make up the Nova 800 Computer, CPU-1, CPU-2, and the 4K Core Memory PCB assembly. CPU-1 contains four logic sections; the ALU Control, IR & States, Timing, and the I/O section. During a Fetch cycle the instruction retrieved from memory is loaded into the IR for decoding. Depending on the type of instruction decoded the operational flow sequence of the machine will either switch to the Defer state, the Execute state, or remain in the Fetch state at the end of the current Fetch cycle. The major states logic of this section indicates the present operational state of the machine and, in conjunction with the decoded instruction data, selectively switches to the next operational state. For example, if the instruction Fetched is an LDA with an indirect address, the Defer state will be the next state entered after the Fetch cycle. (Reference the Nova 800 Timing and Flow Chart, Sheet 1, Drawing No. 001-000121, bound into the Reference Drawings Section, Volume III of this manual under separate cover.) All of the Console Control switch functions logically terminate in this section. This logic is used to detect when a Console Key is actuated, and sets up conditions for executing a Key cycle (e.g., AC Examine, AC Deposit, or Start) and a KeyM (Memory) cycle (e.g., Deposit, Deposit Next, Examine, Examine Next, Program Load) if one is required. Once a Key has been seen, and it is not a Continue, Instruction Step, or Memory Step Key, a $\overline{\text{CON INST}}$ (Console Instruction) signal is transmitted to the Console. $\overline{\text{CON INST}}$ allows the Console control switch data to be placed on the MEM bus. Two power supply monitor signals + 5 OK and MEM OK are also routed to the IR & States section. If the MEM OK (Memory Voltage) signal drops low a STOP RQ (Request) signal will be generated. This stops the Processor synchronously with the end of the present instruction. + 5 OK signal, on the other hand, is sensing the + 5 volt logic power. If this signal drops low the entire Computer is reset immediately.

The Timing section of CPU-1 provides all of the processor and memory timing signals. Basic memory Read, Write, Strobe, Inhibit Controls signals are also produced in the Timing section. This section also contains a 20 MHz crystal controlled oscillator which functions as the primary CPU and Memory clock source. The PTG (Processor Timing Generator) and TS (Time State) signals are also produced in this section by count-down decoding logic.

The ALU Control section converts the various IR decode and States logic signals into

control signals for the Adder, registers or multiplexers on CPU-2. The primary control signals for the MBO bus originate in the ALU Control section. The ALU Control also provides a signal to the Timing section for disabling the memory strobe function during either Program Load, a (Console) Key operation, or during a Read I/O operation.

The I/O section generates the device select code ($\overline{DS0}$ thru $\overline{DS5}$) for selecting the various (optional) I/O device controllers present on the I/O bus. This section decodes other IR signal lines to produce all of the other I/O bus output signal functions, e.g., DATIA, DATOA, INTA, IORST, MSKO, etc. Data Channel input-output control signals are also handled by the logic of this section. Three important bus control signals are produced in this section, specifically READ IO, DRIVE IO, and INH TRANS (Inhibit Transmission). All three signals are routed to the Memory assembly. The Read IO signal allows data on the I/O bus lines ($\overline{DATA0}$ thru $\overline{DATA15}$) to be loaded into the MB register. The Drive IO signal allows the data temporarily stored in the MB register to be gated out onto the I/O bus lines ($\overline{Data0}$ thru $\overline{DATA15}$). The INH TRANS signal prevents data present in the MB register from being gated out onto the MEM bus lines ($\overline{MEM0}$ thru $\overline{MEM15}$). INH TRANS is generated as an Inhibit function either during a Program Interrupt, Key operation, Restart operation, or CPU Instruction. In this regard the IO logic also produces a $\overline{CON\ DATA}$ signal from the CPU Instruction decode. CON DATA enables the configuration set into the Console Data switches to be placed on the MEM bus.

The Power Monitor option, if selected, is also located on CPU-1. The power monitor causes a program interrupt (for programmed housekeeping) when power fails and provides an automatic restart when power is restored.

CPU-2 contains the ALU which consists of a 16 bit Adder, four 16 bit accumulators (AC0 thru AC3), a 16 bit AR register, a 15 bit Program Counter, a 16 bit (Processor) Memory Buffer, a Source Multiplexer, a Destination Multiplexer, and a Memory Buffer Output (MBO) Multiplexer. CPU-2 also contains the Carry logic for the Adder, and the register-multiplexer select and control logic for the remainder of the ALU. The four accumulators are configured from eight 4 X 4 storage devices. Since four of the devices function as the Source Accumulators and the remaining four devices function as the destination accumulators, the accumulators may

be regarded as four, 4-bit chips wide by four files deep. A selective code produced in the Carry logic (CRY) section select one of the four accumulator (files) for reading or writing. Multiplexer selection codes, shift control signals and other complex gating and complementing functions are controlled by signals from the Carry logic section. The MBO multiplexer is enabled to selectively place either the PC outputs or the Adder outputs on the MBO bus to Memory and the Console (display). The outputs from the Adder are gated (by the appropriate control signals from the Carry Logic) out to a 16 bit SUM bus. This bus is internal to the ALU and drives the inputs of both the PC and AR registers. Thus, the results of effective address calculations performed through the Adder can be loaded into PC via the SUM bus. The results of Accumulator orientated operations (Arithmetic, logic, or data transfer) are loaded into the AR via the SUM bus. The MEM bus data is also input to the ALU, and drives a "local" MB register. The data stored in the (ALU) MB register is multiplexed into the Adder via the D-Multiplexer. The two other inputs to the D-Multiplexer are the MQ (Multiplier/Quotient) inputs from the optional Multiply/Divide logic, and the Destination Accumulator (AC0 thru AC3) outputs. Hence, the D-Multiplexer can be selectively enabled to supply the Adder with either the MB outputs (from the MEM bus), the MQ outputs (from Multiply/Divide Option) or one of the four Destination AC's. The other input to the Adder, the SX-Multiplexer is selectively enabled to "look at" either the PC outputs the AR outputs or one of the four Source AC's.

The Program Load option is also located on CPU-2 and is a Read Only Memory, which enabled to drive the MBO bus directly. Each word of Program Load (Bootstrap) data is transferred from the MBO into the MB register of the 4K Memory, and written from this point into core. At the start of Program Load the Memory MA register is forced to zero, and the read-only data is loaded (via the MBO bus) into locations 0 to 37 (octal). At the conclusion of a Program Load the PC and the MA are both zeroed to start the next Fetch from Memory location 0.

The 4K Core Memory assembly contains the core plane, X and Y drivers Sense & Inhibit logic, a Memory Address register (MA), and a Memory-Buffer (MB) register and control. The X and Y driver logic is driven by the Read and Write current signals from the MA & MB Register & Control section. The outputs from Memory Address register are decoded to generate the X and Y selection signals. The memory address register functions primarily as

a buffer register for the X and Y selection logic. The Memory Buffer register is used as a buffer register for data entering or leaving memory. Output data from the Adder is loaded into this register in preparation for writing it into memory. The outputs from this register control the Inhibit Circuitry which either enables or disables core storage as determined by the state of the corresponding Inhibit register flip-flop. When data is read out of memory, the Sense Amplifiers unconditionally set up the MB register. The outputs from the MB register are gated out over the $\overline{\text{MEM}}$ lines back to the ALU section of CPU-2, the IR & States section of CPU-1, and the Console display indicators. The MB register also hold I/O Data either entering or leaving memory. I/O Data is gated into the MB register from the $\overline{\text{DATA}}$ bus by a $\overline{\text{READ IO}}$ signal. I/O Data is gated out from the MB register to the $\overline{\text{DATA}}$ bus by a $\overline{\text{DRIVE IO}}$ signal. Both the $\overline{\text{READ IO}}$ and $\overline{\text{DRIVE IO}}$ signals originate in the I/O logic of CPU-1. As mentioned previously, the $\overline{\text{READ1}}$, $\overline{\text{READ2}}$, STROBE, MB LOAD, INHIBIT, MA LOAD, $\overline{\text{MB CLEAR}}$ Memory control signals are generated in the Timing logic of CPU-1.

The Basic I/O Control assembly contains provisions for four optional device control logic sections which may be added to this basic optional assembly as an additional purchased option. However, the I/O Bus Receiver & Common Select is standard with the basic option and contains bus gates which are common to each additional control logic section. In this particular case, it is assumed that the Teletype Control has been included as an option. The Teletype Control contains Busy, Done and Interrupt logic for controlling the operational data transmissions exchanged between the Teletype and the Processor. The Control also contains a teletype clock (for synchronizing transmission rates with the teletype), a serial to parallel converter, and a parallel to serial converter. The serial to parallel converter receives the serial input data from the teletype keyboard (or reader) and sequentionally shifts it into an 8 bit register. Once the register is loaded an interrupt is generated (if the interrupt request is enabled) requesting the Processor read in the data. The parallel to serial converter is loaded with Processor data from the bus, and this data is subsequently serially shifted from the register out to the Teletype Printer (or punch). Generally speaking, the I/O Bus signals generated by the Input/Output section of CPU-1 are present on the bus simultaneously with the I/O data entering or leaving memory. For example, in performing a DOA instruction

to send data to the Teletype, IR bits 5, 6, and 7 would be decoded in the Input/Output logic to generate bus signal DATOA. If AC0 was designated as the accumulator holding the output data, the outputs of AC0 would be selectively transferred through the Adder, and the MB0 bus into the MB register of Memory. A subsequent $\overline{\text{DRIVE IO}}$ signal would then transfer this information from the MB out via the $\overline{\text{DATA}}$ bus lines to the Teletype. The device code for the Teletype Output, Octal 11 would also be present on Bus lines $\overline{\text{DS0}}$-$\overline{\text{DS5}}$. These signals are also generated in the Input/Output logic section and are derived from IR10 thru IR15 of the Instruction register. (Residual holding facility during the present execution of the I/O instruction just read from core.)

1-3 PHYSICAL DESCRIPTION

The Nova 800 Central Processor by definition consists of the Console/Enclosure Unit, Power Supply Unit, and the CPU-1, and CPU-2 printed circuit board assemblies. An outline drawing of the printed circuit boards used in major Nova 800 assemblies is provided in Appendix A of the "How to use the Nova Computers" reference manual. The Console/Enclosure Unit is so designed that seven 15 X 15 inch Printed Circuit Board (PCB) Assemblies may be plug mounted into a special printed circuit board connector in the Enclosure Chassis. The board assemblies are inserted horizontally into the Enclosure Chassis. A pair of guiding rails are built into the chassis frame (on each board level) to insure proper insertion of the board contacts into the corresponding socket of the multiple printed circuit board connector. The seven connector slots are numbered from the bottom of the chassis up to the top with the first slot reserved for the CPU-1 PCB assembly. The next slot upward is reserved for the CPU-2 assembly. The five remaining slots may be used for memory assemblies, I/O assemblies, or special control board assemblies. Figure 1-2 is a drawing showing the major components of the Nova 800 Computer. The Power supply is mounted in the rear of the Console/Enclosure Unit and contains one fan mounted in the center of the supply for cooling. The Input/Output connector panel for the processor is mounted beside the power supply in a slightly recessed position. When optional peripheral equipment is purchased with the Nova 800, the required I/O connector is mounted on Input/Output connector panel and wired into the

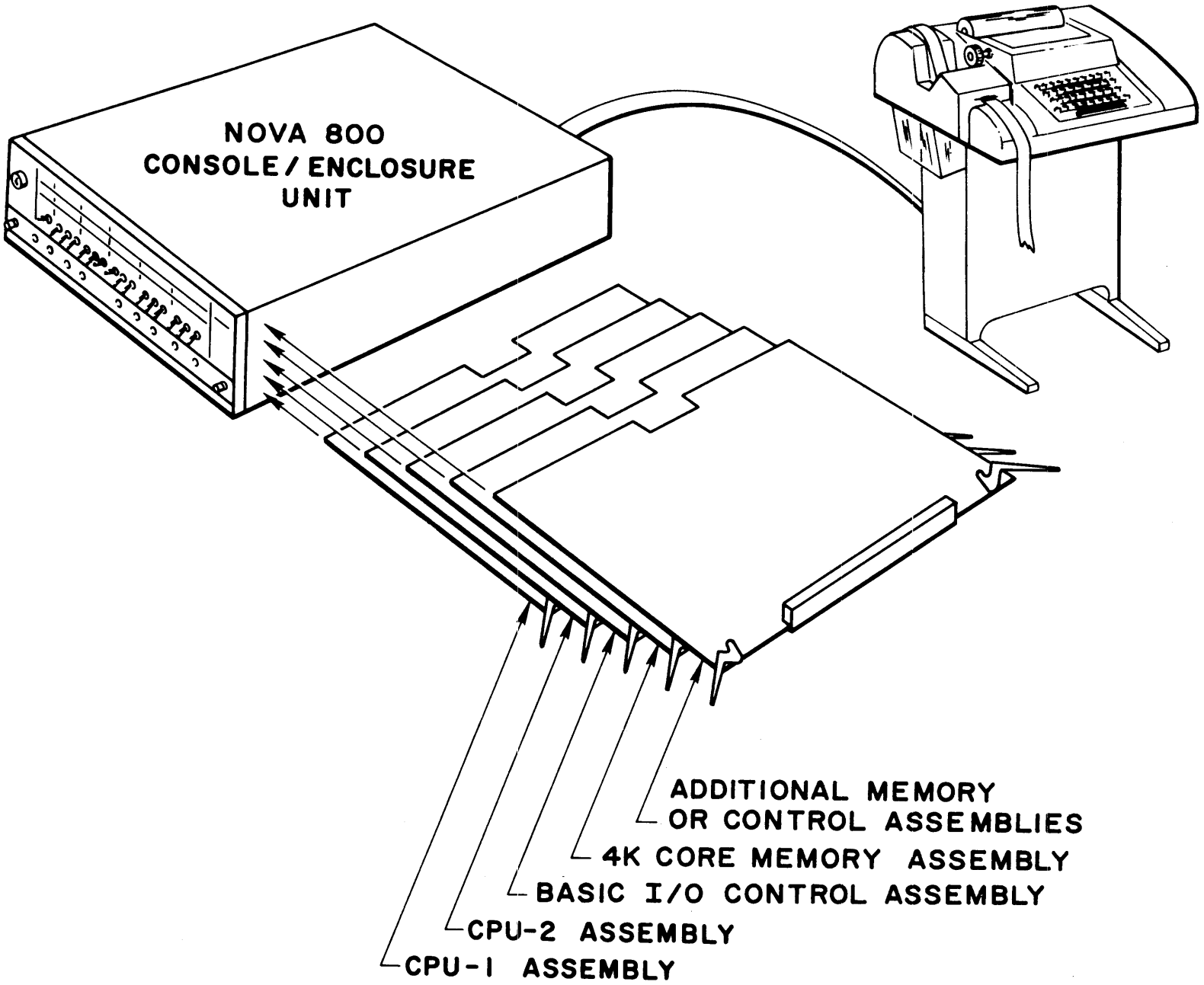


Figure 1-2. Major Components of the Nova 800 Computer

multiple printed circuit board connector. (This wiring is direct from the pins of the selected PCB socket to the pins of the corresponding I/O connector.) It should be noted once a socket is wired for a specific Control PCB Assembly (used to control the optional peripheral device) that particular socket is dedicated to that purpose and must not be used for any other (different type) Control PCB Assembly. For example, a 4K Memory PCB Assembly can be mounted in any spare socket and will work properly. However, the Device Controller PCB Assemblies, on the other hand, are each hardwired from the selected socket position to the output I/O connector and each Controller assembly must be mounted in the (multiple printed circuit board) socket wired for it.

The wiring terminations for the major assembly components of the Nova 800 are brought out to printed circuit edge type connectors. These connectors are mechanically aligned (on the abutting side of each major component to be assembled) so as to home with its connector mate on the adjacent assembly component. This interconnection facility replaces bulky interconnection cable runs and permits easy quick removal or replacement of any major assembly component. Figure 1-3 is a simplified diagram depicting the plugging sequence for assembling the Nova 800. The plugging sequence for a complete processor assembly is performed as follows:

a) The Multiple PCB connector is hardware fastened to the Enclosure Chassis as the first step.
b) The plug receptacle on the end of the Console assembly is then mated with the edge connector on the end of the Multiple PCB Connector, and hardware secured to the Enclosure Chassis when properly positioned.
c) The edge connector of the Resistor Board subassembly is then plugged into the receptacle connector on the opposite end of the Multiple PCB connector and hardware secured to the Enclosure Chassis when properly positioned.
d) The edge connector of the Power Supply Unit is then plugged into the receptacle connector mounted on the Resistor Board subassembly and hardware secured to the Enclosure Chassis when properly positioned.

All that remains to make the processor operational is to plug the CPU-1, CPU-2, Memory, and I/O PCB assemblies into their respective slot locations within the Multiple PCB connector, and plug in the power cord. Table 1-2 is a summary of the Nova 800 Physical Characteristics.

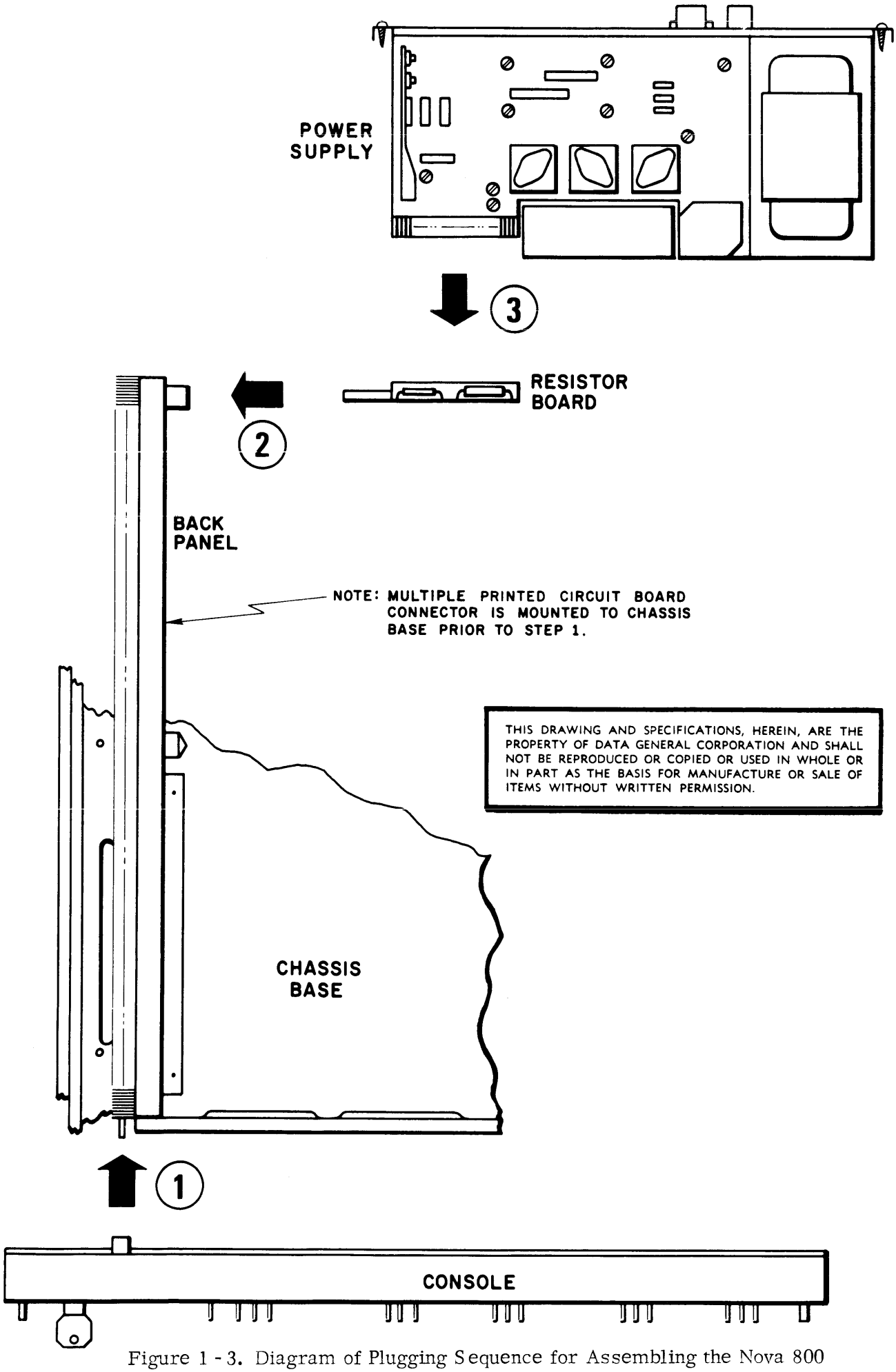


Figure 1 - 3. Diagram of Plugging Sequence for Assembling the Nova 800

Table 1-2. Nova 800 Physical Characteristics

Physical Specifications

Enclosure Height:	5 1/4 inches
Enclosure Width:	19 inches
Enclosure Depth:	21 1/4 inches (23 with Console)
Weight:	50 lbs.
ASR33 Teletype Height (on stand):	45 inches
ASR33 Teletype Width:	22 inches
ASR33 Teletype Depth:	19 inches
ASR Teletype Weight:	56 lbs.

Electrical Specifications

Power Requirements:	115v or 230v* single phase, ±20% 47 to 63 Hz, Minimum Current** 2.2 amperes (250 watts) maximum Current 3.1 amperes (350 watts)
ASR33 Teletype:	115v single phase, 2 amperes. turn on surge 7 amperes. 92 watts.
Bus Signals	Low = 0 volts to + .4 volt nominal High = + 2.2 volts to 3.0 volts nominal
Processor Logic Signals	Low = 0 volts to + .4 volt nominal High = + 2.5 volts to 5.0 volts, + 3.5 volts nominal
Power Supply Voltages + 5 Volt Output:	+ 5 volts (nom.), + 4.7 volts low limit, + 5.45 high limit @ 25°C. Temperature variation: From + 5.1v @ 25° C to + 5.0v @ 55°C Typ.
- 5 Volt Output:	- 5 volts (nom.), -4.5 volts high limit, - 5.45 low limit @ 25°C. Temperature variation: From - 5.0V @ 25° C to - 5.1V @ 55°C Typ.

*230v on Special order

**Based on Minimum Computer Configuration of Processor, teletype interface, and 4K of memory.

Table 1-2. Nova 800 Physical Characteristics (Continued)

Indicator Lamp Voltage ($+V_{Lamp}$)	+ 10.5 + 13.5 Volts
	Temperature variation: N/A
Memory Voltage (Formerly + VINH & + VMEM)	*See Memory Drive Specification.* Temperature variation: From + 15.1v setting @ 25°C to 14.5 (Max.) @ 55°C.
Power Supply Currents	
+ 5 Volt Output:	11.25 amps max.*
- 5 Volt Output:	1.0 amp max.
Memory Drive Nominal	+ 15.0
(+ VMEM) X and Y windings:	+ 15.0 Volts, 390 ma @ 25°C
(+ VINH) Inhibit windings:	+ 15.0 Volts, 740 ma @ 25°C
Functional	
Memory Reference Cycle Time	
With Accumulator:	1.6 microseconds
Without Accumulator:	.8 microseconds
Word Length:	16 bits
Core Memory Size:	4096 words expandable to 32,768 in increments of 4096 words

* + 5 Volt current specification is based on requirements of a Nova 800 with one 4K Memory only. Add 3/4 amp (as an approximation) for each additional 4K Memory installed.

1-4 PERTINENT DOCUMENTS

The following documents serve as source material and complement the information in this manual.

How to Use the Nova Computers

Section 574-100-201 of Bulletin 272B
Volume 1, TECHNICAL MANUAL,
32 AND 33, TELETYPEWRITER SETS

Arithmetic Test
DGC Manual 097-000018

Nova 800/1200 Power Shut Down Test
DGC Manual 097-000022

Bootstrap Loader
DGC Manual 093-000002

Binary Loader
DGC Manual 093-000003

Nova 800 Instruction Timer
DGC Manual 097-000024

Nova 800 Teletype Test
DGC Manual 097-000025

Nova 800 Logic Test
DGC Manual 097-000023

Memory Checkerboard III
DGC Manual 097-000014

Exerciser
DGC Manual 097-000004

1-5 ABBREVIATIONS

Listed below are the most commonly used abbreviations of registers, key operations, components, instruction, and signal names. Signal names not included in this list will be found in Appendix B Signal Origins. Appendix B contains an alphanumerical list of all signal names which appear on drawings, together with the drawing number which contains the generating circuits for the signal.

ACD	Destination Accumulator
ACDP	Accumulator Deposit
ACEX	Accumulator Examine
ACEX + ACDP	Accumulator Examine or Deposit
AC-H-WRITE	Accumulator-High Order (address bit)-Write

ABBREVIATIONS (Continued)

AC-L-WRITE	Accumulator-Low Order (address bit)-Write
ACSX	Source Accumulators
AC WRITE	Write Accumulator
AC0	Accumulator 0
AC1	Accumulator 1
AC2	Accumulator 2
AC3	Accumulator 3
ADDER TO MEM	(Transfer) Adder (outputs) To Memory
ADDER = 0	Adder (outputs) equal zero
ALC	Arithmetic Logic Class (instruction)
ALC + IO SET	ALC or IO (instruction) SET (into the IR)
ALU	Arithmetic Logic Unit
AND	AND (logic instruction)
AND ENABLE	AND (instruction) Enable
AR	Arithmetic Register
AUT DEC	Autodecrement
AUT INC + DEC	Autoincrement or Autodecrement
CARRY	Carry (arithmetic function)
CG	Carry Generate (ALU carry function)
CLK-A	Clock A
CLK-B	Clock B
CLR	Clear
CON DATA	Console Data
CON INST	Console Instruction
CON RQ	Console Request
CONT	Continue switch at Console
CONT + ISTP + MSTP	Continue or Instruction Step or Memory Step Console switches

ABBREVIATIONS (Continued)

CPU	Central Processor Unit
CPU CLK	Central Processor Unit Clock
CPU INST	Central Processor Unit Instruction
CR	Carry Ripple (ALU carry function)
CRY	Carry
CRY SET	Carry Set
CRY TEST	Carry Test
CRY TO AR	Carry to Arithmetic Register
D	Defer
DATIA	Data In A (I/O instruction)
DATIB	Data In B (I/O instruction)
DATIC	Data In C (I/O instruction)
DATOA	Data Out A (I/O instruction)
DATOB	Data Out B (I/O instruction)
DATOC	Data Out C (I/O instruction)
DATA0 thru DATA15	I/O Data bus signals, 16 bits wide
DCH	Data Channels
DCHA	Data Channel Acknowledge
DCHA STUTTER	Extends DCHA during certain High Speed Channel operations
DCHI	Data Channel In
DCH INC EN	Data Channels Increment Enable
DCHM(0 or 1)	Data Channel Mode (0 or 1) Code type of Data Channel Cycle requested by Device
DCHO	Data Channel Out
DCHP IN	Data Channel Priority In
DCHP OUT	Data Channel Priority Out

ABBREVIATIONS (Continued)

DCHR	Data Channel Request
DCH SYNC	Data Channels Synchronization
DEFER	Defer (instruction execution state)
DIV ADD	Division Addition
DIV FIRST	First Division (cycle)
DIV LOAD CRY	Load Division Carry
D-L-H-SEL	Destination Multiplexer-Least Significant Byte-High Order (address) Selector Control line
D-L-READ	Destination Accumulator-Low Order (address bit)-Read
D-H-READ	Destination Accumulator-High Order (address bit)-Read
D-L-SEL	Destination Multiplexer-Low Order (address)-Selector Control line
D-M-H-SEL	Destination Multiplexer-Most Significant Byte-High Order (address)-Selector Control line
D-M-COM	Complement Destination Multiplexer outputs
D-MULT	Destination Multiplexer
DP	Deposit (Console function)
DP + DPN	Deposit or Deposit Next
DPN	Deposit Next (Console function)
DRIVE IO	Drive IO (Data bus)
DSZ	Decrement and Skip if Zero (instruction)
DS0-DS5	Device Select lines 0 thru 5
E	Execute
EFA	Effective Address

ABBREVIATIONS (Continued)

EFA•JSR	Effective Address and JSR (instruction)
EXEC	Execute
E•IO	Execute (State) and Input/Output (instruction)
EX	Examine (Console function)
EXN	Examine Next
EXN + DPN	Examine Next or Deposit Next
EX + EXN + DP + DPN + PL	Examine or Examine Next or Deposit or Deposit Next or Program Load (Console Key)
EX + STRT + ACDP	Examine or Start or Accumulator Deposit (Console Key)
EXT ION EN	External "Interrupt On" Enable
F	Fetch
FAST DCH	Fast (High Speed) Data Channels
FETCH	Fetch (State Accessing next instruction from Memory)
FETCH SKIP	Skip the next instruction
F + PI	Fetch or Program Interrupt (Cycle)
F SET	Fetch (State) Set
FORCE AR SHIFT	Force Arithmetic Register (to) Shift
FORCE D-L-SEL	Force Destination Accumulator-Low Order (address bit) to Selectors
FORCE MEM CY	Force Memory Cycle
FORCE MQ OUT & AC-L-WRITE	Multiply/Divide function

ABBREVIATIONS (Continued)

FORCE PLUS ONE	Adds one to Adder
FORCE SEL X	Control line which manipulates Adder output data via the SEL-N, SEL-L, SEL-R & SEL-S lines.
FORCE SEL Y	Control line which manipulates Adder output data via the SEL-N, SEL-L, SEL-R & SEL-S lines.
FORCE-SX-COM	Force Complement Source Multiplexer outputs
FORCE-SX-H-READ	Force Source (Accumulators) High Order (address bit) Read
FORCE SX-L-READ	Force Source (Accumulators) Low Order (address bit) Read
FORCE-SX-H-SEL	Force Source (Multiplexer) High Order (address bit) Selector line
FORCE-SX-L-SEL	Force Source (Multiplexer) Low Order (address bit) Selector line
GND LAMP	Special Ground for Console Display Lamps
HALT	Halt (Machine State)
HAS E CYCLE	Indicates instruction has execute Cycle
INC PC	Increment Program Counter
INH DCH	Inhibit Data Channels
INH GATE A	Inhibit Gate (signal) A (Memory)
INH GATE B	Inhibit Gate (signal) B (Memory)
INH0 thru INH15	Inhibit (Memory Buffer) Register outputs 0 thru 15
INHIBIT	Inhibit (Memory Writing function)
INHIBIT SELECT	Prevents Memory from being Selected
INH TRANS	Inhibit Transmission
INTP IN	Interrupt Priority In (to Device)
INTP OUT	Interrupt Priority Out (from Device)

ABBREVIATIONS (Continued)

INTR	Interrupt (Bus Signal from Device)
INT RQ	Interrupt Request
IO or I/O	Input/Output
ION	Interrupt On
ION SYNC	Interrupt On Synchronization
IO OUT EN	Input/Output-Output Enable
IO PLS	Input/Output Pulse
IORST	Input/Output Reset
IO SKIP	Input/Output Skip (instruction)
IO SKP PEND	Input/Output Skip Pending
IO SKP SYNC	Input/Output Skip Synchronization
IO STUTTER	Cycle extend for IO operation
IO UNPROTECTED	Indicates IR contains IO instruction
IR0 thru IR15	Instruction Register outputs 0 thru 15
ISTP	Instruction Step (Console switch)
ISZ	Increment and Skip if Zero (instruction)
JMP	Jump (instruction)
JSR	Jump to Subroutine (instruction)
JMP + JSR	Jump or Jump to Subroutine (instruction)
KEY	Operational Cycle manually implemented at the Console
KEY M	Key cycle with Memory (access Cycle).
KEY M • PL	Key Memory and Program Load
KEY • PRESET	Key (cycle) and Preset
KEY SEEN + RESTART	Key Seen or Restart (from Power Monitor Option)
LDA	Load Accumulator (instruction)
LOAD AR	Load Arithmetic Register
LOAD CRY	Load Carry
LOAD PC	Load Program Counter

ABBREVIATIONS (Continued)

MA LOAD	Memory Address Load
MA1 thru MA15	Memory Address Register Outputs 1 thru 15
MB	Memory Buffer
MB CLEAR	Memory Buffer Clear
MB LD EN	Memory Buffer Load Enable
MB LOAD	Memory Buffer Load
MBO INH	Memory Buffer Output (bus) Inhibit
MBO0 thru MBO15	Memory Buffer (bus) Outputs 0 thru 15
MD1 thru MD15	Memory (address) Data (input lines) 1 thru 15
MEM CLK	Memory Clock
MEM CY SET	Memory Cycle Set
MEM LATCH	Memory (Register, CPU-2) Latch
MEM OK	Memory OK (Power Supply Monitor signal)
MEM OUT	Memory (bus) Out
MEM 0 thru MEM15	Memory Bus lines 0 thru 15
MID	Midpoint (of 800ns extended DCH or IO cycle)
MQ0 thru MQ15	Multiplier Quotient Register Outputs 0 thru 15
MSKO	Mask Out (instruction)
MSTP	Memory Step (Console switch)
MUL + DIV	Multiply or Divide (instruction)
MUL + DIV DECODE	Multiply or Divide Decode
MUL + DIV TC	Multiply or Divide Terminal Count
NEW CRY	New Carry
NON ACD INST	Non Destination Accumulator Instruction
OMIT STROBE	Omit (Memory) Strobe
OVFLO	Signal to Device that memory location being incremented Via Data Channels has Overflowed

ABBREVIATIONS (Continued)

OVFLO TO DIV	Overflow (signal) to Divide Control
PC	Program Counter
PC CLK	Program Counter Clock
PC TO MEM	Program Counter to Memory
PI	Program Interrupt
PI SET	Program Interrupt Set
PL	Program Load
PL LAST	Program Load Last
PL• LAST WORD	Program Load Last Word
PLUS ONE	Plus One (to the Adder)
PRESET	Preset (Computer initializing signal)
PTG0 thru PTG3	Processor Timing (pulses) 0 thru 3
PWR LOW	Power Low (Power Monitor output signal)
READ CY	(Memory) Read Cycle
READ IO	Read IO (Data bus)
READ 1	Read 1 (Memory Timing signal, CPU-1)
READ 2	Read 2 (Memory Timing signal, CPU-1)
READ 1B	Read 1B (Memory Timing signal, Memory)
READ 2B	Read 2B (Memory Timing signal, Memory)
REAL IO INST	Indicates instruction is not Multiply/ Divide
RELOAD DISABLE	Disable Load inputs of (Memory) MB Register
RESTART	RESTART (power Monitor output signal)
RESTART ENABLE	Signal that permits RST and STOP Console Key functions
RINH0 thru RINH15	(Collector) Resistor, Inhibit Driver
ROM ENABLE	PL Read Only Memory Enable

ABBREVIATIONS (Continued)

RQENB	Request Enable
RST	Restart (Console switch)
RUN	Primary operational requirement for program execution
RUN SET	Input signal for Run Flip-flop
SARD	Select Address
SELB	Selected Busy (Bus signal)
SELD	Selected Done (Bus signal)
SELECT	Decoded (Memory) Select signal
SEL-L	Select Left Shift (Adder output data)
SEL-N	Select No Shift (Adder output data)
SEL-R	Select Right Shift (Adder output data)
SEL-S	Select Swap (Adder output data bytes)
SHIFT AR LEFT	Shift Arithmetic Register Left
SNS0 thru SNS15	Sense Amplifier Outputs 0 thru 15
STA	Store Accumulator (instruction)
STATE SUPPRESS	Supersedes Major States for DCH & certain Key cycle operations
STOP	(Processor) Stop
STROBE	Strobe (signal, CPU-1)
STOP RQ	(Processor) Stop Request
STRB A	Strobe A (Memory Stack)
STRB B	Strobe B (Memory Stack)
STRB C	Strobe C (Memory Stack)
STRB D	Strobe D (Memory Stack)
STRT	Start (Console switch)
SUM CRY	Sum Carry
SUPPRESS	Suppress signal implemented by Multiply/Divide

ABBREVIATIONS (Continued)

SX-COM	Complement Source Multiplexer outputs
SX-H-READ	Source Accumulator-High Order (address bit)-Read
SX-H-SEL	Source Multiplexer-High Order (address bit)-Selector Control line
SX-L-READ	Source Accumulator-Low Order (address bit)-Read
SX-L-SEL	Source Multiplexer-Low Order (address bit)-Selector Control line
SX-MULT	Source Multiplexer
TSM	PTG States 1 or 2, equivalent to 2nd half of TS0 & first half of TS3
TS0	Time State 0
TS3	Time State 3
TT	Teletype
TTI	Teletype In (Teletype Keyboard/ Reader Buffer)
TTO	Teletype Out (Teletype Teleprinter/ Punch (Buffer)
WAIT	Implements Processor pause during High Speed DCH operation
WRITE	Control function, Memory Cycle Timing, CPU-1
WRITE AC	Write Accumulator (logically associated with AC Write signal)
WRITE MEM	Write Memory (enables X and Y Memory drivers)

ABBREVIATIONS (Continued)

WRITE SYNC	Control function, Memory Cycle Timing, CPU-1
XRS	X (plane) Read Source (Memory Stack)
XWS	X (plane) Write Source (Memory Stack)
YRS	Y (plane) Read Source (Memory Stack)
YWS	Y (plane) Write Source (Memory Stack)
32 VNR	+ 32 Volts, Not Regulated
±SL0 thru ± SL15	Memory Stack Bipolar sense inputs to Sense Amplifiers
+ VINH	+ (Memory) Inhibit Voltage
+ V_{Lamp}	+ Lamp Voltage (Console indicators)
+ VMEM	+ Voltage Memory
+ 5 OK	+ 5 Volt (power) Operating properly
=0 ENABLE	Enables "Adder =0", gates for AC =0 and for auto indexing addressing

SECTION II

INSTALLATION

2-1 GENERAL

This section provides detailed information and procedures for installing the basic Nova 800 Computer. The Computer and teletype are shipped in separate containers. Prior to performing any installation procedures inspect both shipping containers for any visible intransit damage such as would result from dropping or being punctured or crushed. Contact the carrier and Data General immediately if any damage is discovered, specifying the nature and extent of damage. Physical installation data and descriptions are provided in Appendix B of the "How to Use the Nova Computers" reference manual.

2-2 UNPACKING INSTRUCTIONS

The following two paragraphs describe the proper method of unpacking the Nova 800 Computer and an ASR33 Teletype. The first paragraph describes the approved procedures for unpacking the Computer. The second paragraph describes the procedures for unpacking the Teletype. It is recommended that all shipping hardware, shims, packing and carton be saved and stored after unpacking in the event either machine is ever reshipped.

2-2.1 Unpacking the Nova 800

a) Lift inner carton from box (two people, Jumbo Nova only).
b) Remove strapping and cut tape.
c) Remove styrofoam top cover.
   Note: On Jumbo Nova, only remove styrofoam expansion rings. Computer should now be exposed.
d) Remove attached hardware (keys, mounting hardware, etc.).
e) Lift the unit from the carton. (Requires two people.)
   Do not lift from the sides of the Computer. Hands should be placed on the rear and underside of the Power Supply and by the front of the console. Check unit for shipping damage. Remove keys from plastic bag, insert the key, turn completely counterclockwise to the "Off" position.
f) Remove all packing material and General Purpose frames from the system. Standard circuit boards should not be removed.
g) The Computer is ready to apply power. It is suggested that the operator read the procedures listed under the Nova 800 Start-up and Checkout paragraph of this Section before applying power to the machine.

2-2.2 Unpacking the ASR33 Teletype

The complete ASR33 is packaged in one carton. After opening the ASR33 shipping container perform the steps of the following procedure:

a) Remove Styrofoam pads (2). (See Figure 2-1.)
b) Remove corner braces (4) and Teletype Stand. (See Figure 2-2.) Manuals will be packed inside Teletype Stand.
c) Locate teletype bulletin 273B Vol. 1 in the manual set. Refer to page 1 of the section 574-100-201TC and read unpacking instructions.
d) Remove cardboard insets, accessory kit, and Typing Unit. (See Figure 2-3.) Typing Unit is mounted on a paste board shipping pallet by seven screws.

NOTE: DO NOT USE OR ATTEMPT TO OPERATE TYPING UNIT BEFORE REMOVING THE (3) HEX HEAD BOLTS FROM THE BOTTOM OF THE SHIPPING PALLET.

e) Remove three pieces of adhesive nylon tape; two pieces are securing the paper supply and lid, the other piece is securing the paper tape supply, punch and reader.
f) Remove the Typing Unit Cover to expose the carriage. The carriage is tied to the chassis with a pipe cleaner. This securing wire must be removed before operating the Teletype.
g) Some Teletypes are equipped with a yellow spacer spring holding the reader fingers stationary. This must be removed prior to operating the Teletype.

2-3 NOVA 800 START-UP AND CHECKOUT

The procedures listed below describe the proper methods for initial turn-on and subsequent checkout of the Nova 800 Computer. These procedures should be performed immediately after the Nova 800 has been unpacked. The procedures are listed below in the exact order of performance, and must be performed in the order of appearance, during initial turn-on.

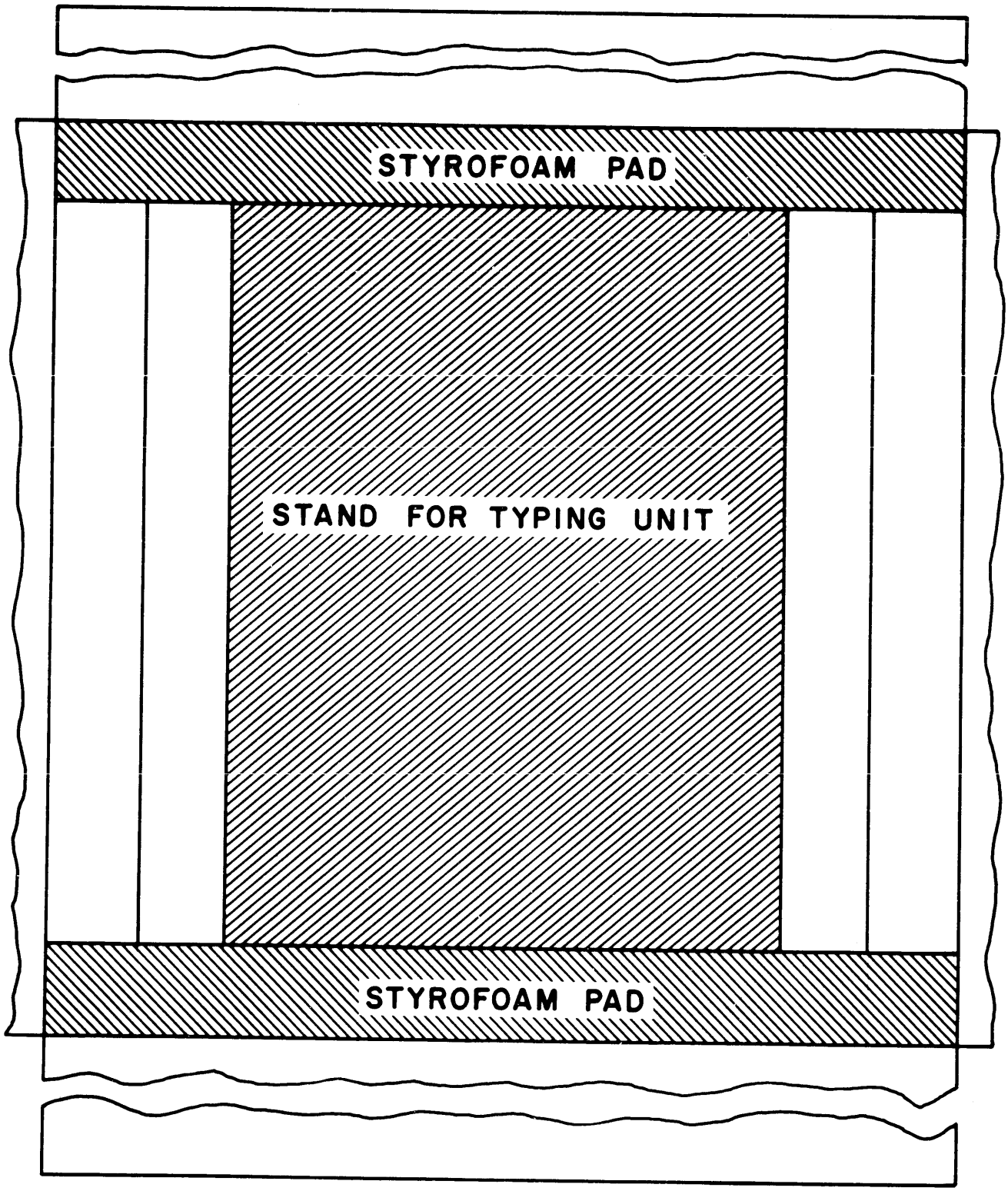


Figure 2-1. Location of Styrofoam Pads

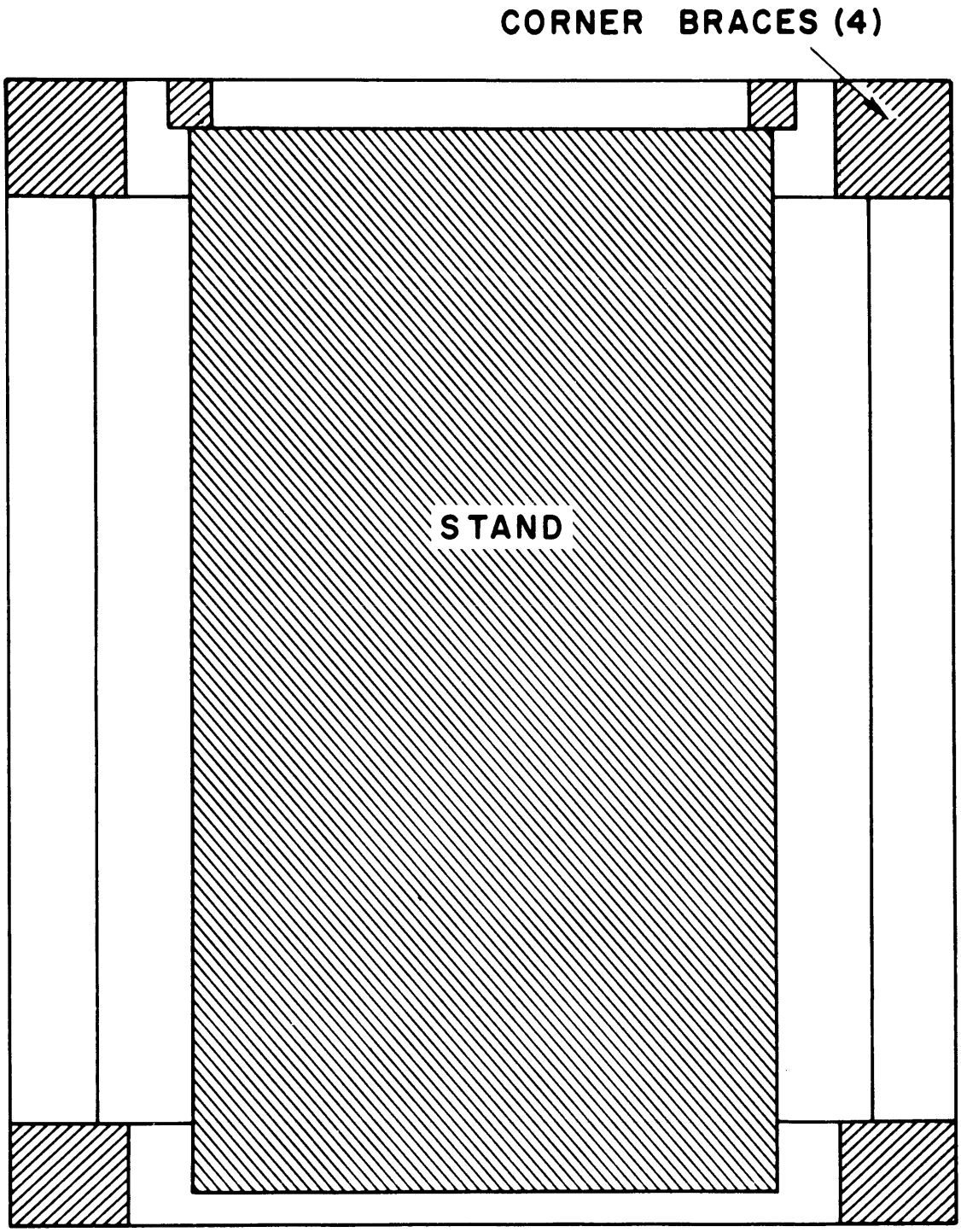


Figure 2-2. Location of Corner Braces in Teletype Carton

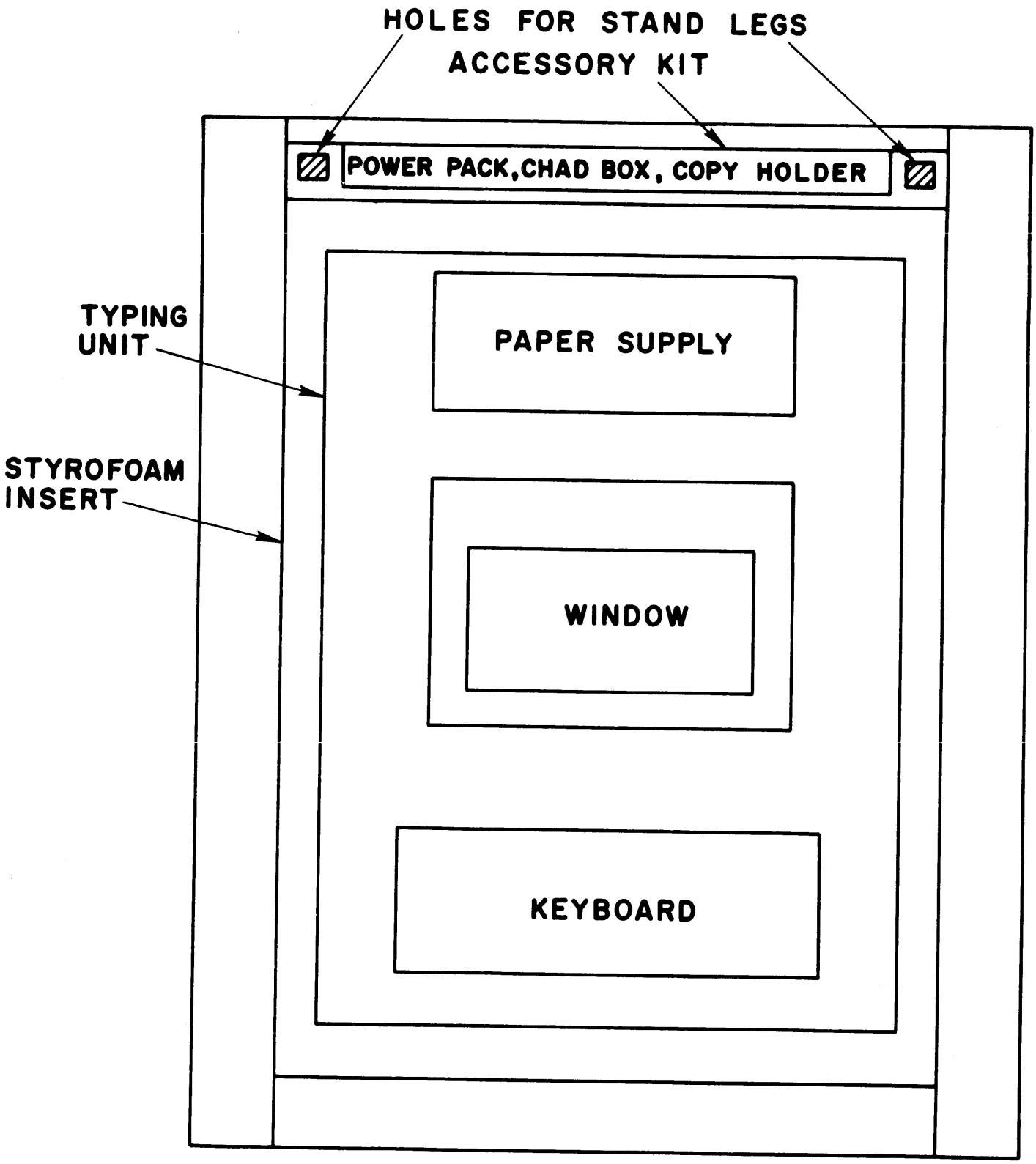


Figure 2-3. Location of Teletype Typing Unit in Carton

2-3.1 Start-Up Procedure

The general purpose Exerciser (Program Listing and Tape #095-000012, Manual #097-000004) is always the last program run in each Computer prior to shipping. The Exerciser diagnostic program checks the entire instruction repertoire and all memory locations associated with that particular processor system. This program should still be intact within the memory and available for execution. To start this program, only Console data switch 14 on the operator's Console should be raised. Turn power "ON" by setting the key to the vertical position. Raise the Reset/Stop switch momentarily to the Reset position and then raise the Start switch momentarily to the Start position. Program should now be in execution with both the Fetch and Run indicators on. With typical operation of the program, an incrementing sequence from $1,000_8$ to approximately $3,400_8$ will be observed in the Data Register display. The cycle takes approximately 15 seconds for a 4K system and becomes significantly longer depending on a maximum core size of the system. Any halt of the Computer and improper indications constitutes an error. Should you encounter any difficulty with the start-up procedure, please contact the Data General representative in your area, or our Field Service Department at the Southboro factory (Area Code: 617-485-9100).

2-3.2 Check-Out Procedures

The Nova 800 check-out sequence consists of static and dynamic tests. Static tests are performed manually at the Operator's Console. Dynamic tests, on the other hand, are a series of tests performed under program control, and either terminate successfully or halt at some specific location to indicate detection of some failure by the diagnostic. In the normal sequence of testing, the static tests are performed first to verify all of the manual controls are working properly. Once the Console is verified as operational, the dynamic testing may be performed. Successful completion of all the recommended diagnostic tests should be considered verification of the first check-out step. Complete verification is obtained when all of the diagnostic program tapes supplied with the documentation package have been run successfully.

2-3.2.1 Static Tests

a) Turn power on and check that the fan is operating.
b) Verify console indicators operate properly.
c) Deposit and examine all zeros in Memory location 0.
d) Deposit and examine all ones in Memory location 0.
e) Deposit and examine all zeros in Accumulators ACC 0-3.
f) Deposit and examine all ones in Accumulators ACC 0-3.
g) Deposit 000017 in ACC 0.
h) Deposit 000360 in ACC 1.
i) Deposit 007400 in ACC 2.
j) Deposit 170000 in ACC 3.
k) Examine all accumulators and verify no data changed.
l) Deposit all zeros in Memory location 0.
m) Momentarily raise the Start/Continue switch to the Start position. Verify run indicator is on.
n) Lock computer. Verify reset/stop toggle is functionally inoperative.
o) Unlock and stop computer.
p) Continually depress DEPOSIT NEXT. Verify PC increments.
q) Continually depress EXAMINE NEXT. Verify PC increments.
r) Place all zeros in Memory location 0.

2-3.2.2 Dynamic Tests. At this point and prior to performing the first dynamic test it is necessary to connect the Teletype to the Nova 800. These procedures assume all of the unpacking steps for the Teletype listed in paragraph 2-2.2 have been completed.

2-3.2.2.1 Teletype to Computer Connection Procedure

a) Turn off computer.
b) Plug in TTY to 115v outlet in the rear of the Processor Enclosure.
c) Plug in the 9 pin connector to the receptacle indicated by the connector layout diagram attached to the rear of the Nova 800 Enclosure. The proper connector is labeled 4010 (Data General Model number for TTY).
d) Turn the line/local switch on the lower-right-front-panel of TTY to local.
e) Place roll of tape in punch and turn on punch.
f) Type all characters on keyboard. Note correct typing and also that punch is operating. It is not necessary to verify tape produced. That will be done in the sequence of tests that follow.
g) Turn punch off and return line/local switch to line. The unit is now ready for use by the computer.

The next step in the dynamic test sequence is to place the Binary Loader program into core. This may be done either manually or automatically. Automatic loading requires that the Program Load Option be included with the processor. This option

operates in conjunction with the Nova 800/1200 Program Load tape supplied as part of the Program Load Option. The Program Load Read-Only hardware loads the Bootstrap program into memory location 0-37. The Bootstrap program is executed immediately and reads a Prologue (to the Binary Loader) into locations 40 to 120. The Prologue is then executed immediately after loading, and thereby loads the Binary Loader from the last section of the Program Load Tape. Paragraph 2-3.2.2.2 describes the loading procedures for those Processors with the Automatic Program Load option. Paragraph 2-3.2.2.2 describes the manual loading procedures for Processors without the Automatic Program Load option.

2-3.2.2.2 Loading Procedures for Program Load Option

a) Turn on Computer. Verify Teletype is on-line.
b) Set Teletype reader switch to FREE.
c) Thread the Nova 800/1200 Program Load tape into Teletype reader, and set reader switch to START.
d) Set the TTI device code 010_8 (or 012_8 for High Speed Paper Tape Reader) into the six rightmost Console switches (bits 10-15).
e) Raise the Program Load switch on the Console to the Program Load position.
f) Tape should move through the Teletype reader. When the tape halts verify Console Address register displays 00120_8 as the halt location.
g) Set Teletype reader switch to FREE, and remove the Nova 800/1200 Program Load tape from Teletype reader.
h) Thread the test program tape Checkerboard III (Binary tape #095-000031, Manual #097-000014) into the Teletype reader, and set reader switch to START.
i) Press the Start/Continue switch on the Console to the Continue position.
j) Verify tape moves through the Teletype reader. When tape halts Checkerboard III will be loaded and ready for execution.
k) It should be noted that for subsequent loading of other programs after a program other than the Binary Loader has been executed, first thread the program tape in the Teletype reader, then load X7777 into the console switches and raise the Start/Continue switch to Start. This will rerun the Binary Loader and bring the new program into core.

2-3.2.2.3 Loading Procedures Without Program Load Option

a) Using the Console switches, key in the Bootstrap Loader Manual #093-000002 instructions listed below.

BOOTSTRAP LOADER

TTI:	** = 10	PTR:	** = 12	
X7757	126440	GET:	SUBO	1,1
X7760	0636**		SKPDN	X

```
X7761    000777            JMP     . -1
X7762    0605**            DIAS    0, X
X7763    127100            ADDL    1, 1
X7764    127100            ADDL    1, 1
X7765    107003            ADD     0, 1, SNC
 TTI:
X7766    000772            JMP     GET+1
X7767    001400            JMP     0, 3
X7770    0601**   BSTRP:   NIOS      X
X7771    004766            JSR     GET
X7772    044402            STA     1, .+2
X7773    004764            JSR     GET
                           ...
                           ...
```

b) Set the Teletype reader switch to FREE and thread the Binary Loader (Special Format #091-000004 , Manual #093-000003) into the Teletype reader (or the High Speed Paper Tape Reader).
c) Verify Teletype is on-line and set the Teletype reader switch to START.
d) Set the Console data switches to X7777.
e) Momentarily raise the Start/Continue switch to the Start position
f) Verify that the tape moves through the Teletype reader. When the tape halts verify Console address register displays X7775.
g) Set Teletype reader switch to FREE, and remove the Binary Loader tape.
h) Thread the test program tape Checkerboard III (Binary tape #095-000031 , Manual #097-000014) into the Teletype reader, and set reader switch to START.
i) Set the Console data switches to X7777.
j) Momentarily raise the Start/Continue switch to the Start position.
k) Verify tape moves through the Teletype reader. When the tape halts verify Console address register displays X7743.

2-3.2.2.4 Memory Test. After the memory test program, Checkerboard III has been loaded as per paragraph 2-3.2.2.2, or paragraph 2-3.2.2.3, perform the following steps to run the test program:

a) Place 000002_8 in Console switches.
b) Momentarily raise the Start/Continue switch to the Start position.
c) Verify program cycle.
d) Raise Console switch 0 = 1 to include worst case. Verify program cycle.
e) Allow program to cycle 15 minutes.
f) Reset the computer.

2-3.2.2.5 Logic Test. Load the test program (Binary #095-000045) as per paragraph 2-3.2.2.2 or paragraph 2-3.2.2.3.

a) Set the Console switches to 000377_8.
b) Momentarily raise the Start/Continue switch to the Start/Continue switch to the Start position.
c) Verify computer halts at location 000400_8. Press the Start/Continue switch to the Continue position.
d) Program run-time for one complete pass is in the millisecond range. Allow the program to run for several minutes, then stop the computer.
e) Note the stop location of the computer, and perform a series of single instructions by toggling the Instruction Step switch. Verify PC follows the program (as listed in program documentation).
f) Toggle the Memory Step switch as in step e. Verify PC follows the program (as listed in program documentation).

2-3.2.2.6 Teletype Test. Load the test program (Binary #095-000048) as per paragraph 2-3.2.2.2, or paragraph 2-3.2.2.3.

a) Turn on Punch of TTY.
b) With the TTY in local, depress "Here is" to generate leader.
c) Return on-line and place tape in TTY Reader.
d) Place reader in start position.
e) Place 000050_8 in Console switches. Raise the Start/Continue switch to the Start position.
f) Program will cycle and type "PASS" on the end pass.
g) Allow program to cycle for 5 passes.
h) Reset the computer.
i) Place reader of TTY in "FREE" position.
j) Place 000055_8 in Data switches and raise the Start/Continue switch to the Start position.
k) After teletype starts punching data, place leader of tape in TTY and push START on TTY reader.
l) Program should cycle for minimum 1 minute.

2-3.2.2.7 Instruction Timer. Load the test program (Binary #095-000046) as per paragraph 2-3.2.2.2, or paragraph 2-3.2.2.3.

a) Set Console switches to 000003_8.
b Momentarily raise the Start/Continue switch to the Start position.
c) Allow the program to run for several minutes, and verify no teletype type-outs occur. Program has a built-in tolerance of ±20 nanoseconds for the execution time of each instruction tested and will print out the time of any instruction exceeding this limit.

d) Depress Console Reset/Stop switch to the Stop position.
e) Set 000002_8 into the Console switches.
f) Momentarily raise the Console Start/Continue switch to the Start position.
g) Starting at location 000002_8 cause the program to type out the execution time for each instruction in nanoseconds.
h) If excessive execution time was detected during step c, contact the Data General representative in your area or our Field Service Department.
i) If step c was completed successfully save the listing of instruction execution times generated during step g. This listing should be filed as part of the maintenance record for the Nova 800. Typical instruction execution times (within ±20 ns) for the Nova 800 are listed below.

INSTRUCTION EXECUTION TIMES

MOV	800
ADD	800
AND	800
COM	1000
LDA	1600
STA	1600
ISZ	1800
DSZ	1800
JMP	800
JSR	800
LDA	1600
LDA	2400
LDA	2600
LDA	2600
LDA	3200
DIA	2200
DOA	2200
NIOS	2800
INTA	2200
SKPBN	1400
SKPBZ	1600
*DIVIDE	8795
*MULTIPLY	8795

*If Mul/Div. option installed.

2-3.2.2.8 Arithmetic Test. Load the test program (Binary #095-000037) as per paragraph 2-3.2.2.2, or paragraph 2-3.2.2.3.

a) Set Console switches to 000002_8.
b) Momentarily raise the Start/Continue switch to the Start position.
c) Upon starting program should issue a message stating "Last Location in Memory is XXXXX".

d) Verify that the value printed corresponds to the actual System Memory size. If they do not agree an error has occurred.
e) The program will cycle continuously, and type out the word "PASS" on each program iteration.

2-3.2.2.9 Power Shutdown Test (No Power Monitor). Load the test program (Binary #095-000044) as per paragraph 2-3.2.2.2, or paragraph 2-3.2.2.3.

a) If Computer has Power Monitor Option perform paragraph 2-3.2.2.10 instead of these procedures.
b) Set Console switches to 000002_8.
c) Momentarily raise the Start/Continue switch to the Start position.
d) The program will request the operator to turn the computer off, on and to restart it.
e) Upon restart the program will ring the teletype bell three (3) times. The Operator should repeatedly perform the power off-restart sequence. After each restart allow 2-3 seconds for a possible error message.

2-3.2.2.10 Power Shutdown Test (with Power Monitor). Load the test program (Binary #095-000044) as per paragraph 2-3.2.2.2, or paragraph 2-3.2.2.3.

a) Set Console switches to 000002_8.
b) Momentarily raise the Start/Continue switch to the Start position.
c) The program will request the operator to turn the computer off, on, and to restart it.
d) After several restarts lock the Console and remove the AC line plug.
e) Return the AC line plug to its original connection. After power comes back up the program will restart without operator intervention. The teletype bell will be rung three (3) times each time power is restored.
f) The operator should repeatedly remove and restore power in this manner. At each restoration of power allow 2-3 seconds for possible error messages.

2-3.2.2.11 Exerciser. Load test program (Binary #095-000012) as per paragraph 2-3.2.2.2, or paragraph 2-3.2.2.3.

a) Set Console switches to 000002_8.
b) Momentarily raise the Console Start/Continue switch to the Start position.
c) Computer will cycle. Any Halt constitutes error. After one pass raise Console switches 2 and 3.
d) Turn on TTY punch. Set reader switch to FREE.
e) After TTY starts punching data, place leader in TTY read station.
f) Push START on TTY.
g) Allow computer to cycle for five minutes.
h) Lower Console switches 2 and 3 to terminate teletype test.

This test completes the start-up checkout for the Nova 800 Computer.

2-3.3 Repacking

In order to properly repack the Nova 800 or the Teletype, reverse the procedures listed in paragraph 2-2. Only strict adherence to the particulars described in each step will prevent serious damage to each machine during shipment. All retaining hardware and packing should be replaced into the original positions within the carton before the units are shipped. The following special packing considerations must be observed for the Teletype:

a) Make sure (3) hex head screws are replaced in the original position underneath the Shipping Pallet.
b) The Console front switches and keyboard must be protected with some form of resilient packing or extensive damage will occur during shipment.

SECTION III

OPERATION

3-1 GENERAL

This section identifies and describes all of the manual controls and indicators used to operate the Nova 800 Computer. A description of the controls and indicators is also provided in paragraph 2.7 of the "How to Use the Nova Computers" reference manual. Figure 3-1 is a drawing of the Nova 800 operator's Console showing the controls and indicators referenced throughout this section.

3-2 CONSOLE CONTROLS

Used in conjunction with a teletypewriter and peripheral devices, the control console contains all controls necessary to operate the Nova 800 Computer system. Each console control is described briefly in the following paragraphs.

3-2.1 Power Switch

The key-operated power switch controls the ac (primary power) input to the Nova 800 power supply. In the OFF position, the ac input line is removed from the power supply. In the ON position, ac power is applied to the power supply and the Computer is operational. However, in the LOCK position all Console Control switches are disabled except for the power switch itself. The LOCK position allows a program to run without interference from occasional or accidental "switch diddling" or any other unscheduled attempts to operate the Computer. However, the Console Data switches remain operational to allow the operator to supply information to the program (when requested by the program). It should also be noted that all of the Console indicators remain operational when the power switch is in the LOCK position. The Console key can only be removed when the power switch is in the LOCK position.

3-2.2 AC0, AC1, AC2 and AC3 Deposit/Examine Switches

These four switches are used both for depositing data into the corresponding Accumulator, and examining their contents. The DEPOSIT position of any switch operates in conjunction with the relative positions of the (16) Console Data switches. Placing any one of the four AC switches in the DEPOSIT position will load the configuration of the Console Data

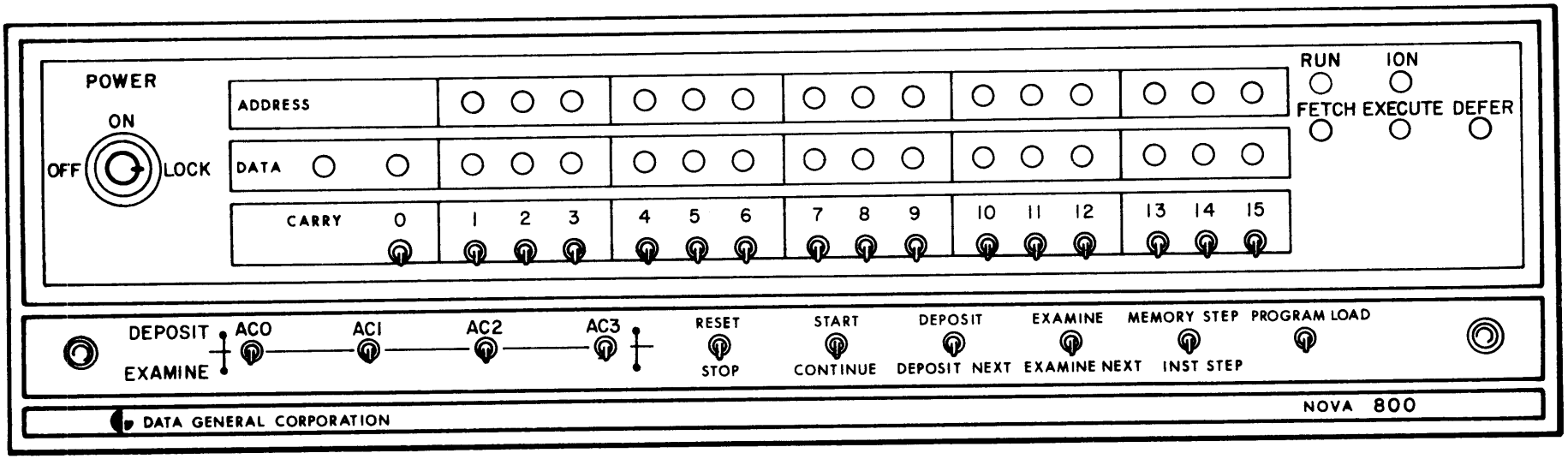


Figure 3-1. Nova 800 Operator's Console

switches into the specified Accumulator. Placing any one of the four AC switches in the EXAMINE position will display the contents of the specified Accumulator in the Console DATA lights.

3-2.3 Reset/Stop Switch

Placing the RESET/STOP switch in the RESET position causes the Nova 800 to stop at the end of the current processor cycle. RESET also: clears flags in all I/O devices, clears Interrupt On, places the processor in supervisor mode, and sets the clock to line frequency. It should be noted that if the RESET/STOP switch is momentarily raised immediately after an AC deposit the data will not be allowed to reach the selected AC even though this data appears in the Console Data display. Pressing the RESET/STOP switch to the STOP position causes the Nova 800 to stop before fetching the next instruction. The address indicators point to the next instruction. If the current instruction contains an infinitely long indirect addressing chain or there are continuous data channel requests, pressing STOP will not stop the computer. Under these conditions it is necessary to momentarily raise the switch to the RESET position rather than pressing it to the STOP position.

3-2.4 Start/Continue Switch

Momentarily raising the START/CONTINUE switch to the START position causes the Nova 800 to load the address contained in the Console Data switches into PC, light the FETCH and RUN indicators, and begin normal operation by executing the instruction at the location specified by the PC. Pressing the START/CONTINUE switch to the CONTINUE position causes the Nova 800 to turn on the RUN indicator and begin normal operation in the state indicated by the (five) indicators on the right-hand side of the Console (i.e., RUN, ION, FETCH, etc.) It should be noted that instruction stepping can be performed by momentarily raising the START/CONTINUE switch to the position while pressing the RESET/STOP switch to the STOP position.

3-2.5 Deposit/Deposit Next

Momentarily raising the switch to the DEPOSIT position will deposit the contents of the Console Data switches into the memory location specified by the address lights. Upon completion of the deposit the Console Data lights will display the word deposited. Pressing the switch to the DEPOSIT NEXT position will add 1 to the PC address displayed in the address lights and deposit the contents of the Console Data switches into the memory location specified by the incremented address. Upon completion of the deposit next the Console Data lights will display the word deposited. This switch is generally used in conjunction with the EXAMINE/EXAMINE NEXT switch. See paragraph below for an example switching sequence utilizing both switches.

3-2.6 Examine/Examine Next Switch

Momentarily raising the switch to the EXAMINE position will load the address contained in the Console Data switches into PC (which is displayed in the address lights) and display the contents of the addressed location in the Console Data lights. Pressing the switch to the EXAMINE NEXT position will add 1 to the PC address displayed in the address lights and display the contents of the location specified by the incremented address in the Console Data lights. The DEPOSIT/DEPOSIT NEXT, EXAMINE/EXAMINE NEXT switches can be used for a sequence of operations on consecutive memory locations. The sequence must begin with EXAMINE to supply the initial address unless PC already points to the right location. Suppose the Console Data switches are set to octal 100 initially. Then the following sequence of switch settings produces the effects listed.

EXAMINE	Display location 100.
EXAMINE NEXT	Display location 101.
EXAMINE NEXT	Display location 102.
DEPOSIT	Load Data switches into 102.
EXAMINE NEXT	Display location 103.
DEPOSIT	Load Data switches into 103.
DEPOSIT NEXT	Load Data switches into 104.
EXAMINE NEXT	Display location 105.

It should be noted that the EXAMINE position can be used to load the PC for beginning any single step procedure.

3-2.7 Memory Step/Inst Step Switch

Momentarily raising the switch to the MEMORY STEP position will perform a single processor cycle in the state indicated by the Operational Indicators and then stop. Upon completion the Operational Indicators will point to the next Operational state to be executed. The address lights will display the contents of the PC, the data lights will display the data fetched from the last memory location accessed. Actuating the AC switches between memory steps within an instruction usually destroys information (in the Accumulator) necessary for the execution of the rest of the instruction. To use the various examine and deposit switches between instruction steps, simply remember what PC is and restore it before continuing.

Pressing the switch to the INST STEP position will begin operation in the state indicated by the lights but then stop as though STOP had been pressed at the same time. If the stop occurs at the end of an instruction, the data displayed by the data lights depends on the instruction as follows.

LDA, STA	Operand
ISZ, DSZ	Operand
JMP	Direct→Instruction
JSR	Direct→Instruction
	Indirect→Effective Address
Arithmetic and logical	Instruction
In/out	Data

Note that the AC switches can be used between instruction steps without requiring any readjustment.

3-2.8 Program Load Switch

The PROGRAM LOAD option for the Nova 800 will deposit the contents of the bootstrap read-only memory into locations 0-37, light the RUN indicator and begin normal operation at location 0.

3-3 CONSOLE INDICATORS

The Console indicators are composed of two (register driven) indicators strings and five individual (flip-flop driven) function indicators. The two indicator strings are displays for the ADDRESS (or present contents of the PC), and the DATA content of a memory location or an Accumulator. The five function indicators indicate the operation

state of the Processor. A brief description of each display is presented in the following paragraphs.

3-3.1 Address Display

This section of the Console displays the present contents of the PC. When performing an EXAMINE, this display should be identical to the Address configuration set into the Console Data switches.

3-3.2 Data Display

This section of the Console can display either the contents of any one of the four Accumulators, or display the contents of a memory location. For example, during an EXAMINE, the selected (by the Console Data switches) address will be displayed by the ADDRESS indicators, and the contents of the selected memory location will appear in the DATA display.

3-3.3 Operational Indicators

When any indicator is lit the associated flip-flop is in the 1 state verifying that the associated function is true. A few indicators display useful information while the processor is running, but most change too frequently and are therefore discussed in terms of the information they display when the processor has stopped. Each functional indicator is listed below with its indicative interpretation.

RUN	The processor is in normal operation with one instruction following another. When the light goes off, the computer stops.
ION	The program interrupt is enabled (this is the Interrupt On flag).
FETCH	The next processor cycle will be used to fetch an instruction from memory.
DEFER	The next processor cycle will be used to fetch an address word in an indirectly addressed memory reference instruction.
EXECUTE	The next processor cycle will be used to reference memory for an operand in a move data or modify memory instruction.

FETCH, DEFER, and EXECUTE are the state indicators: they specify the state (the type of cycle) the processor will enter if operations are continued by pressing the CONTINUE or MEMORY STEP switch. At the most, only one indicator is lit at any one time. Unless otherwise indicated, use of any operating switch leaves the processor ready to enter the fetch state.

SECTION IV

THEORY OF OPERATION

4-1 INTRODUCTION

This section contains detailed information describing the functional relationships of the major logic sections comprising the Nova 800 Computer. This section is essentionally a continuation of the general functional description of the Nova 800 architecture provided in paragraph 1-2 of this manual. It is pointed out here that this text is intended to familiarize personnel with the functional operation of the major Processor logic sections, hence, the descriptions presented in this section are designed to provide basic conceptual information concerning the operation of the Nova 800. In this regard it is suggested that Figure 4-1, the Nova 800 Functional Block Diagram, be referenced along with the appropriate logic diagrams (bound Section VII of this manual under separate cover) while reviewing the descriptions of this section.

The Integrated Circuits of the Nova 800 are operationally synchronized with the CPU Clock (CLK) signal. The clock timing is arranged such that when the required enabling signals are present simultaneously with the negative-going edge of the CPU CLK signal, the corresponding logical operation will occur. Detailed information concerning the pin nomenclature and the signal requirements of the IC packages is provided in Appendix A of this manual. Signal origins for the Nova 800 are listed in Appendix B.

4-2 DETAILED FUNCTIONAL DESCRIPTION

The discussion presented herein will consider the operation of the major logic sections of the Nova 800. The drawing numbers of the logic diagrams of the major logic sections are listed below with their titles as a convenient reference.

NOVA 800 & 1200 CONSOLE	001-000092
CPU-1 NOVA 800	001-000092
Nova 800 Timing	(Sheet 1 of 4)
Nova 800 IR and States	(Sheet 2 of 4)
Nova 800 ALU Control	(Sheet 3 of 4)
Nova 800 IO	(Sheet 4 of 4)

This Page Left

Blank Intentionally

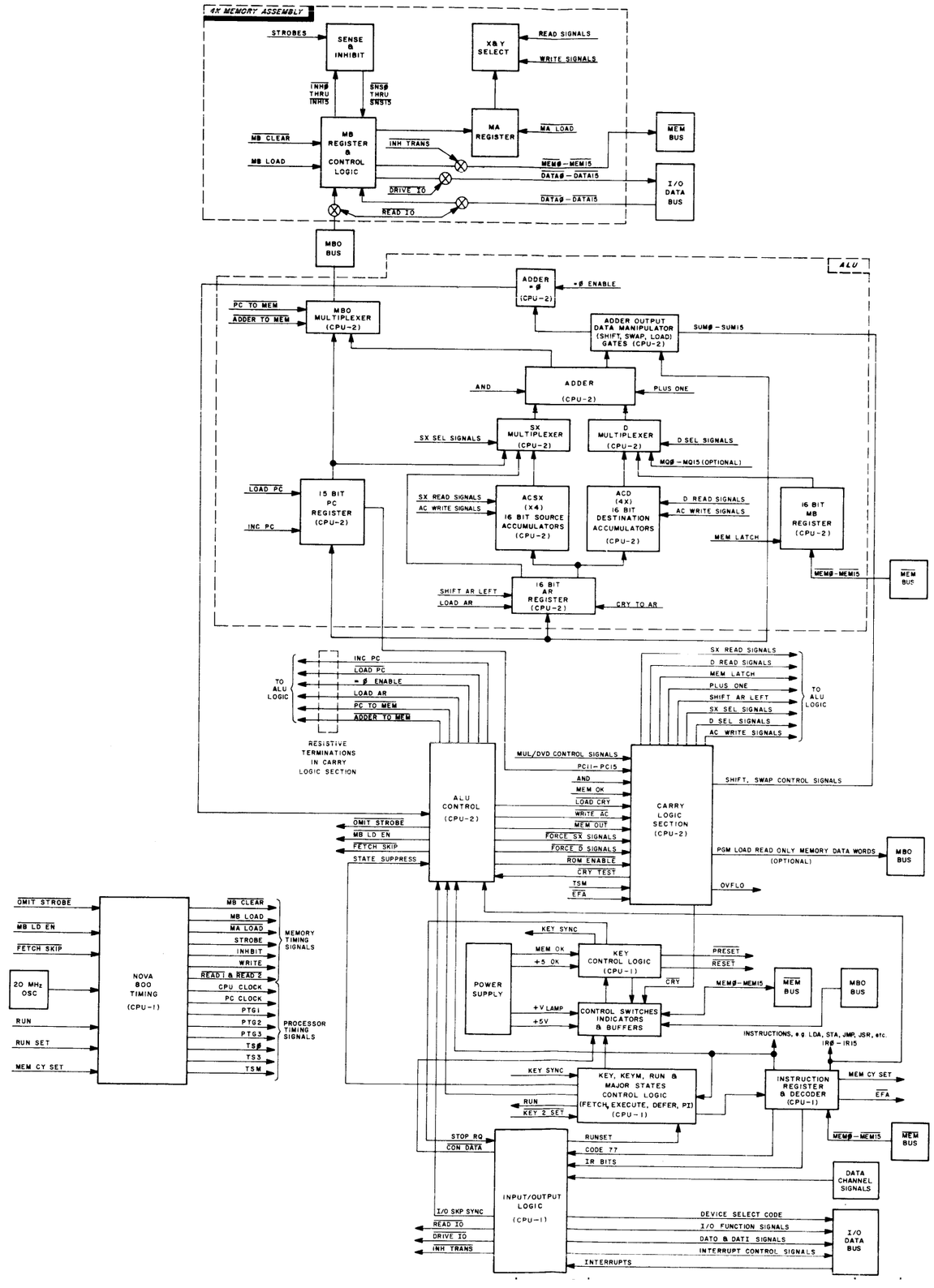


Figure 4-1. Nova 800 Functional Block Diagram

CPU-2 NOVA 800	001-000093
Nova 800 CRY	(Sheet 1 of 3)
Nova 800 ALU Bits 0-7	(Sheet 2 of 3)
Nova 800 ALU Bits 8-15	(Sheet 3 of 3)
4K MEMORY	001-000104
MA & MB Register & Control	(Sheet 1 of 4)
Sense & Inhibit	(Sheet 2 of 4)
X Drivers	(Sheet 3 of 4)
Y Drivers	(Sheet 4 of 4)

As mentioned previously Figure 4-1 is a functional block diagram of the major logic sections of the Nova 800. Certain common distribution functions such as logic and memory power lines from the power supply and the $\overline{\text{PRESET}}$, $\overline{\text{RESET}}$, and CPU CLOCK lines are not shown on the block diagram in the interest of clarity. The signals shown are included mainly to facilitate the following functional discussion and are not intended to represent the entire complement of internal logic signals.

4-2.1 Turn-On & Initialization Functions

Power is applied to the Nova 800 by turning the Console key to the ON position. As shown on the Console logic diagram, setting the key to the ON position switches 115 VAC into the Nova 800 power supply. After power has been turned on, the power supply produces the + and -5 volt logic power along with the + VMEM voltage for the memory. The power supply contains precision differential circuitry which monitors the + 5 volt and + VMEM voltage outputs. These circuits will produce a + 5 (volt) O.K. logic signal to indicate the + 5 output level is correct and a MEM O.K. logic signal to indicate the + VMEM voltage level is correct. The power supply also contains a Power Failure ($\overline{\text{PWR FAIL}}$) monitor circuit which is used in conjunction with the Power Monitor and Auto-restart option. The full-wave lamp voltage + V_{lamp} is also produced in the power supply. The + 5 O.K. and MEM O.K. lines carry power status signals from the power supply to CPU-1 and CPU-2.

When power is first turned on, the + 5 O.K. line provides a positive transition (to approximately + 5 volts) as the + 5 volt output rises to its proper level. This positive transition is gated into a differentiating capacitor to produce a pulse which in turn drives

the RESET and PRESET gates. $\overline{\text{RESET}}$ and $\overline{\text{PRESET}}$ initialize the control logic of the Nova. It will be noted (on the IR and States Logic diagram) that the + 5 O.K. line from the power supply is OR gated with the $\overline{\text{RST}}$ (Reset) line from the Console. $\overline{\text{RST}}$ becomes low when the Console Reset switch is actuated, the positive-going trailing edge (caused by releasing the switch) of $\overline{\text{RST}}$ is differentiated to generate the $\overline{\text{RESET}}$ and $\overline{\text{PRESET}}$ signals. The one-to-zero transition of the $\overline{\text{RST}}$ line (caused by actuating the Console Reset switch) enables logic which sequentially clears the RUN flip-flop. The zero-to-one switching transition occurring on the $\overline{\text{RST}}$ line is utilized to enable the $\overline{\text{PRESET}}$ and $\overline{\text{RESET}}$ initializing logic. It is noted that one of the functions of $\overline{\text{RESET}}$ is to drive the Master Reset (MR) input of a 4 Bit Discretes Register in the States logic, unconditionally clearing all of the outputs including RUN to the zero state. Hence, the main reason Reset ($\overline{\text{RST}}$) is bi-functional (i.e., clears RUN and then generates $\overline{\text{RESET}}$ and $\overline{\text{PRESET}}$ on the zero-to-one transition) is to allow the processor to stop synchronously with the completion of the last instruction, thereby allowing the program to be continued from the last instruction when the processor is started again. It should be noted that since there is contact bounce in the switch - the $\overline{\text{PRESET}}$ action will occur before the switch is released but after the synchronous stop. Hence, the Console Reset switch may also be actuated to initialize the Nova 800.

The $\overline{\text{RST}}$ line from the Console Reset switch is also OR gated with both the + 5 O.K. and MEM O.K. lines and the output of a gating tree monitoring the remaining Console control Keys. A low signal on any input to the OR gate generates a STOP RQ signal, which stops the Computer synchronously at the end of the current instruction. The main difference between these functions is that $\overline{\text{RST}}$ or + 5 O.K. becoming low clears the machine immediately, whereas actuating the Stop Key at the Console will stop the machine at the end of the present instruction. The STOP RQ signal at the logic 1 or high level enables another OR gate (the other input is a HALT instruction decode) the output of which places a logic 0 reset signal on the input to a RUN SET flip-flop stage of a 4 Bit Discretes register. The output from the RUN SET flip-flop stage controls (through additional gating) the input to the RUN flip-flop stage of another 4 Bit Discretes register, thereby resetting the RUN flip-flop in sequence. (RUN SET is reset at PTG2 time and RUN is reset as a result at PTG3 time.)

It should be noted that the Nova 800 CPU logic utilizes approximately 5 of these 4 Bit Discretes Registers to store discrete information defining the operational state of the processor. In each case each flip-flop stage can be set or reset independent of the state of any other flip-flop stage in the register. Further, conditions for either setting or resetting any stage is set up by decision gates at the stage input prior to the coincident arrival of a "load strobe" pulse together with the trailing edge of the register clock pulse. At this time the output logic levels presented by the gates driving the input to each stage are latched into the register, with relatching occurring on each load and clock pulse combination.) It is noted further that some of the 4 Bit Discrete Registers are also connected to perform right shift functions. Under these conditions a logical bit loaded into the DS input of the register will be shifted on each clock with the Shift input enabled.

Therefore, with RUN SET reset the next coincident occurrence of the PTG3 and the trailing edge of the CPU CLK pulse, a zero will be loaded into the RUN stage of the register. RUN SET is routed to the Timing logic, where in the reset state it inhibits the start of the next memory cycle. RUN is also used in the Timing logic, where in the reset state it prevents the PTG (Processor Timing) signals from being generated. Thus, with RUN SET and RUN reset the Processor will enter the stop state.

Hence, to return to the initialization discussion, as power comes on in the machine signals $\overline{\text{RESET}}$ and $\overline{\text{PRESET}}$ initialize the logic of the entire machine, leaving the machine in the Stop state. The next operation in the sequence must be performed from the Console. The Console switch and indicator functions are described briefly in the following.

4-2.2 Console Display

The Console displays two sets of data, address and memory data, along with data concerning the state of the machine. The address lights are driven by the MBO Bus from CPU-2, and display the next address to be referenced when the machine is stopped. While the machine is running the data lights display the contents of the MB register in the Memory selected at the time. The lights are driven directly off the Memory bus. The states of the Carry, Run, Ion, Fetch, Defer, and Execute flip-flops are also displayed.

It will be noted on Figure 4-1 that the $\overline{\text{CRY}}$ signal from the Carry Logic section is shown as a direct line to the Console Switches Indicators & Buffers block. This was done to simplify the drawing. $\overline{\text{CRY}}$ actually is bussed through the States logic section, and then out to the Console Indicators as a $\overline{\text{CARRY}}$ signal. All signals displayed on the Console are asserted negative. A non-inverting current driver is used to drive the lamps. A parallel resistance is used to provide a continuous flow of about ten milliamperes through the lamp which keeps the filament hot and eliminates large surge currents when the lamp is turned on.

4-2.3 Console Data Switches

The Data switches are tied directly to the Memory bus. The non-inverting buffers have open-collector (OC) type outputs and are normally in the off state. During console operations the READS (DIA-, CPU) instruction causes the $\overline{\text{CON DATA}}$ line to be switched low, thereby causing all buffer outputs to go low. Those switches which are closed, designating a 'one', will put low levels on the bus. At this time all memories are disconnected from the bus by $\overline{\text{INH TRANS}}$ from the Input/Output Logic of CPU-1.

4-2.4 Console Control Switches

All pull-up resistors on the control switches, with the exceptions of STOP and RESET, are connected to a common node. This node is connected to the base of Q1. (See the Nova 800 & 1200 Console Drawing #001-000089.) The circuit formed by Q1 and related components performs two functions: 1) it senses current flowing through any of the pull-up resistors, and 2) it provides a delay of about 25 milliseconds from the time the switch is first actuated to the time $\overline{\text{CON RQ}}$ is asserted low. This delay guarantees that all switch bounce has subsided before the Key Control logic of CPU-1 attempts to perform the function requested. $\overline{\text{CON RQ}}$ must switch cleanly for proper operation. The Key Control logic, upon receiving the $\overline{\text{CON RQ}}$ signal drops the $\overline{\text{CON INST}}$ line. This allows the control switches to be connected to the Memory bus through OC (Open Collector) - gates U1 and U2. Several switches may be ORed into one

gate. For example, if switch ACDP 2 is actuated $\overline{\text{MEM 0}}$, $\overline{\text{MEM 1}}$, $\overline{\text{MEM 4}}$, $\overline{\text{MEM 5}}$ would go high. All other bits driven by U1 and U2 would go low. Bits 8 through 15 would also be high, but they are not used to encode the Console functions. Reset, Stop, Memory Step, and Instruction Step are not encoded into the Memory bits, but rather unique lines are generated which define these functions.

4-2.5 Examine and Deposit

The next operational step after the Nova 800 has been initialized may be either to toggle in a small program at the Console, or to command an automatic program load at the Console. To simplify the discussion at this point it will be assumed that a single instruction loop is to be toggled in (e.g., deposit zero in location 000000 vis-a-vis jump •LOC 0). This is accomplished by clearing all of the Console Data switches to the (down) zero position, lifting the Examine switch, then lifting the Deposit switch. At this point it is worthwhile to develop a technique for reading the Flow Chart. The Flow Chart (reference Drawing 001-000121, sheet 1) functions as a general, undetailed map of the sequence of major machine operations, and serves as an excellent preface prior to reviewing the actual engineering logic diagrams for the signal sequence for any specific operation. The flow chart may be considered to be a matrix; with the three categories of processor logic states as variables. The timing states are shown at each side of the chart; the machine cycles are listed across the top; and the instructions are branches in the flow paths.

To determine from the flow chart the sequence of operations for any instruction, start at the top (PTG0) of the machine cycle FETCH for any programmed instruction, or the KEY cycle for any console initiated instruction; and follow the path downward on the page. When any junction is reached, the instruction determines which branch to follow. At the end of each machine cycle, conditions listed will tell which cycle is next. Proceed to the top of the next cycle; and continue following the chart until reaching F-COM, which is the end of an instruction. Any blocks reached in following the flow chart indicate logical operations performed by the machine. From the position of a block on the matrix the instruction, machine cycle and timing state of the operation can be determined. These factors indicate that specific

hardware gates exist in the processor to produce whatever control signals are required by the hardware to perform the function shown in the block.

Since the first operation to be considered is an Examine followed by a Deposit the initial flow will be down the KEY flow, first for the Examine and then for the Deposit. (It should be noted that any sequence of Deposits must begin with an Examine to supply the initial address unless the PC is already pointing to the deposit location.) As shown on Drawing 001-000121, sheet 1, the Key flow is entered whenever a Console Key is actuated. In this case (Examine Key) a CONT + ISTP + MSTP signal will not be present, therefore the $\overline{\text{CONT + ISTP + MSTP}}$ path is correct. Continuing down the path that is identified as DP, DPN, EX, EXN, PL (since this is an Examine (EX) Key flow), the operation CON→PC is performed. This operation transfers the configuration set into the Console data switches into the PC. The contents of the PC are then copied into the MA register (of Memory) and the flow then switches to the KEYM (Memory) flow. The KEYM flow for Examine essentially does nothing but wait for the Memory Cycle to be completed and then stop. Since the PC was loaded into the MA this particular Memory Cycle will retrieve data from the location addressed by the Console Data switches, place it into the (Memory) MB register, and gate it out from the MB to the Console Indicators via the $\overline{\text{MEM}}$ Bus. This is the flow of operations for an Examine Key cycle as defined by the Flow Chart on Drawing 001-000121. The next portion of the discussion retraces the same operational steps but this time in reference to the Functional Block diagram and the applicable engineering logic diagrams. It should be noted that general information not directly relative to the operation flow being discussed are bounded in the text by a pair of square brackets ([ ]).

Actuating the Examine and Examine Next Key to the Examine position generates a $\overline{\text{CON RQ}}$ signal. $\overline{\text{CON RQ}}$ sets the Key Seen flip-flop (with the RUN flip-flop reset), which enables the Key Sync flip-flop to become set on the next $\overline{\text{PC CLK}}$. The logic 1 KEY SYNC signal enables the Shift and DS inputs of the 4 Bit Discretes register containing the KEY, RUN, DCH, and KEYM flip-flops. The next CPU CLK to occur will shift a logic 1 into the Key flip-flop stage. The next CPU CLK will right shift the logic 1 into the RUN flip-flop while shifting another logic 1 into the Key flip-flop. When the Run flip-flop sets it resets the Key Seen flip-flop, which in turn causes the Key Sync flip-flop to reset on the next $\overline{\text{PC CLK}}$ signal. Setting the Run flip-flop also removes the inhibit from the Timing section so that

PTG signals may be generated. At some point after the KEY actuation and prior to the Key flip-flop becoming set, the $\overline{\text{CON INST}}$ signal becomes true and gates the states of the Console control switches out to the $\overline{\text{MEM}}$ Bus lines $\overline{\text{MEM0}}$ thru $\overline{\text{MEM7}}$. This information is loaded into stages IR0 thru IR7 of the Instruction register. When the Key and Run flip-flops become set a $\overline{\text{CON DATA}}$ signal is sent from the I/O logic section of CPU-1 back to the Console. $\overline{\text{CON DATA}}$ enables buffers driving the Console Data switches, thereby allowing the electrical configuration of the Data switches to be placed on the $\overline{\text{MEM}}$ Bus lines. Signal TSM (Time State Middle or PTG1 or PTG2 time) produces a MEM LATCH signal in the Carry logic section of CPU-2 which latches the data on the $\overline{\text{MEM}}$ Bus into the MB register of the ALU. TSM is produced automatically as long as the Computer is running, and hence provides access into the ALU (at this time) for data present on the $\overline{\text{MEM}}$ Bus. The Address data flow is a transfer from the MB through the D Multiplexer, through the Adder, out through the Adder Output Data Manipulator gates to the SUM0 thru SUM15 lines, and finally around to the input of the PC register. It is important at this point to consider the operation of both the Source and Destination Multiplexers, even though the Source Multiplexer is not used in the Examine operation.

Data flow into the Adder is supplied through two multiplexers called SX-MULT and D-MULT. These multiplexers determine what data will be supplied to the Adder by selecting from three sets of data inputs, each set coming from a data register.

The input data to be selected by the multiplexer is determined by a two bit coded select input: for SX-MULT, the signals SX-H-SEL and SX-L-SEL; for D-MULT, the signals D-H-SEL and S-L-SEL. It should be noted that the D-H-SEL signal is actually identified as either D-M-H-SEL as the high order select line for ALU Bits 0 thru 7, and D-L-H-SEL as the high order select line for ALU Bits 8 thru 15. The letter M in the code stands for "Most Significant", whereas the letter L stands for "Least Significant." However, the M and L designations have been dropped from codes shown below as the codes are applicable to both sections (M & L) of the ALU.

These signals are decoded to select the registers as shown below:

SX-H-SEL	SX-L-SEL	OUTPUT
0	0	0
0	1	AR
1	0	PC
1	1	ACSX

D-H-SEL	D-L-SEL	OUT
0	0	0
0	1	MQ*
1	0	MB
1	1	ACD

*MUL/DIV Option

There is a complement signal to each multiplexer (except for D-MULT ALU Bits 8-15); (SX-COM, D-M-COM) which, when true, causes the output data to be inverted. The Destination Multiplexer code to select the MB is produced by a $\overline{\text{MEM OUT}}$ signal in conjunction with the $\overline{\text{FORCE D-L-SEL}}$ signal being absent. Under these conditions the inputs to the Adder from the D Multiplexer will consist of the original data from the Console Data switches.

At this time (due to the STATE SUPPRESS* signal being present) no SX-SEL signals will be produced, resulting in a 0 output from the SX Multiplexer being applied to the other set of Adder inputs. [The operational ground rules for the Adder is described in the following discussion. The Adder performs binary addition on pairs of inputs, and produces carry out if the sum exceeds 2^{16}. The Adder sum can be incremented (the signal "plus one") by forcing a carry into the least significant bit. When the signal AND is true, the Adder performs the logical function AND on each pair of inputs.]

Therefore, the Examine address (originated at the Console Data switches) is added to zero (from the Source Multiplexer) and applied to the Adder Output Gates. Since an ALC IR code is not present (as this is an Examine operation) both $\overline{\text{FORCE SELX}}$ and $\overline{\text{FORCE SELY}}$ (shift and swap control) signals will not be present, thereby allowing the Carry logic section

*The STATE SUPPRESS function, generated whenever the processor is in a Key, KEYM, or DCH Cycle, is used to inhibit the operation of the Major States logic.

to generate an SEL-N (Select No Shift) signal for the Adder Output gates. This places the Adder output data onto the SUM0 thru SUM15 lines exactly as it appears at the outputs of the Adder. [To depart from the discussion for a moment it is noted here that the three other shift and swap signals to the Adder Output Data Manipulator Gates have the following command interpretations: SEL-L = Select Left Shift, SEL-R = Select Right Shift, and SEL-S = Select Swap. The Data Manipulator gates have the function (depending on which gate in the section is enabled) of shifting output data from the Adder either left or right, or the two bytes of the Adder Output word may be swapped. These are done on a bit by bit basis. For example, if the data is to be shifted left, the output of the section for each bit is the value of the Adder data bit to its right: bit 6 takes the value of bit 5, bit 0 takes the value of carry, etc. If the bytes are swapped, the carry bit remains unchanged.]

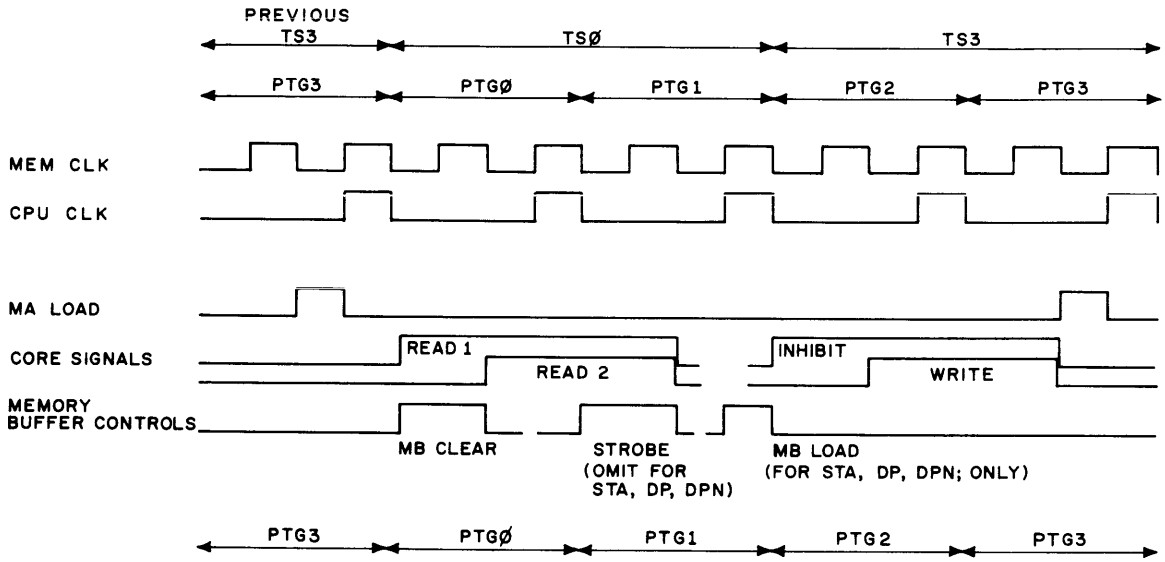
To return to the discussion the next signal to consider in the Examine sequence is a $\overline{\text{LOAD PC}}$ signal from the ALU Control section. A low logic level on this line loads the data present on the SUM0 thru SUM15 lines into the PC register. [At this point it is worthwhile to consider the general operational ground rules for the PC. The PC can be loaded during jump instructions (by the signal "LOAD PC") or incremented twice during an instruction which skips. Data from the PC may go directly to the Memory for addressing, or to the Adder (through SX-MULT) for address calculation. Data into the PC always comes from the Adder. Operations in the PC are effected by the leading edge of the clock signal PC CLK. That is, if the signal INC PC is true, the binary number stored in the register will be increased by one when the positive going edge of PC CLK occurs.]

To return to the discussion of the Examine sequence, the $\overline{\text{PC TO MEM}}$ signal from the ALU Control causes the contents of the PC register to be gated through the MBO Multiplexer out onto the MBO Bus. At this time the $\overline{\text{READ IO}}$ signal will be at a logic 1 level and thereby gate the data on the MBO Bus onto the inputs of the Memory MB register. This data is gated into the Memory MB register by a MB LOAD signal from the Timing section of CPU-1 (at PTG1 time). It should be noted that the data loaded into the Memory MB register at this time is of no consequence to the Examine Operation, as it is subsequently cleared at the start of the upcoming Memory cycle. MB loading at this time may be considered an automatic function which facilitates certain operations.

The last transfer of the Key Examine cycle is performed by the $\overline{\text{MA LOAD}}$ signal which is produced in the Timing section of CPU-1 at PTG3 time (with both RUN SET and MEM CY SET present as logic 1's). The $\overline{\text{MA LOAD}}$ signal loads the data on the MBO Bus lines $\overline{\text{MBO1}}$ thru $\overline{\text{MBO15}}$ into the MA register via Memory lines MD1 thru MD15. At this time all is ready for a Memory cycle and the trailing edge of PTG3 loads a logic 1 into the KEYM stage of the 4-Bit Discretes register of the Key Control logic (Drawing 001-000092, sheet 2).

As mentioned previously a Memory cycle will be performed starting on the next PTG0. Figure 4-2 is a timing diagram of the basic Memory cycle. The STROBE signal enables the Memory sense amplifiers, the outputs of which are unconditionally loaded into the Memory MB register. At this time the $\overline{\text{INH TRANS}}$ signal will be in the logic 1 state, allowing the MB register outputs to be gated out via the $\overline{\text{MEM}}$ Bus to the Console indicators. ($\overline{\text{INH TRANS}}$ is in the logic 0 state during the Key Cycle to isolate the $\overline{\text{MEM}}$ Bus from Memory, and allow the Console data to be transferred over the Bus without interference. Once the Key flip-flop becomes reset $\overline{\text{INH TRANS}}$ is switched to the logic 1 state. $\overline{\text{INH TRANS}}$ is switched low initially by KEY SEEN + RESTART, and then held low by RUN and KEY.) Note that on the timing diagram a STROBE is not produced for a STA instruction, or a Deposit or Deposit Next Console operation. However, an MB LOAD is produced for STA Deposit, or Deposit Next operations. The set output from the KEYM flip-flop is used to produce a STOP RQ signal during the Examine KEYM cycle. The logic 1 STOP RQ signal enables a gate (in the I/O logic section) which in turn places a logic 0 on the input to the Run Set flip-flop stage of a 4 Bit Discretes register. Thus, at PTG2 time (of the KEYM cycle) the Run Set flip-flop will be reset. The Run flip-flop, chained into the sequence, will be reset at PTG3 time due to the reset state of the Run Set flip-flop. As described previously resetting the Run flip-flop inhibits the PTG logic in the Timing logic section thereby stopping the Computer.

4-2.5.1 Deposit. The Deposit Key operation is similar to and follows the Examine Key operation. Referencing the Flow Chart (Drawing 001-000121, sheet 1), the Deposit operation enters the Key flow path when the Console Deposit Key is actuated. In this case (Deposit Key) a CONT + ISTP + MSTP signal will not be present, therefore the $\overline{\text{CONT + ISTP + MSTP}}$ path is correct. Continuing down the path that is identified as DP, DPN, EX, EXN, PL



BASIC MEMORY CYCLE
(FETCH; PI; DEFER IF $\overline{\text{AUTX}}$; EXECUTE OF LDA, STA; KEYM)

Figure 4 -2. Timing Diagram of a Basic Memory Cycle

(since this is a Deposit (DP) Key flow), the operations CON→AR and PC→MA are performed. These operations transfer the configuration set into the Console Data switches into the AR register of the ALU, and transfer the contents of the PC into the MA register (of Memory) respectively. The flow is then switched to the KEYM path which (for DP and DPN) transfers the AR into the Memory MB during the KEYM Memory Cycle and then stops the Computer at the end of the KEYM Cycle. It is noted that the PC contains the proper core address for the deposit, a residual condition from the previous Examine operation. The next portion of the discussion retraces the same operational steps but this time in reference to the Functional Block diagram and the applicable engineering logic diagrams.

Actuating the Deposit and Deposit Next Key to the Deposit position generates a $\overline{\text{CON RQ}}$ signal. $\overline{\text{CON RQ}}$ set the Key Seen flip-flop (with the RUN flip-flop reset), which in turn enables the Key Sync flip-flop to become set on the next $\overline{\text{PC CLK}}$. The reset output from the Key Seen flip-flop enables a gate to produce a logic 1 KEY SEEN + RESTART signal. This signal, as described previously, produces (from the logic 1 state) an $\overline{\text{INH TRANS}}$ while the Run flip-flop is reset. The RUN and KEY signals subsequently hold $\overline{\text{INH TRANS}}$ low for the remainder of the Key cycle. All of the logical operations for the Examine Key cycle, up to the point where the data is placed on the SUM0 thru SUM15 lines, are the same (for this Deposit Key cycle) as previously described. The exception here is that a LOAD AR signal is produced instead of a $\overline{\text{LOAD PC}}$ signal, and loads the data on SUM0 thru SUM15 into the AR register of the ALU. The AR is used to store data from the Adder which is to be used in further arithmetic or written into an accumulator. It is written into on the trailing (negative going) edge of CPU CLK when the signal LOAD AR is true. [It is also used as a shift register by the MUL/DIV option. The AR is also vital to DCH operations as it holds the last generated address.] The LOAD AR signal is generated at PTG 3 time and loads the SUM data into the AR register.

The $\overline{\text{PC TO MEM}}$ signal from the ALU Control allows the contents of the PC register to be gated through the MBO Multiplexer out onto the MBO Bus. The last transfer of the Key Deposit cycle is performed by the $\overline{\text{MA LOAD}}$ signal which is produced in the Timing section of CPU-1 at PTG 3 time (with both RUN SET and MEM CY SET present as logic 1's). The $\overline{\text{MA LOAD}}$ signal loads the data on the MBO Bus lines $\overline{\text{MBO1}}$ thru $\overline{\text{MBO15}}$ into the MA register

via Memory lines MD1 thru MD15. The trailing edge of PTG3 loads a logic 1 into the KEYM stage of the 4-Bit Discretes register of the Key Control logic (Drawing 001-000092, sheet 2).

The contents of the AR are routed to the SX Multiplexer for passage through the Adder. Several signals produced in the ALU Control logic implement this data flow. As was previously listed, the SX-H-SEL line must be a logic 0 with the SX-L-SEL line a logic 1 to enable the SX Multiplexer to select the AR register outputs for inputs to the Adder. The FORCE SX-L-SEL line is brought to the low state by the logic 1 IR 6 signal from the IR register. None of the $\overline{\text{FORCE SX-H-SEL}}$ enabling signals are present at this time, thus this line remains high. Both signal states are inverted in the Carry logic section to apply a final high SX-L-SEL and low SX-H-SEL signals to the SX Multiplexer. Both input lines to the D Multiplexer are at the 0 level causing the D Multiplexer to place 0's on its inputs to the Adder. Therefore, the contents of the AR register are passed through the SX Multiplexer and added to zero. The second signal produced in the ALU Control logic as an implementor is the $\overline{\text{ADDER TO MEM}}$. This signal is generated prior to PTG3 time and transfers the Adder outputs through the MBO Multiplexer out to the MBO Bus. All of the foregoing operations occurred between PTG3 of the previous Key cycle and PTG2 of the KEYM Cycle. At PTG1 time of the KEYM cycle a MB LOAD signal is produced in the Timing logic section, and loads the (Adder output) data into the Memory MB register (since $\overline{\text{READ IO}}$ is high for this operation). At PTG2 time of this cycle the data now present in the Memory MB register will be written into core. To regress slightly, no Memory STROBE is produced at PTG0 time due to the inhibit placed on the input gate of the Strobe flip-flop by the $\overline{\text{OMIT STROBE}}$ signal. $\overline{\text{OMIT STROBE}}$ is produced in the ALU Control logic and becomes a logic 0 by virtue of Bit 2 of the IR register in the reset state with the KEYM logic 1 signal present. Bit 2 of the IR register was originally forced to the reset state by the logic 0 on the $\overline{\text{MEM2}}$ line from the Console. ($\overline{\text{MEM2}}$ is low for actuations of either the Deposit or Deposit Next switch.) Thus, the KEYM cycle for a Deposit omits the Memory Strobe and substitutes a MB LOAD signal later on in the Memory cycle. At the conclusion of the Memory cycle the Computer is stopped (through the facilities of the STOP RQ logic) in the same manner as described previously for the Examine operation.

4-2.6 Fetch

The discussion concerning the logical operations constituting a Fetch cycle is in the same format as the previous discussions i.e., the first part of the discussion will review the Fetch path as shown on the Nova 800 Flow Chart. The second part of the discussion will describe the same logical operations for the Fetch, but relative to the Functional Block diagram and the engineering logic diagrams. A Console Key (or switch) must be actuated in order to set the Nova 800 logic to the RUN state. Specifically, the starting Memory Address of the program to be run is loaded into the Console Data Switches, and then the Start switch is lifted. The processor will perform a Start (STRT) Key cycle in the same manner as that described for the Examine operation. The STRT path indicates that the CON→PC (Console Data switch information is transferred to the PC) operation is performed at PTG2 time of the Key cycle. At PTG3 time the PC→MA (PC transfers to the Memory MA register via the MBO Bus) occurs. Note that this Key flow is exactly like the Examine flow except that the STRT flow terminates in Fetch being set, whereas the Examine flow terminates in KEYM being set. The logical distinction concerning this difference is described later on in the second half of this discussion.

Upon entering PTG0 of the Fetch state a Memory cycle is automatically begun. The Memory data accessed during the Read and Strobe portion of the cycle is written into the Memory MB, and then transferred via the $\overline{\text{MEM}}$ Bus to the IR. The trailing edge of PTG1 latches the $\overline{\text{MEM}}$ Bus data into the IR. This data is decoded immediately and depending on the particular decode the flow will either increment the PC and load the updated PC back into the PC and the AR (denoted PC + 1→ PC, AR), or increment the PC and load the updated PC into the AR only (denoted PC + 1 →AR). If the instruction decode, for example, is an LDA, the logic will increment the PC and load the updated data into the AR to perform the effective address (EFA) calculations. The EFA is the actual memory address (relative to the PC) which is to be accessed for reading or writing of memory data. The calculated EFA is then (assuming the LDA flow path) loaded into the Memory MA register. The IR5 and $\overline{\text{IR5}}$ notations on the flow chart represent alternate switching for the flow path dependent on the state of IR register Bit 5. Bit 5 is defined as the indirect addressing bit and if it is set

automatically commands that the flow switch to the Defer State to access the indirect reference location. If IR Bit 5 is not set the flow will switch to the Execute cycle (for LDA, STA, ISZ, and DSZ instructions). Undeferred JSR and JMP instructions return to Fetch through the common operations shown under F-COM. ALC and I/O instructions are not deferrable and ALC's return to Fetch through F-COM (unless a program interrupt is present). I/O instruction Fetch flow is terminated by a switch to the Execute path.

The Execute path for the LDA instruction (to continue the example) consists of another Memory cycle during which the PC is incremented. At PTG3 time the Memory MB register data is transferred via the $\overline{\text{MEM}}$ Bus (and the MB register in CPU-2) through the Adder into the AR register. The flow then enters F-COM, the first operation of which transfers the contents of the AR register into the selected destination accumulator. If there are no program interrupts present, F-COM will transfer the updated PC (PC + 1) into the Memory MA register and switches the flow back to the Fetch path for retrieving the next instruction.

4-2.6.1 Fetch Logic Functions. This discussion will continue to use the LDA instruction as an example of the operations performed during the Fetch cycle. Obviously, there are many possible instructions that could be retrieved during any Fetch cycle. Even within the classification of the LDA instruction chosen as an example for this description, it is possible to nest several indirect references, or utilize the autoindexing locations in conjunction with the instruction. Therefore, the description presented here of the LDA Fetch and Execution are only intended to familiarize one with the correlated use of the Flow Chart and the Functional Block Diagram with the engineering logic diagrams. A brief description of the various instruction data paths is provided after this discussion.

As mentioned previously, actuating the Start Key will perform the same logical operations described earlier under the Examine discussion. However, at the conclusion of the STRT Key cycle KEYM does not become set, rather the Fetch Major state is entered. The logical conditions supporting this state is described in the following discussion.

In the STRT Key cycle, and due to the Console $\overline{\text{MEM}}$ code for the Start switch, $\overline{\text{IR6}}$ is in the logic 0 state. This level prevents (through gating) the KEYM stage of the 4 Bit Discretes register from becoming set at PTG3 time. One 4 Bit Discretes register (Ref. Drawing 001-000092, sheet 2) functions both as the Major States register and the auto-increment/autodecrement register. The last two stages of the 4 Bit Discretes register are set to binary configurations 0 0 through 1 1 which are decoded to enable one of the four following gates, $\overline{\text{FETCH}}$, $\overline{\text{PI}}$, $\overline{\text{DEFER}}$, or $\overline{\text{EXECUTE}}$. These two stages are generally cleared at the start of an operation, which is decoded as the FETCH state. Initially for any Key actuation except $\overline{\text{CON} + \text{ISTP} + \text{MSTP}}$, the $\overline{\text{PRESET}}$ signal is generated. This signal clears out the entire 4 Bit Discretes register, resulting in an output decode defining the FETCH machine state. The output decodes for the 3rd and 4th stages of this register are as follows:

3rd	4th	Machine State
0	0	FETCH
0	1	PI
1	0	DEFER
1	1	EXECUTE

It should be noted that in either the KEY, DCH, or KEYM state, a STATE SUPPRESS signal is produced which inhibits both the loading mechanism of the autoincrement/autodecrement - Major States 4 Bit Discretes register, and the (register) output decoder gates. Therefore, this logic is rendered inoperative during the KEY, KEYM, or DCH cycles. During STATE SUPPRESS all nominal instruction decoding/control functions are inhibited. However, at the end of the previous state, the Major State logic was loaded. Thus, the circuitry involving the "next" state really involves the SUPPRESSED STATE. These are principally MA LOAD functions. The STATE SUPPRESS signal also controls the 2-input multiplexer associated with the Major States outputs. This multiplexer is switched by STATE SUPPRESS such that the multiplexer outputs indicate (after the machine has been stopped) that the next state to be entered (when the machine is started) is a Program Interrupt (PI SET) or a Fetch (F-SET). The multiplexer inputs, with STATE SUPPRESS true, are supplied by the 3rd and 4th stage outputs from the 4 Bit Discretes register. On the other hand the multiplexer inputs, with STATE SUPPRESS false, are supplied by the control gates driving the inputs of the 3rd and 4th stages of the 4 Bit Discretes register.

With a logic 1 F-SET signal present a MEM CY SET signal is produced which allows the CPU CLK (at PTG3 of the Key cycle) to set the READ CY flip-flop (Timing logic of CPU-1) starting the Memory read-write cycle of the Fetch.

During Fetch and other Major States many operations occur simultaneously in the Nova 800 logic. One such instance are the operations which Increment the (current) PC and Loads this updated value into the AR register. These particular operations occur concurrently with the Memory Strobe during PTG1. These functions are automatic (INC PC and LOAD AR) and occur during each PTG1 of a Fetch cycle. At this time the SX Multiplexer is selecting the PC (even though the instruction data has not been Strobed out of Memory yet), the D Multiplexer is disabled (to allow the Contents of the PC to be added to zero), the Adder Output Data Manipulator gates are set to SEL-N (Select No Shift), and finally the logic 1 LOAD AR signal is present. This provides a path, during PTG1, from the PC through the SX Multiplexer and Adder, and out via the SUM Bus to the AR register. (The SX Multiplexer selects the PC whenever the $\overline{\text{FORCE SX-H-SEL}}$ and $\overline{\text{FORCE SX-L-SEL}}$ lines from the ALU Control equal a logic 0 and 1 respectively. This condition is enabled in the ALU Control by a logic 1 F signal and a logic 1 TS0 signal indicating TS0 of a Fetch Cycle.) Since the INC PC signal also occurs at this time the AR will be loaded with the updated PC.

The trailing edge of PTG1 also performs two important simultaneous functions relative to the Fetch sequence. It loads the IR with the data present on the $\overline{\text{MEM}}$ Bus as one of the functions. It also causes clock signal TSM to switch low, thereby switching MEM LATCH low, which latches the data on the $\overline{\text{MEM}}$ Bus into the Processor MB register.

A point to note here is that in the event a JSR instruction was decoded during the Fetch the next CPU CLK signal would generate the AC WRITE signals for the AC3. This operation copies the contents of the AR register (which during a Fetch is the updated PC) into both the Source and Destination AC3 accumulators. This principle also holds true for any ALC instruction. The duplicative use of both Source and Destination AC3's is simply an expeditious means of mechanizing the handling of data flow in the ALU. However, only the Destination AC3 is read during the return from the subroutine. [The ground rules for accumulator selection are described in the following discussion. There are two sets of four 16 bit wide accumulators. (ACSX, ACD), each set consists of 4 chips with 4 bits of each of the 4 accumulators in each

chip. Both sets always contain the same data.

Accumulator selection is made by two bit codes for writing and for reading. The accumulator sets are written into simultaneously, but they may be read separately.

Each pair of selection codes consists of an "L" signal and an "H" signal. The coding is:

H	L	AC SELECTED
0	0	0
0	1	1
1	0	2
1	1	3

Data from the AR is written into whichever accumulator is selected by the write inputs on the trailing edge of the signal $\overline{\text{AC WRITE}}$. The data contained in the accumulator currently selected by the read selection inputs is always available at the outputs.]

As stated previously, the Memory Strobe is also generated in the Timing logic at PTG1 time. This signal is conditional relative to the state of the $\overline{\text{OMIT STROBE}}$ line from the ALU Control logic. Since none of the $\overline{\text{OMIT STROBE}}$ inhibit conditions are present during the Fetch, $\overline{\text{OMIT STROBE}}$ will be at a logic 1 level, thereby allowing a logic 1 STROBE signal to be sent to Memory. The Strobe initiated Memory data (in this case the LDA instruction) is then forced into the Memory MB, which in turn places the data onto the $\overline{\text{MEM}}$ Bus. (As mentioned previously the trailing edge of the PTG1 CPU CLK latches the data on the $\overline{\text{MEM}}$ Bus into the IR register. This same $\overline{\text{MEM}}$ Bus data is also latched into the Processor MB register by a MEM LATCH logic 1 signal. It should be recalled that MEM LATCH is produced automatically by clock signal TSM. The Processor MB register is used in this instance to store the displacement for the effective address calculations.) The outputs from the IR register are decoded immediately to produce the logic 1 LDA and logic 0 NON ACD INST signals.

Upon decoding the instruction it is important to analyze the switching structure of the SX and D Multiplexers and the flow of the Adder output data relative to the instruction decode. In the case of the LDA example there is no change in the SX Multiplexer input control signals and though it is slightly repetitious it reviewed again in the following discussion. The next

step in the logical operation is to move the contents of the PC register through the Adder for the effective address calculations. This necessitates that the proper SX SEL signals (for the SX Multiplexer) and the SEL-N signal (for the Adder Output Data Manipulator gates) be generated. Referencing the SX SEL signals listed in paragraph 4-2.5 the signals required to select the PC for input are:

SX-H-SEL	SX-L-SEL
1	0

As shown on Drawing 001-000092 (sheet 3) the states of IR Bits 6 and 7 are sampled for IR 6 in the reset state and IR 7 in the set state, i.e., $\overline{\text{MEM 6}} = 0$ and $\overline{\text{MEM 7}} = 1$. This configuration of IR Bits 6 and 7 define Relative addressing where the 8 bit signed displacement (- 200 to + 177) encoded in the instruction is added to the address in the PC. Therefore if $\overline{\text{IR 7}}$ is a logic 0 and IR 6 is a logic 0 this condition will be satisfied and the logic 0 will be held on the $\overline{\text{FORCE SX - H - SEL}}$ line and the logic 1 will be held on the $\overline{\text{FORCE SX - L- SEL}}$ line. The signals on both lines leave the ALU Control and are inverted in the Carry logic section before being applied to the control inputs of the SX Multiplexer. This inversion results in the correct (PC select) logic levels on the SX-H-SEL (logic 1) and SX-L-SEL (logic 0) Control lines. It is noted that signal $\overline{\text{EFA}}$ is instrumental in enabling the $\overline{\text{FORCE SX}}$ logic in the ALU Control section. EFA and $\overline{\text{EFA}}$ are produced in the IR register decoder section and are produced by the fact that a Fetch cycle is in progress, and an LDA decode is present.

At this point the preceding discussion described how the contents of the PC are applied to one set of Adder inputs. The other set of Adder inputs are input by the D Multiplexer and represents the effective address displacement. It is important here to digress slightly to consider effective address calculating,not just for the LDA example being explained but for all instructions.

[The least significant 8 bits of a memory reference instruction are the displacement: a number which, when added to the base number, produces the address. The displacement may be positive or negative. This displacement resides in the least significant 8 bits of the processor MB registers (since the most significant 8 bits contain the instruction.) The least-significant 8 bits of the D Multiplexer passes the displacement into the Adder. The

D Multiplexer for the most-significant 8 bits sends 0's into the Adder for the most-significant bit, if the displacement is positive. The displacement is considered negative if bit 8 is a one and page zero addressing is not being used. (Page zero addressing is specified when bits 6 and 7 of the instruction word is zero. If the displacement is negative, the most significant D Multiplexer bits are complemented (D-M-COM is true) and bits 0-7 are 1's. From these two halves of the multiplexer formed is a 16 bit wide number which represents the positive or negative displacement. Through the facilities of the Adder this number is added to the base number. The base number is either the PC contents, or AC 2 or 3 contents; the SX Multiplexer selects one of the three possible sources to input to the Adder.

For STA, LDA, ISZ, DSZ, JMP and JSR instructions, the calculated EFA is put onto the MBO bus at MA load time. The memory address register loads the address to be referenced. For JMP or JSR, the EFA is loaded into the PC as well as the MA. An effective address is also calculated in the same manner (relative to a base number of 0) during the second machine cycle (DEFER) of an indirect memory reference instruction.]

To return to the discussion of the LDA Fetch, the data appearing at the Adder outputs represents the effective address data. This data is placed on the MBO Bus to Memory by the $\overline{\text{ADDER TO MEM}}$ logic 0 signal. (As noted on the Functional Block Diagram INC PC, $\overline{\text{LOAD PC}}$, = 0 ENABLE, LOAD AR, $\overline{\text{PC TO MEM}}$,$\overline{\text{ADDER TO MEM}}$ signals (from the ALU Control logic) are routed to terminations in the Carry logic section, and from these points to the various sections of the ALU.) The $\overline{\text{MA LOAD}}$ signal comes true synchronously with $\overline{\text{ADDER TO MEM}}$ becoming false. This is due to the fact both signals are controlled by PTG3; $\overline{\text{ADDER TO MEM}}$ by $\overline{\text{PTG3}}$ and $\overline{\text{MA LOAD}}$ by PTG3. However, since the MBO data (MD1-MD15) is present prior to $\overline{\text{MA LOAD}}$, $\overline{\text{MA LOAD}}$ simply strobes the data into the MA register.

The Major States section of the 4 Bit Discretes register is switched to the Execute state (or 1 1 state) by $\overline{\text{PTG3}}$ and the trailing edge of CPU CLK. The inputs to these two flip-flops (of the 4 Bit Discretes register) are controlled by a pair of gates (and one inverter). These gates will enable logic 1's to be loaded into both stages if IR 5 is not set and a logic 0 is present on the $\overline{\text{HAS E CYCLE}}$ line. $\overline{\text{HAS E CYCLE}}$ is derived from a gating tree which

determines if the instruction has an Execution cycle and if the machine state is such that an Execution cycle should be performed next. The next Memory cycle is also started at PTG3 time. At PTG1 time of the Execute cycle the logic 1 NON ACD INST, the logic 0 $\overline{E}$ signal, and a logic 1 ADDER = 0 signal (ADDER = 0 indicates in this case both Multiplexers have shut off all inputs to the Adder) allow the ALU Control to generate a INC PC signal. As mentioned previously the $\overline{\text{MEM}}$ Bus data is automatically latched into the Processor MB register by MEM LATCH (on the trailing edge of TSM). The logic 1 E signal from the Major States decoder gates, along with a logic 1 on the $\overline{\text{FORCE D-L-SEL}}$ line, allows the ALU Control to generate a logic 0 $\overline{\text{MEM OUT}}$ signal. $\overline{\text{MEM OUT}}$ is sent to the Carry logic where it enables a pair of gates which place logic 1's on the D-M-H-SEL and D-L-H-SEL lines to the D Multiplexer. At this time a logic 0 is present on the D-L-SEL line to the D Multiplexer (since $\overline{\text{FORCE D-L-SEL}}$ is not true at this time). Referencing the Multiplexer selection codes specified in paragraph 4-2.5, the Processor MB is selected with a code of D-H-SEL= 1 and D-L-SEL = 0. This code allows the contents of the Processor MB to pass through the Adder, and since an ALC decode is not present, $\overline{\text{FORCE SEL X}}$ and $\overline{\text{FORCE SEL Y}}$ are logic 1's to produce a logic 1 SEL-N (Select No Shift). Thus, the Adder outputs are applied to the SUM Bus without modification. It should be noted that there is no interference from the SX Multiplexer at this time, since neither $\overline{\text{ALC}}$ or $\overline{\text{EFA}}$ are present (during the Execute cycle of an LDA) to activate the $\overline{\text{FORCE SX}}$ logic of the ALU Control. Hence, the SX Multiplexer inputs zeroes to the Adder which are added to the MB data provided by the D Multiplexer.

One of the last operations of the LDA Execute cycle occurs (at the end of PTG3 time) in the ALU Control where (with an F SET logic 1 signal present) a $\overline{\text{PC TO MEM}}$ logic 0 signal is generated. This signal transfers the contents of the PC out to the Memory via the MBO Multiplexer and MBO Bus. Since F SET is true MEM CY SET is present as a logic 1, allowing an $\overline{\text{MA LOAD}}$ logic 0 signal to be produced at the end of PTG3 time. $\overline{\text{MA LOAD}}$ loads the PC data present on the MBO Bus into the Memory MA register (via the MD1 thru MD15 lines). $\overline{\text{PTG3}}$ and CPU CLK are instrumental in switching the Major States flip-flops back to the 0 0 state which is decoded as the Fetch State. $\overline{\text{HAS E CYCLE}}$, as mentioned earlier, enables

the input control gates of the Major States flip-flops to load in the 1 1 code for the Execute State decode. However, one of the inputs to the $\overline{\text{HAS E CYCLE}}$ gates is provided by the output from the 2nd Major States flip-flop (or the 4th Stage of the 4 Bit Discretes register). Thus, as soon as this flip-flop becomes set for the Execute State, it disables the gates producing the logic 0 $\overline{\text{HAS E CYCLE}}$ signal, causing $\overline{\text{HAS E CYCLE}}$ to switch to a logic 1. Therefore, when $\overline{\text{PTG3}}$ occurs to load this register, the logic 1 on the $\overline{\text{HAS E CYCLE}}$ line will (assuming no Program Interrupt is present) cause the input control gates (of the Major States flip-flops) to hold logic 0's on the inputs to both flip-flops. Under these conditions the trailing edge of CPU CLK will load the 0 0 configuration into the flip-flops as a command for the next Fetch Cycle.

The last operation occurs after PTG3 time, and during PTG0 of the next cycle and writes data from the AR register into the selected accumulator. During PTG3 time a LOAD AR signal was produced in the ALU Control which loaded the AR register with the Adder output data present on the SUM Bus (SUM0 - SUM15). A $\overline{\text{WRITE AC}}$ signal generated in the ALU Control causes (in the logic 0 state) an $\overline{\text{AC WRITE}}$ signal to be produced in the Carry logic section at PTG0 time. The AC-H-WRITE and AC-L-WRITE Accumulator address signals are produced in the Carry Logic by $\overline{\text{PTG3}}$ and the trailing edge of CPU CLK, and logically coded in respect to the configuration on the $\overline{\text{IR3}}$ and $\overline{\text{IR4}}$ lines (which specifies the Accumulator to be loaded by the LDA instruction). Therefore during PTG0 of the next cycle the specified Accumulator AC0, AC1, AC2, or AC3 copies the data held in the AR register.

4-2.7 Carry Bit Functions

The Carry bit is a seventeenth, and the most significant bit for arithmetic operations. The primary factors in determining the state of Carry are: the base value for Carry (selected by the instruction); the previous state of Carry; the Carry generated from the Adder during an arithmetic operation. The signals CR and CG represent the Carry from the Adder. The state of the Carry bit is determined by the last stage of a 4 Bit Discretes register which is shown at the left center of the Drawing 001-000093. This flip-flop stage is loaded at the end of every

machine cycle (CPU CLK of PTG3) from the conditions determining Carry. The output from this stage produces the signal CRY, which goes to the carry light on the Console, and is the current state of Carry.

Shown just above the Carry stage of the register on Drawing 001-000093 are two open-collector gates whose outputs are tied together: These gates adjust the current carry to reflect the base value which has been chosen. This value may be: the current state of Carry, the complement of the current state of Carry, one, zero. This adjusted value is then logically exclusive OR'd with the Carry out of the Adder. This new value goes to one of the Data Manipulator gates and becomes the new value of Carry, loaded at the end of the cycle.

4-2.8 Data Paths

A brief description of the data paths for major instructions are provided in the following discussion.

4-2.8.1 ARITHMETIC. Arithmetic instructions are performed by moving the contents of 1 or 2 accumulators along a data path through the logic elements, and loading the result into a destination accumulator.

4-2.8.2 MOV (Move). The contents of the selected accumulator (instruction bits 1, 2) are read out of ACSX through SX-MULT (multiplexer) into the Adder. The D-MULT selects no input and puts zero into the Adder. The Adder adds the data to zero and the result goes through the Data Manipulator gates (where it may be shifted or swapped if the instruction so specifies) and is loaded into the AR. The data is then stored in the instruction selected accumulator in both ACSX and ACD.

4-2.8.3 COM (Complement). The data paths are the same as move. When the data from ACSX passes through SX-MULT, it is complemented (SX-COM is true (high)) and then added to zero in the Adder. This operation is the logical or arithmetic one's complement.

4-2.8.4 NEG (Negate). The same as COM except that as the data is added to zero in the Adder, the signal "PLUS ONE" is true (high) on the least significant bit, causing (1) one to be added to the sum of the complemented data and zero. This operation is the arithmetic two's complement.

4-2.8.5 INC (Increment). The same path as MOV. Data is not complemented, but 1 is added. ("PLUS ONE " is true).

4-2.8.6 ADD. The AC selected as "source" is read out of ACSX as the AC selected as "destination" is read out of ACD. The data passes through the respective multiplexers and is added in the Adder. The sum may be shifted or swapped by the Data Manipulator gates. The sum is loaded into the AR and then into the accumulator specified as AC destination (both ACSX and ACD).

4-2.8.7 AND. Same data paths as ADD. As the data passes through the Adder, the signal "AND" is true, causing the pairs of inputs to be logically ANDed.

4-2.8.8 ADC (add the complement). Same data paths as ADD. As source accumulator data passes through the SX multiplexer, the signal SX-COM is true, causing the data in the source accumulator to be complemented before being added to the data from the destination accumulator (ACD).

4-2.8.9 SUB (Subtract). The same paths as ACD, except that as the ACD data is added to the complemented ACSX data in the Adder, the signal "PLUS ONE" is true, adding one to the sum, (one binary number is subtracted from another by adding its complement plus one.)

4-2.8.10 I/O. For Input/Output instructions, the Processor uses the MB register on the currently selected Memory Assembly and an I/O data buffer. The data path for Data IN is: Data to Memory MB register. Memory MB data to Processor MB (using $\overline{\text{MEM}}$ Bus). MB, through the D-MULT and the Adder, into the AR. From the AR data is loaded into the destination accumulator.

The path for Data OUT is: ACD through the D-MULT and Adder onto the MBO bus to the Memory MB register. Memory MB to Data Bus.

4-2.8.11 JMP. For a jump instruction, the Processor calculates the EFA and sends it to Memory as address. The Processor features multilevel indirect addressing which permits "N" number of EFA calculations, however only the first may be relative to other than base 0. The second, third, fourth, etc., during Defer, use the contents of the Memory location given by the first calculation as a 15 bit displacement.

this location are loaded into the MB during the read portion of the Memory cycle, and the EFA calculation occurs during the write portion.

4-2.8.12 JSR. JSR uses the same data paths as jump. In addition, during the FETCH cycle of the subsequent instruction, or during a DCH or PI cycle the AR (which contains PC + 1) is loaded into AC3.

4-2.8.13 STA. The first cycle of an STA (the FETCH cycle) is an effective address calculation. During the second cycle (EXECUTE), when the Memory is referencing the calculated effective address, the SX-MULT passes the contents of the selected accumulator to the Adder, where it is added to 0 and sent out via the MBO bus. It is then loaded into the Memory MB register and written into Memory at the effective address.

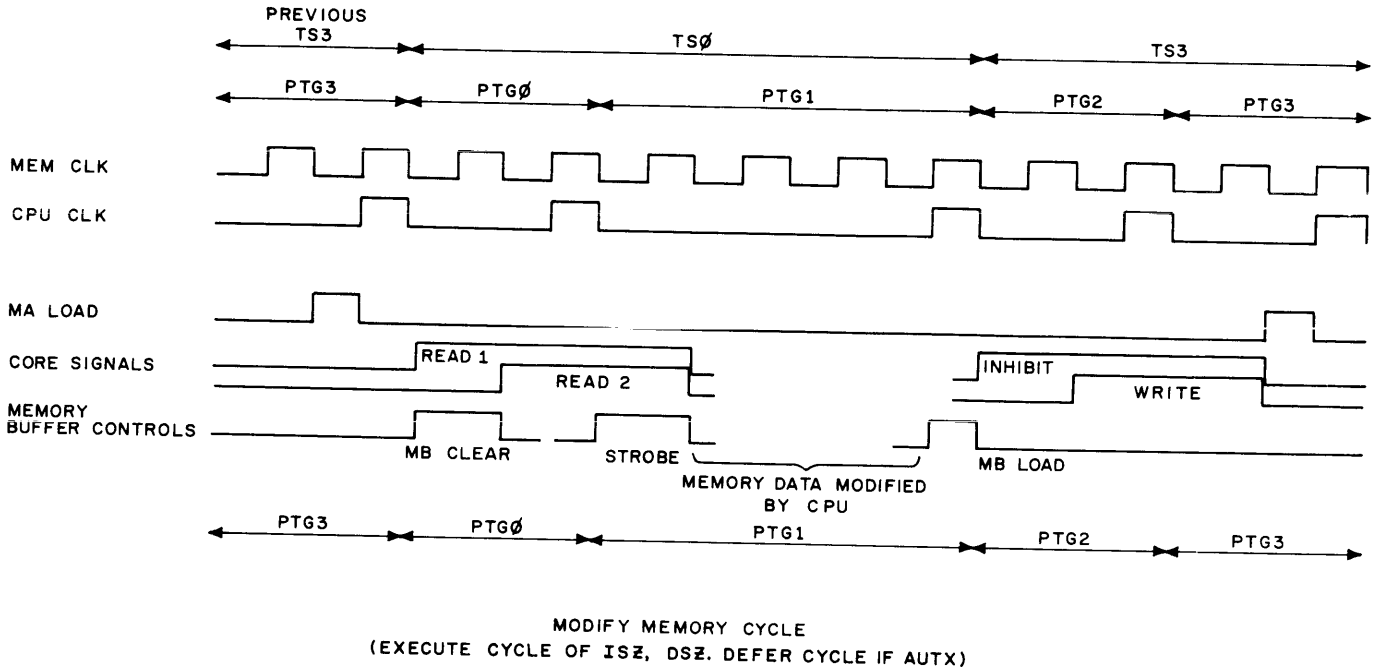
4-2.8.14 LDA. The FETCH cycle of an LDA is an effective address calculation. During the execute cycle the calculated Memory address is strobed. Data from the Memory passes over the $\overline{\text{MEM}}$ bus, is loaded into the Processor MB, passes through D-MULT, passes through the Adder (the SX-MULT side of the Adder is 0), and through the Data Manipulator gates into the AR. From the AR it is loaded into the selected destination accumulator during the fetch cycle of the next instruction.

4-2.8.15 ISZ. The FETCH cycle is an EFA calculation. During the EXECUTE cycle (which is extended in duration), the data from the calculated address is read into the Processor MB, passes through the D-MULT and through the Adder, added to 0 from the SX-MULT. However, the "PLUS ONE" signal is true therefore plus one is added by the Adder. Data goes out to the Memory along the MBO bus and is written back into the same address. All this occurs during one Memory cycle which is extended from 800 ns to 1000 ns where PTG1 is lengthened to 400 ns from 200 ns.

4-2.9 Nova 800 Timing

The timing diagrams for the Modified Memory Cycle, Programmed In, Programmed Out, IO Skip, standard DCH (In, Out, and INC) and Fast DCH (In, Out, and INC), appear on Drawing 001-000121 sheets 2 and 3. These timing diagrams are also provided here for reference purposes in Figures 4-3 thru 4-12.

4-2.9.1 Modify Memory Cycle. The Modify Memory Cycle is basically the timing required for modifying a selected memory location, with modification restricted to incrementing, decrementing, either in an ISZ or DSZ instruction, a DCH increment operation, or as part of addressing an autoindexing location. The timing diagram of Figure 4-3 shows the last CPU CLK of the previous cycle, including the MA LOAD, which establishes the address of the word to be modified, (or the instruction referencing an autoindex location). The normal Memory signals are generated and the Strobe pulse loads the instruction data into the Memory MB register. From this point, as described previously, the instruction data is loaded into the IR register and the Processor MB register via the $\overline{\text{MEM}}$ Bus. It will be noted as shown on Figure 4-3 that the PTG1 timing interval is extended for an extra 200 nanoseconds. This extension is defined as a "STUTTER" operation and allows additional operations variant from unmodified Memory cycle to be performed during the Stutter. A point to remember about Stutter operations is that only the PC CLK and CPU CLK are interrupted. The MEM CLK is not interrupted by the Stutter. The logical signals implementing the Stutter are shown on Drawing 001-000092, sheet 1. The output CPU CLK is produced by a pair of parallel AND gates, which are driven by a count down pair of flip-flops (CLK-A and CLK-B) at one half the frequency of the MEM CLK. MEM CLK is 50 nanoseconds wide and occurs every 100 nanoseconds. CPU CLK is 50 nanoseconds wide and occurs every 200 nanoseconds. The CPU CLK gates are also controlled by the output of a gating tree, which has one of its path requirements satisfied by PTG1, $\overline{\text{MB LD EN}}$, and $\overline{\text{WRITE SYNC}}$. $\overline{\text{WRITE SYNC}}$ is basically a (MEM CLK derived) timing signal for the $\overline{\text{MB LD EN}}$ Stutter functions, i.e., when during the Memory Cycle the Stutter should occur. $\overline{\text{MB LD EN}}$ will become low on three distinct occasions; for PTG1 with DCH INC EN (Data Channel Increment Enable) for the E (Execute) Cycle of a NON ACD INST (representing that an ISZ or DSZ or JMP or JSR instruction is present), and for the D (Defer) Cycle of an AUT INC + DEC (Autoindex) instruction. The $\overline{\text{MB LD EN}}$ gates are shown on Drawing 001-000092, sheet 3. The key to the Stutter control logic is a 4 Bit Discrete register (Reference Drawing 001-000092, sheet 1) which is interconnected into the CPU CLK (Stutter) gating tree. This register is always loading its stages in parallel



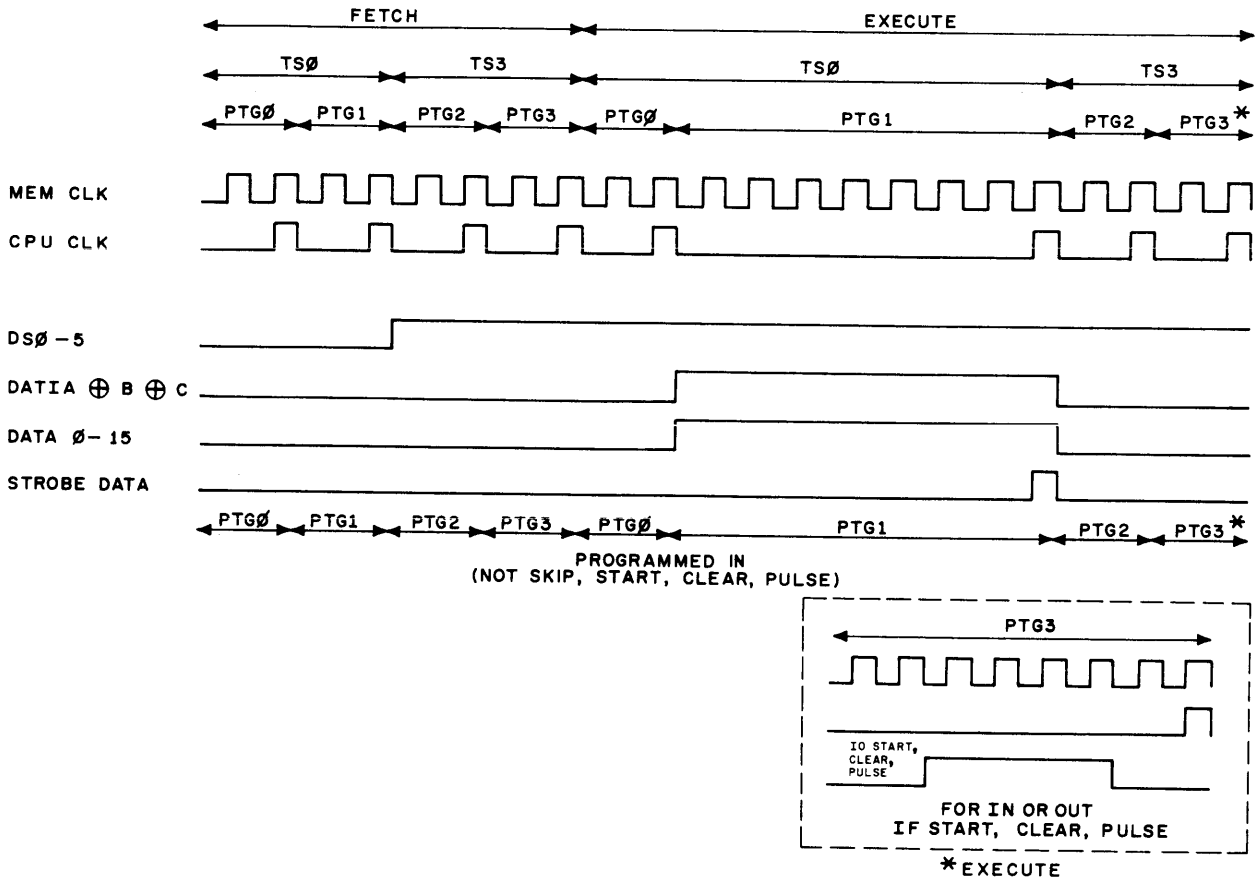
MODIFY MEMORY CYCLE
(EXECUTE CYCLE OF ISZ, DSZ. DEFER CYCLE IF AUTX)

NOTE: All Times typical. PTG Signals 200 nanoseconds except where indicated. MEM CLK is 100 nanoseconds from edge to edge between any two consecutive MEM CLK's.

Figure 4-3. Modify Memory Cycle

when the $\overline{\text{IO STUTTER}}$ signal is not present. The third stage of the register is wired such that it provides a feedback path from an AND gate controlling the OR gate used to actually perform the CPU CLK inhibit function. If the output from the AND gate is high, CPU CLK is inhibited. However, the next CLK-B to occur will latch the (high) output from the AND gate into the register, thereby producing a high output from the same stage, which being fedback around to the (logic0) OR section of the AND gate, disables the AND gate. Therefore, after WRITE SYNC becomes set, the next CPU CLK is inhibited. The unique feedback path then disables the CPU CLK inhibit gating tree, allowing the next CPU CLK to be generated. During the extended cycle the Memory Data is modified, incremented or decremented, through the Adder and reloaded into the Memory MB register. This modified data is subsequently rewritten back into Memory to end the Cycle.

4-2.9.2 Programmed In Timing. Figure 4-4 is a timing diagram for the Programmed In I/O instructions. The Programmed In timing is dependent on whether an I/O SKIP function has been coded into the instruction or not. The I/O SKIP timing for I/O In or I/O out is shown on Figure 4-6. The DS0-5 signal represents the data on the Device Select lines, which are present at the start of PTG2 of the Fetch cycle. The $\overline{\text{DS0}}$ thru $\overline{\text{DS5}}$ data on these lines represents the configuration loaded into IR bits 10 thru 15 selecting the I/O device to input data to the Processor. As shown on Figure 4-4 an IO STUTTER occurs on the trailing edge of the first CPU CLK of the I/O Execute Cycle. (Actually PTG1·IO·E·IR7·$\overline{\text{FAST DCH}}$ produces IO STUTTER. PTG1·IO·E·IR7 also produces READ IO.) The IO STUTTER function performed in the Timing logic section (Reference Drawing 001-000092, sheet 1) utilizes the same 4 Bit Discretes register, but in a shifting mode rather than the parallel loading mode. IO STUTTER extends the PTG1 cycle from 200 nanoseconds to 800 nanoseconds. When IO STUTTER becomes low, it enables (through an inverter) the Shift control input of the register. The low $\overline{\text{IO STUTTER}}$ signal also enables the inhibit gating tree, shutting off the CPU CLK. The trailing edge of the first CLK-B pulse (actually the second Clock Pulse of PTG1) shifts a logic 1 (from the output of an inverter being driven by the normal logic 0 output from the second stage of the register) from the DS input of the register into the first stage. This produces a signal MID used to strobe all of the DATO, STRT, CLR, and IOPLS output lines. The DATO output

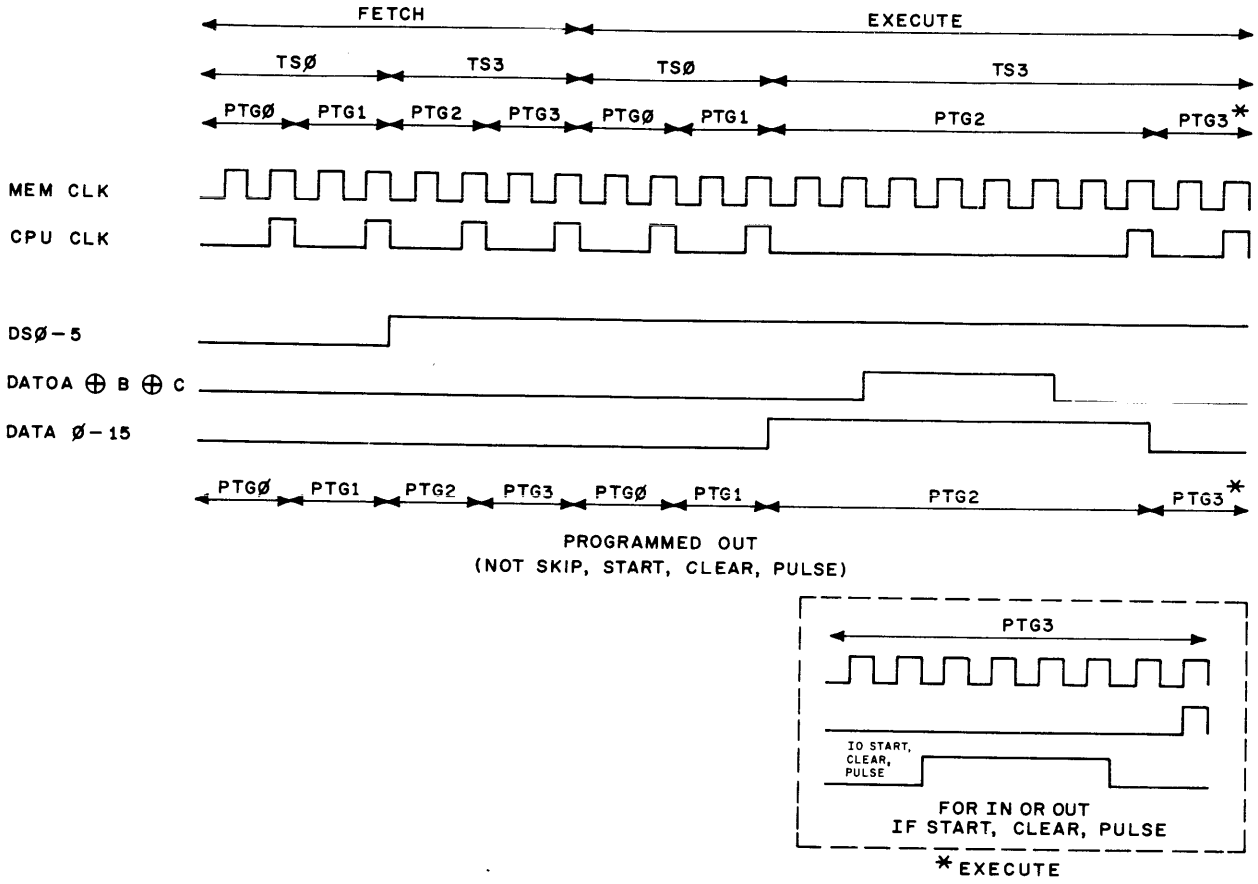


NOTE: All Times typical. PTG Signals 200 nanoseconds except where indicated. MEM CLK is 100 nanoseconds from edge to edge between any two consecutive MEM CLK's.

Figure 4-4. Programmed In Timing Diagram

lines are functional during the Execution of Programmed Out I/O instructions: The trailing edge of the second CLK-B pulse shifts the logic 1 into the first two stages of the register. (The logic 1 from the second stage causes the inverter driving the DS input of the register to switch the input to a logic 0. The trailing edge of the third CLK-B pulse shifts a logic 1 into the third stage of the register, thereby disabling the inhibit gating tree and releasing the next CPU CLK to occur. The third CLK-B pulse also shifts a logic 0 into the first stage of the register, causing the MID signal to switch to the false (logic 0) state. As mentioned previously PTG1·IO·E·IR7 produce READ IO. READ IO in turn produces a logic 0 $\overline{\text{OMIT STROBE}}$ (in the ALU Control) which in turn enables a MB LOAD to be produced once the CPU CLK is released. This MB LOAD signal is identified on the timing diagram of Figure 4-4 as STROBE DATA. The DATIA, DATIB, or DATIC command the device to place its data on I/O Bus lines $\overline{\text{DATA0}}$ thru $\overline{\text{DATA15}}$. The MB LOAD signal loads the Memory MB register with data present on I/O Data lines $\overline{\text{DATA0}}$ thru $\overline{\text{DATA15}}$. This data in turn is transferred via the $\overline{\text{MEM}}$ bus to the Processor MB register where it is loaded by the trailing edge of TSM (which brings MEM LATCH low). The contents of the Processor MB are then passed through the Adder, the sum of which is loaded into the AR register, and then subsequently into the selected Destination accumulator. As shown on Figure 4-4, PTG3 of the I/O Execute cycle is extended from 200 nanoseconds to 800 nanoseconds if a function pulse (Start, Clear, or IOPLS) is required.

4-2.9.3 Programmed Out Timing. Figure 4-5 is a timing diagram for the Programmed Out I/O instruction. The Programmed Out timing is also dependent on whether an I/O SKIP function has been coded into the instruction or not. The I/O SKIP timing for I/O, In or I/O Out is shown on Figure 4-6. The DS0-5 signal represents the data on the Device Select lines, which are present at the start of PTG2 of the Fetch Cycle. The $\overline{\text{DS0}}$ thru $\overline{\text{DS5}}$ data on these lines represents the configuration loaded into IR bits 10 thru 15 selecting the I/O device to receive data from the Processor. $\overline{\text{IO STUTTER}}$ is also generated during the Programmed Out operations, however the $\overline{\text{IO STUTTER}}$ for Programmed Out is generated on the trailing

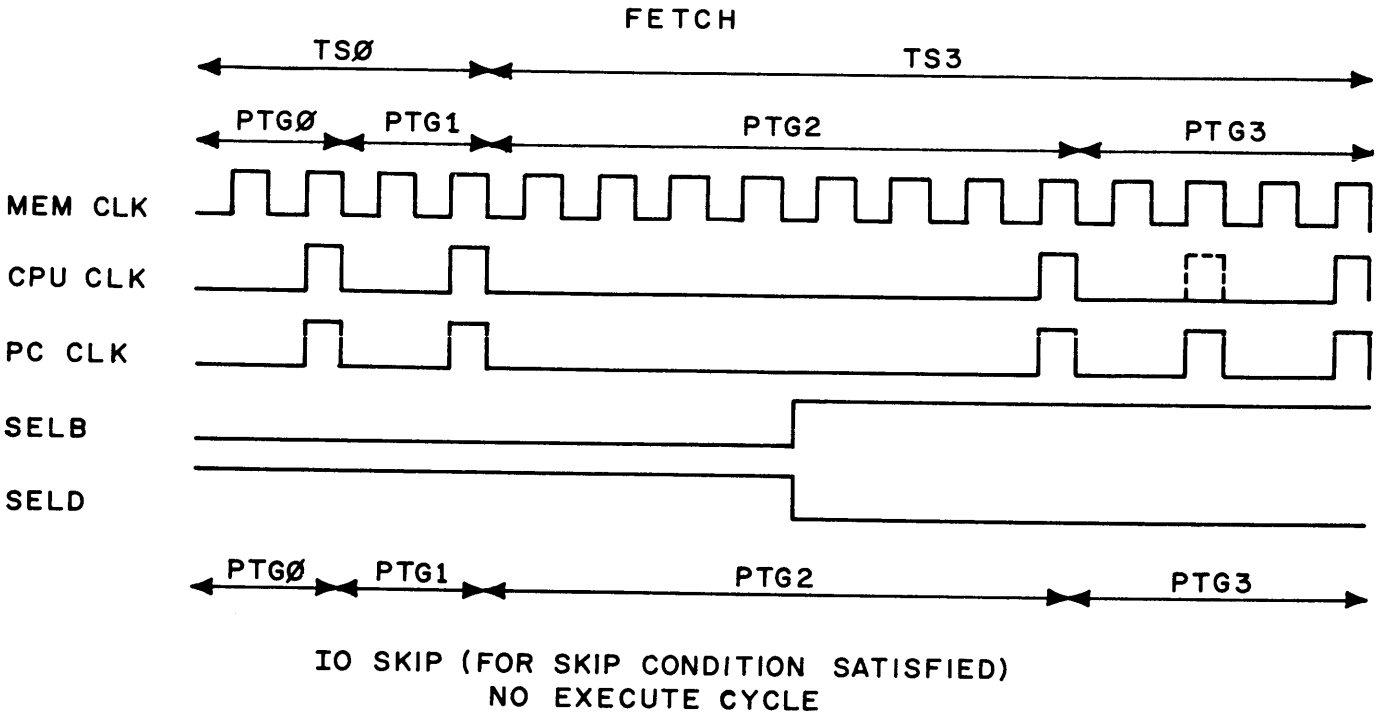


NOTE: All Times typical. PTG Signals 200 nanoseconds except where indicated. MEM CLK is 100 nanoseconds from edge to edge between any two consecutive MEM CLK's.

Figure 4-5. Programmed Out Timing Diagram

edge of PTG1, effectively stretching out PTG2 time from 200 to 800 nanoseconds. $\overline{\text{IO STUTTER}}$ is produced in this case (Reference Drawing 001-000092, sheet 4) by $\overline{\text{PTG2}}$·E·IO·$\overline{\text{IR7}}$ all present in their true logic state (along with a logic 0 FAST DCH signal, representing no Fast DCH Operation). The logic 0 $\overline{\text{IO STUTTER}}$ signal implements the 4 Bit Discretes register and the CPU CLK inhibit gating tree logic in the Timing logic section (Reference Drawing 001-000092) exactly as described in the previous paragraph. The output data is accessed from the Destination accumulator (ACD) and transferred through the Adder onto the MBO bus to the Memory MB register. The $\overline{\text{PTG2}}$ logic 0 signal is used directly as the $\overline{\text{DRIVE IO}}$ signal which gates the contents of the Memory MB register out onto the I/O Data Bus. During the Memory Cycle the $\overline{\text{OMIT STROBE}}$ line is again held to a logic 0 by the presence of a logic 1 E·IO signal. The logic 1·E IO signal together with a logic 0 HALT signal (HALT is not true) generate a $\overline{\text{FORCE D-L-SEL}}$ signal which enables the D-Multiplexer to pass the ACD data onto the Adder. As mentioned previously $\overline{\text{OMIT STROBE}}$, in the logic 0 state, inhibits the Strobe and produces MB LOAD instead which loads the ACD data (present on the MBO Bus via the Adder output and MBO Multiplexer) into the Memory MB. Also as mentioned previously signals MID and PTG2 strobe the DATOA, DATOB, and DATOC output gates, allowing the properly enabled gate to place a logic 1 on its associated I/O Bus line. As shown on Figure 4-5, PTG3 of the I/O Execute cycle is extended from 200 nanoseconds to 800 nanoseconds if a function pulse (Start, Clear, or IOPLS) is required.

4-2.9.4 IO SKIP Timing. Figure 4-6 is an IO SKIP timing diagram for a satisfied program specified skip condition, e.g., skip on busy. The IO SKIP test and qualification is performed during the Fetch cycle. The IO SKIP code present in the IR is decoded to produce an IO SKIP signal which is gated together with PTG2 (Reference Drawing 001-000092, sheet 4) to generate a logic 0 $\overline{\text{IO STUTTER}}$ signal which extends PTG2 from 200 to 800 nanoseconds. $\overline{\text{IO STUTTER}}$ implements the same CPU CLK inhibit logic in the Timing section exactly as described earlier in the Programmed In timing discussion. This timing extension allows sufficient time for interrogating and testing the SELB (Selected Busy) and SELD (Selected Done) logic of the selected I/O device. A dotted or phantom CPU CLK pulse is shown on the timing diagram of Figure 4-6. This phantom pulse represents a Stutter extension from 200 to 400 nanoseconds for PTG3 if the IO SKIP condition is satisfied (by the states of the SELB and SELD lines). This



NOTE: All Times typical.
PTG Signals 200 nanoseconds except where indicated.
MEM CLK is 100 nanoseconds from edge to edge between any two consecutive MEM CLK's.

Figure 4-6. IO Skip Timing Diagram

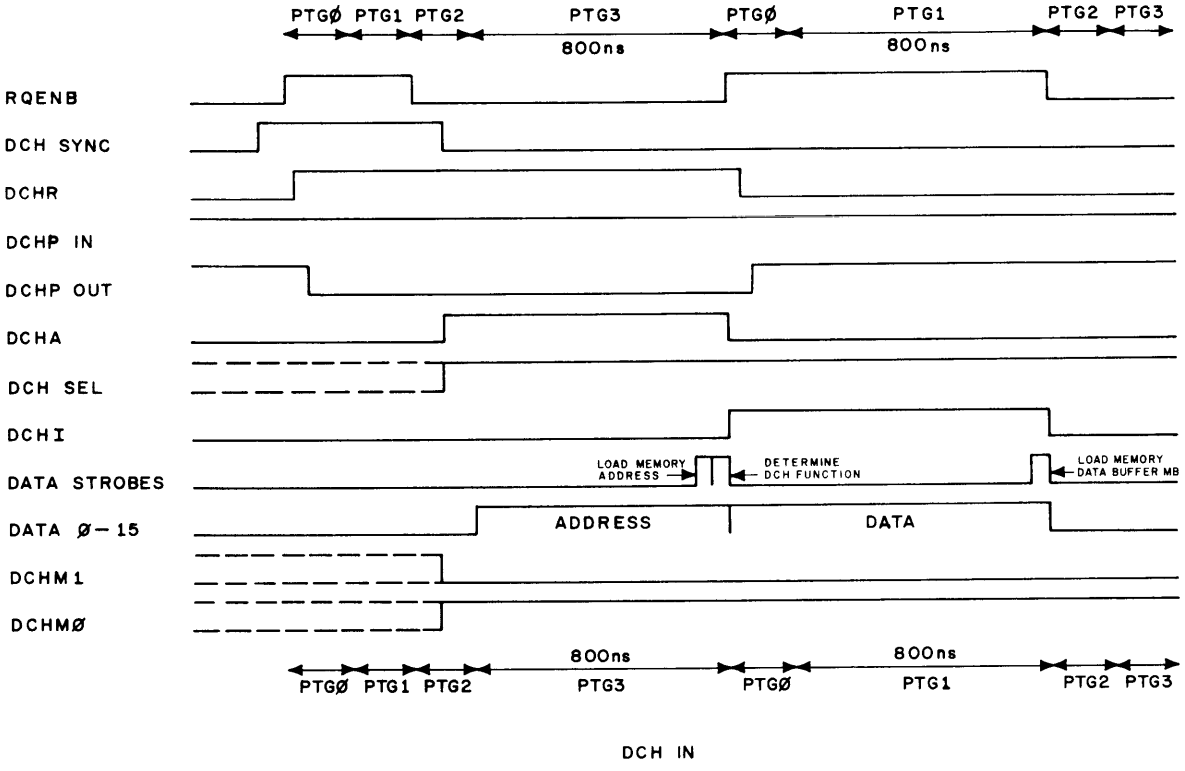
200 nanosecond extension is provided by the $\overline{\text{FETCH SKIP}}$ (AND-OR-INVERT) gates in the ALU Control logic (Reference Drawing 001-000092, sheet 3). A flip-flop driven by the PC CLK (not effected by the Stutter) and enabled by PTG2 determines the length of time $\overline{\text{FETCH SKIP}}$ will remain a logic 0. The PC CLK coincident with the end of PTG2 sets the flip-flop, and if IO SKP SYNC, IO SKIP, and $\overline{\text{PROTECT}}$ are all present as logic 1's, a logic 0 $\overline{\text{FETCH SKIP}}$ will be generated. $\overline{\text{FETCH SKIP}}$ from the logic 0 level inhibits the CPU CLK gates in the Timing logic section. The next PC CLK (by virtue of the logic 0 on the PTG2 line) clears the flip-flop to the reset state, raising $\overline{\text{FETCH SKIP}}$ to a logic 1, thereby removing the inhibit from the CPU CLK gates. The Major States flip-flops remain set to the Fetch (or 0 0) condition due to the logic 1 IO SKIP signal which enables a series OR gating configuration, the output of which places a logic 0 on the $\overline{\text{HAS E CYCLE}}$ line to the input gates of the Major States flip-flops (Reference Drawing 001-000092, sheet 2). Hence, the machine remains in the Fetch state at the end of the IO SKIP Fetch Cycle, and a new Fetch cycle is started. It should be noted that if the Skip condition was satisfied $\overline{\text{FETCH SKIP}}$ in the logic 0 state also produces an INC PC signal. Therefore, the next Fetch will be either PC + 2 or PC + 1 depending on whether the Skip condition was satisfied.

4-2.9.5 Data Channel Timing. Figures 4-7 through 4-12 are timing diagrams of the Data Channel timing. Figure 4-7, 4-8, and 4-9 are for standard speed DCH operations. Figures 4-10, 4-11, and 4-12 are for Fast DCH operations. The basic difference is that the standard DCH cycle has two 800 nanosecond DCHA STUTTER operations in its cyclic sequence and the Fast DCH does not. The initializing sequence is common to all DCH modes. $\overline{\text{A DCHR}}$ signal is ANDed with a logic 0 INH DCH signal to load a logic 1 into the shift input of a 4 Bit Discretes register. $\overline{\text{DCHR}}$ is a logic 0 signal representing a Data Channel transfer Request from the high speed I/O device. INH DCH is normally a logic 0 to enable Data Channel operation and is switched to the logic 1 state with the occurrence of a STOP RQ, HALT, $\overline{\text{KEY}}$, or $\overline{\text{KEYM}}$ signal. The trailing edge of CPU CLK (at PTG1 time) shifts the logic 1 into the first stage of the register. (Reference Drawing 001-000092, sheet 4.) The output from this stage is a logic 1 signal identified as DCH SYNC. DCH SYNC strobes a variety of functional gates;

DCHA STUTTER gate, $\overline{\text{DCHA}}$ gating chain, $\overline{\text{DCHM1}}$ input gate, and the $\overline{\text{DCHM0}}$ input gate. DCH SYNC also is used to hold off lowering RUN SET, in the event the machine receives a HALT or STOP RQ while a DCH operation is in process. DCH SYNC holds RUN SET to the logic 1 level until the DCH operation is completed. DCH SYNC also (Reference Drawing 001-000092, sheet 2) allows a logic 1 to be loaded into the third stage of a 4 Bit Discretes register (Key, Run, DCH, and KEYM). The logic 1 output from this stage is a signal identified as DCH, which also drives an inverter to produce $\overline{\text{DCH}}$. DCH and $\overline{\text{DCH}}$ are also used to assist in deriving the remaining DCHO, DCHI, $\overline{\text{DCHA}}$, $\overline{\text{READ IO}}$, or $\overline{\text{DRIVE IO}}$ signals. (Reference Drawing 001-000092, sheet 4) DCH SYNC is gated with PTG3 to produce an $\overline{\text{IO STUTTER}}$ logic 0 signal, which extends PTG3 out to 800 nanoseconds as described earlier in the Programmed In discussion. This provides sufficient time to load the Memory Address to be operated on (i.e., transferred into, out of, or incremented). The DCH transfer function (in, out, or increment) is also determined at the end of the PTG3 Stutter extension. Depending on whether the DCH is an Output operation or an Input operation, the next stutter sequence will come at PTG2 or PTG1 respectively. It will be noted that signal MID is generated during the $\overline{\text{IO STUTTER}}$ as an output timing function similar to that it performs for the DATO Programmed Out operations. The 400 nanosecond Stutter during PTG1 of a DCH INC operation is accomplished by a logic 1 DCH INC EN signal (Reference Drawing 001-000092, sheet 4) which at PTG1 time produces a logic 0 $\overline{\text{MB LD EN}}$ signal (Reference Drawing 001-000092, sheet 3). $\overline{\text{MB LD EN}}$ is gated with $\overline{\text{WRITE SYNC}}$ to enable the CPU CLK inhibit gates in the Timing logic (Reference 001-000092, sheet 1). The operation of this section of the logic was described earlier during the Modify Memory cycle discussion.

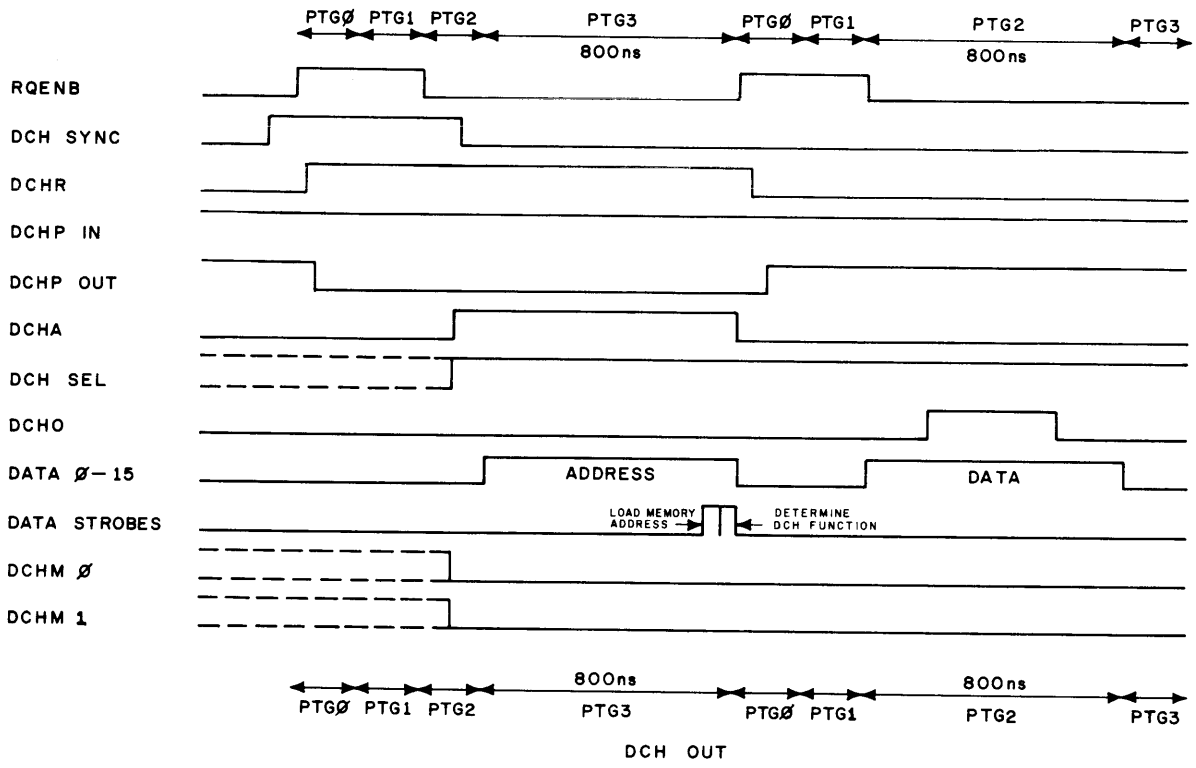
4-2.10 Memory

As mentioned previously each 1K, 2K, or 4K block of memory contains a MA (memory address) register, a MB (memory buffer) register, a set of $\overline{\text{MEM}}$ bus drivers, a set of IO bus drivers, and an input multiplexer which allows the MA and the MB registers to be loaded from either the MBO bus or the IO bus. MA Bits 1-3 (4K) or 1-4 (2K) determine



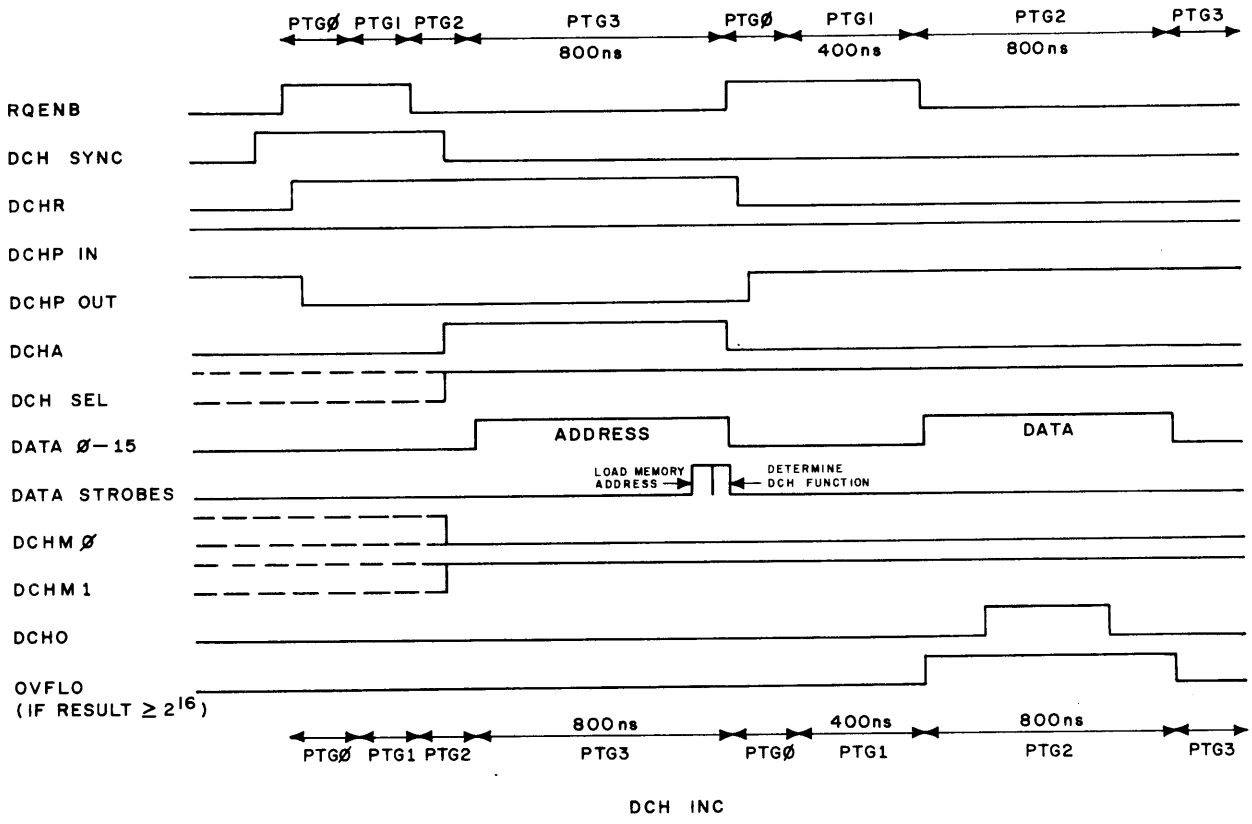
NOTE: All Times typical.
PTG Signals 200
nanoseconds except
where indicated.

Figure 4-7. DCH In Timing Diagram



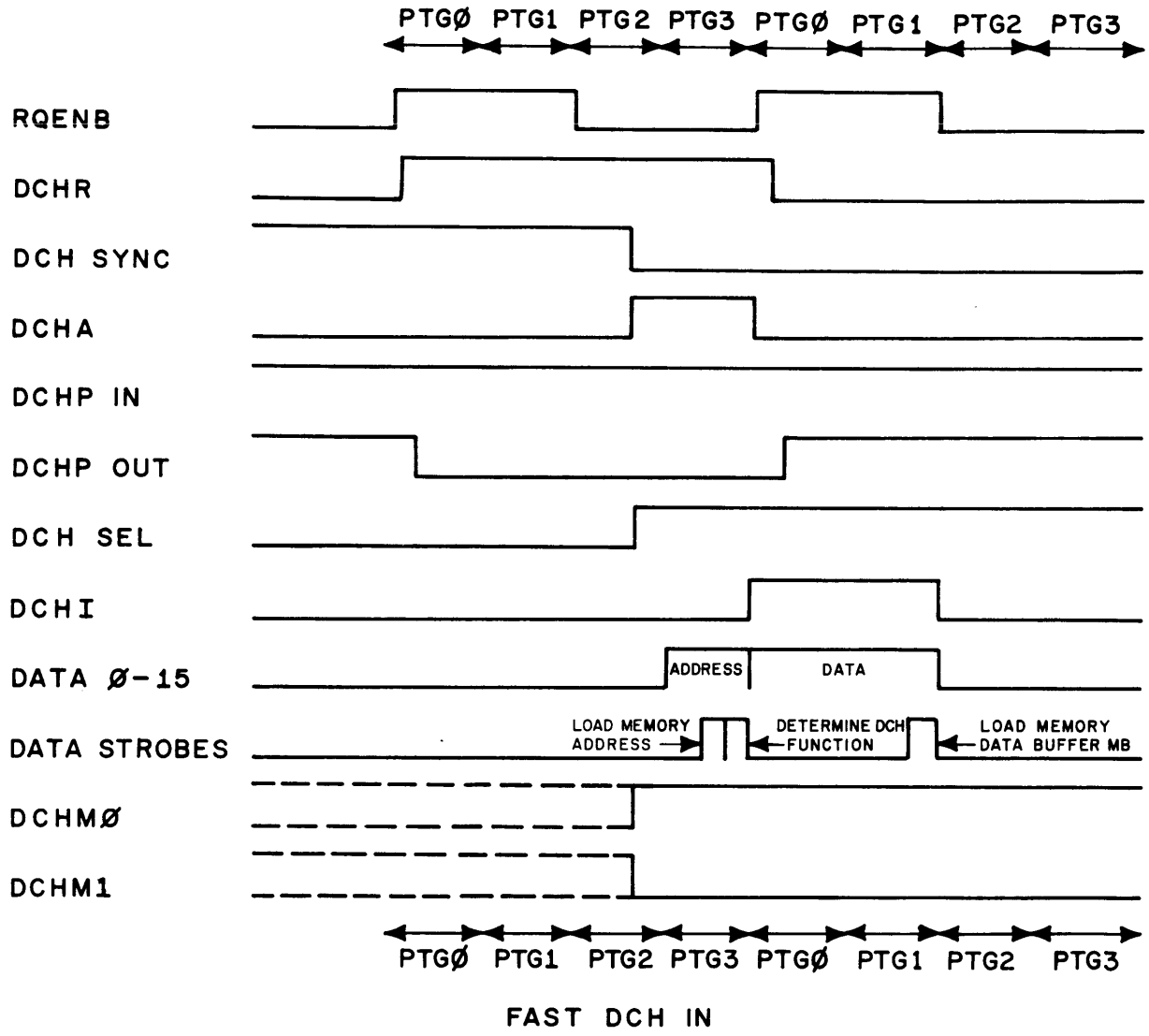
NOTE: All Times typical. PTG Signals 200 nanoseconds except where indicated.

Figure 4-8. DCH Out Timing Diagram



NOTE: All Times typical. PTG Signals 200 nanoseconds except where indicated.

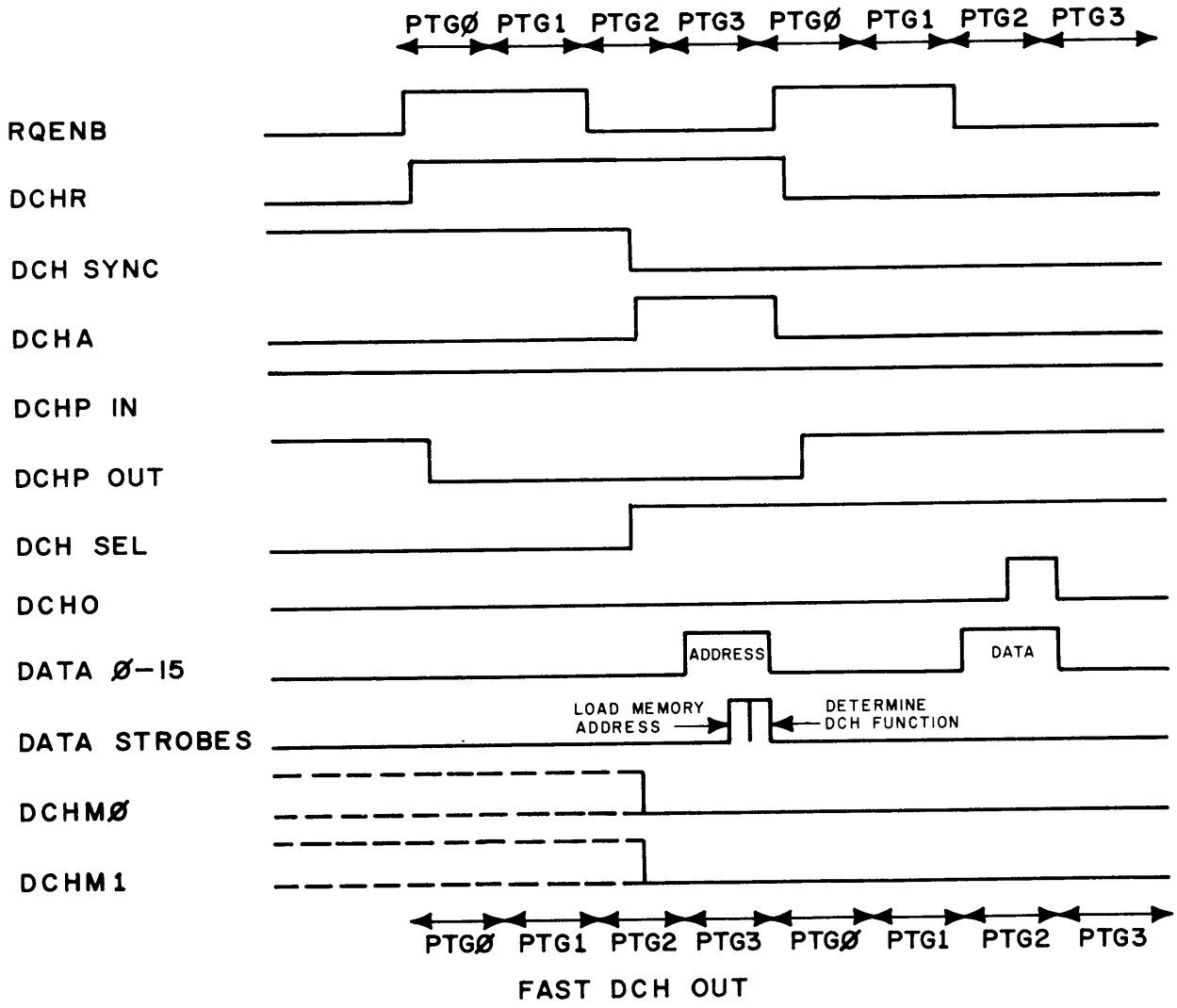
Figure 4-9. DCH INC Timing Diagram



FAST DCH IN

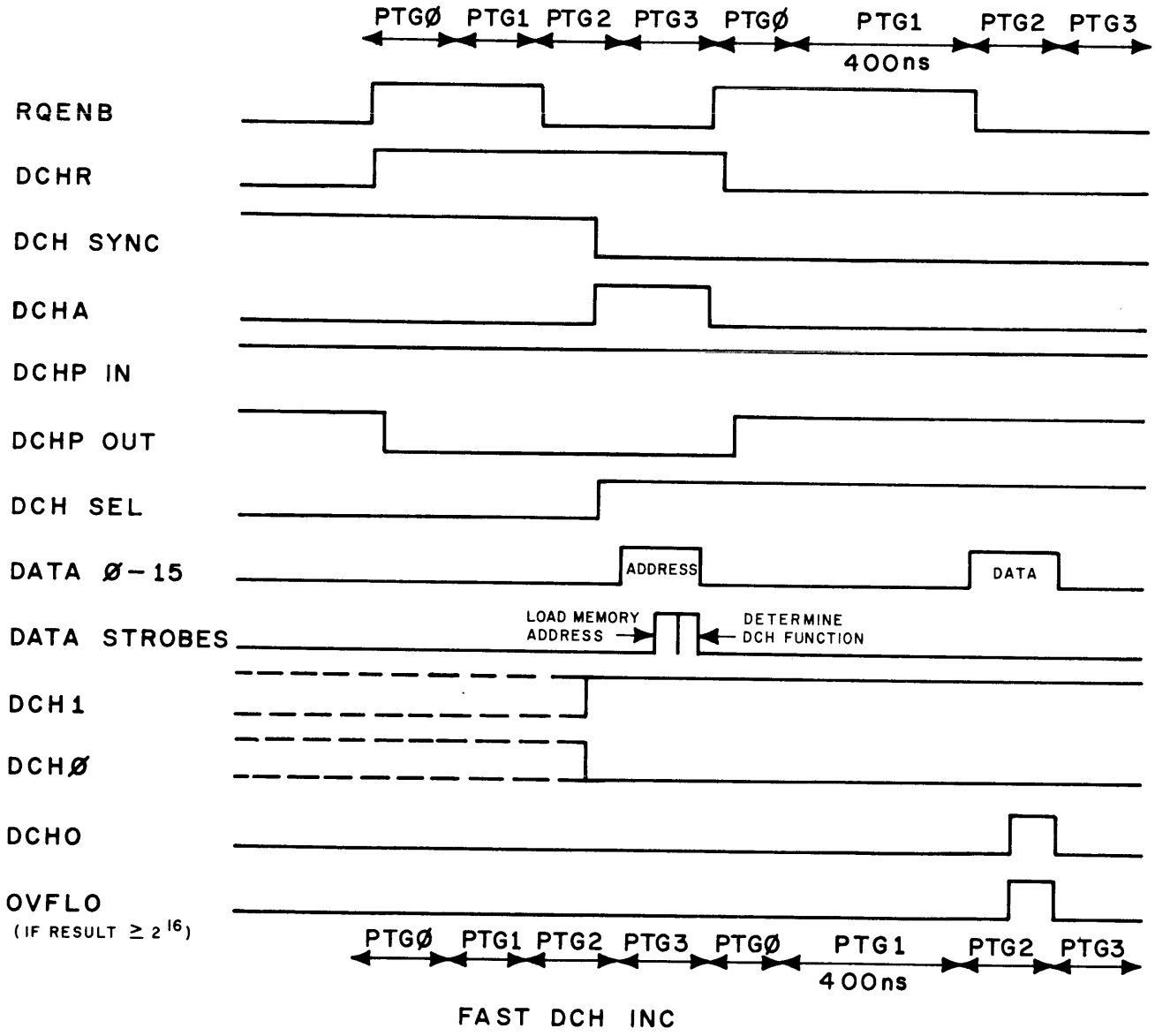
NOTE: All Times typical.
PTG Signals 200 nanoseconds except where indicated.

Figure 4-10. Fast DCH In Timing Diagram



NOTE: All Times typical.
PTG Signals 200 nanoseconds except where indicated.

Figure 4-11. Fast DCH Out Timing Diagram



NOTE: All Times typical.
PTG Signals 200
nanoseconds except
where indicated.

Figure 4-12. Fast DCH INC Timing Diagram

which memory is to be selected. (Reference Drawing 001-000104, (sheets 1-4) bound in Section VII of this manual under separate cover.) Timing signals for memory are generated in the Timing section of CPU-1 and are derived from the MEM CLK timing signals. MA bits 4 through 9 control the Y Driver current logic, while MA bits 10 through 15 control the X Driver current logic. The stack sense lines are differentiated to drive the input gates to the direct set inputs of the MB register. Therefore, with the READ 1 and READ 2 signals present, STROBE A, B, C, and D enable the sense output gates to place the memory data onto the sense lines $\overline{\text{SNS0}}$ through $\overline{\text{SNS15}}$. The configuration on these lines will be unconditionally jam transferred into the MB register. The outputs of the MB drive both the $\overline{\text{DATA0}}$ through $\overline{\text{DATA15}}$ and $\overline{\text{MEM0}}$ through $\overline{\text{MEM15}}$ bus gates. The MB is also loaded (at other times) from its input multiplexer which selects either the I/O Bus $\overline{\text{DATA0}}$ through $\overline{\text{DATA15}}$ lines as an input, or $\overline{\text{MBO0}}$ through $\overline{\text{MBO15}}$ as an input. Either source is gated into the MB by the MB LOAD signal from CPU-1. The MA is loaded from the 15 least significant outputs of the same multiplexer,identified as MD1 through MD15. The MA load control signal, $\overline{\text{MA LOAD}}$ is also produced in CPU-1.

As mentioned previously, the memory must be SELECTED in order to operate. Memory writing requires an INHIBIT signal from the CPU together with a SELECT decode signal. These signals are gated together to enable the INH GATE A, INH GATE B, and WRITE MEM signals to be produced. WRITE MEM turns on the memory current logic, while the INH GATE A and B signals strobe the reset outputs (INH0 through INH15) from the MB register into the inhibit drivers. The presence of inhibit current in any core winding reduces the coercive effect of the normal memory current such that the associated core will not become set. Thus, the data configuration held in the MB will be effectively written into memory via the inhibit drivers.

4-2.11 Power Supply

Reference Drawing 001-000091 bound into Section VII of this manual (under separate cover) when reviewing the descriptions presented in this paragraph. Source power for the power supply may be either standard 117 VAC lines or 220 VAC lines regulated to ±20%, and capable of supplying 325W. Power supply output voltage and currents are as follows:

Voltage	MAX. Current	Primary Use	Remarks
+15v($+V_{mem}$)	9A.	XY & Inhibit drivers in core memory.	short-circuit & over-voltage protected.
+11v($+V_{lamp}$)	2A.	Console lamps	full-wave rectified, non-filtered, non-regulated.
+5v	12A.	IC logic	short-circuit & over-voltage protected.
-5v	1A.	Sense Amplifiers in core memory	Will tolerate ground shorts up to 10 sec.
-15v	2A.	Not used in basic machine. Provided for options.	non-regulated.

The power supply generates four signals which are used by the Processor, $\overline{\text{POWER FAIL}}$ is used by the Power Monitor and Auto-restart options to set a Power Low flag in the Processor causing an interrupt to be generated when the line voltage falls to eighty percent its nominal value. MEM OK goes low when $+V_{mem}$ drops to a point the memory will no longer function reliably. +5 OK goes high when the +5 volt output is approx. 4.4 volts. This edge generates a reset pulse in the processor initializing it at power turn on. 60 Cycle is a sine wave used by the real-time clock. It may be either 50 or 60 Hertz in frequency, depending on the line frequency.

The power supply is composed of five separate, functional parts, +30 VNR generation, +5 volt regulator, +15 volt regulator, -5 volt regulator and associated circuitry, and control signal generation.

4-2.11.1 +30 VNR Generation. +30 VNR is a filtered, non-regulated voltage which is used by the +15 volt and +5 volt regulators. At nominal line voltages, +30 VNR will be between 32 volt and 35.5 volts depending upon the load. The two transformer primaries are wired in parallel for 117 VAC operation, and in series for 220 VAC operation. The fan is always wired in parallel with the BLK-BRN primary, causing it to be effectively wired

into an auto-transformer during 220 VAC operation. A maximum of six amperes may be drawn from the convenience receptacle. The convenience receptacle is switched, and is protected by a 10 ampere fuse. Consequently, improper operation of any device connected to the convenience receptacle may cause power to be lost by the CPU itself.

4-2.11.2 +5 Volt Regulator. A self-oscillating, switching regulator is used to generate the +5 volt output. (A simplified diagram of the regulator appears in Figure 4-13 below.) The operation of this type of regulator is described briefly in the following discussions. A reference voltage is compared with the output voltage. If the output voltage is less than the reference voltage, a series pass transistor is turned on. The transistor drives an LC filter. When the pass transistor turns on, the output voltage of the filter rises linearly until the output voltage equals the reference voltage. At this point the pass transistor is turned off. The field across the inductor now reverses, allowing the inductor to recover through the commutating diode. The output current is now drawn from the energy stored in the LC filter. When the output voltage falls below the reference voltage, the cycle repeats. The output voltage will be sawtooth waveform, centered around the nominal output voltage.

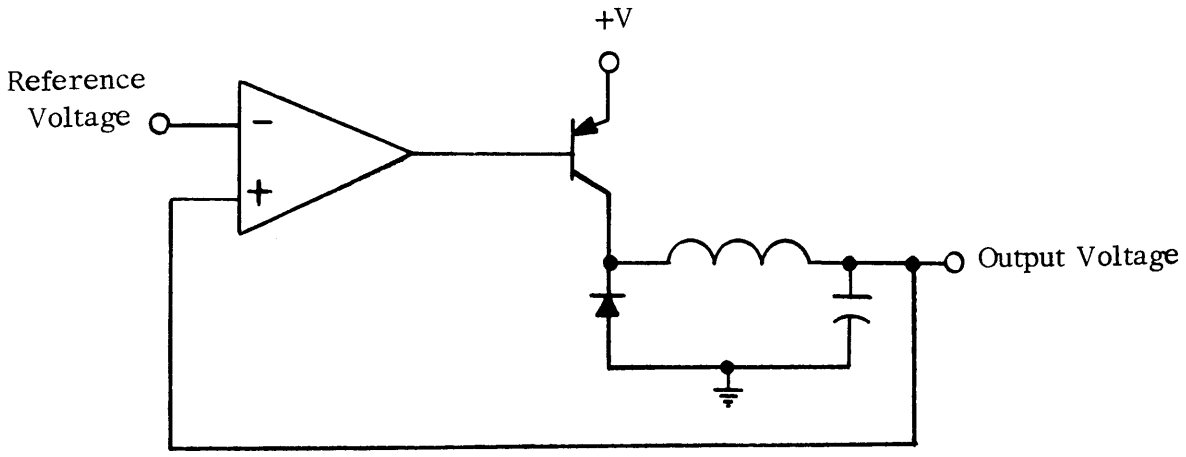


Figure 4-13. +5 Volt Regulator Functional Diagram

The +5 volt regulator consists of chip U2 and its associated circuitry. U2 contains circuits which generate a reference voltage, a voltage comparator, and disable circuits used by the short-circuit protection circuits. The reference voltage at pin 6, nominally 7.15 volts, is divided down to 5 volts and applied to one input of the comparator at pin 5. The output voltage is brought directly to the other side at pin 4. When the output voltage is less than 5.2 volts approximately, pin 11 will drop to + 6 volts, turning both the pre-driver, GE D43C5, and the pass transistor, 2N4399, on. When the output voltage reaches 5.4 volts, the voltage at pin 11 switches to approximately + 30 VNR, turning both transistors off. The difference in switching points is due to the hysteresis added by returning the collector of the 2N4399 through a 220K resistor to pin 5. The frequency of oscillation will vary with load. As the load increases, the frequency increases, reaching a maximum of about 25 KHz at full load.

Short-circuit protection is provided by the circuit to the right of the LC filter. The current flowing through the pass transistor generates a voltage drop across the 0.2 ohm resistor. The voltage is monitored by the 2N4403 transistor, and when the current rises above the level defined to be short-circuit current the 2N4403 turns on, applying a positive level to pin 2 of U2. This forces pin 11 to go high, turning the pass transistor off. This state is maintained by the 2N4400 transistor which turns on once the 2N4403 turns on, and holds the base of the 2N4403 negative with respect to its emitter, even after current has stopped flowing through the pass transistor. The result is that the output voltage drops to zero and stays there, even after the short is removed. To restore power, AC power should be turned off, allowing the 2N4400 and 2N4403 to unlatch, and then turned back on again.

Over-voltage protection is provided by an SCR 2N4441. When + 5 volts rises above approximately 7.5 volts the SCR turns on, blowing the 15 ampere fuse which removes + 30 VNR from the regulator.

4-2.11.3 +15 Volt Regulator. The +15 volt regulator differs from the +5 volt regulator in only two respects. The output voltage, rather than the reference voltage is divided down, before being applied to the voltage comparator. Secondly, the output voltage is reduced with rising

ambient temperature. This causes the memory currents to be reduced with temperature, which is necessary to maintain good margins. At 55 degrees centigrade, the output voltage will drop to +14.4 volts ± 0.1 volts. The short-circuit protection and the over-voltage protection are identical to the + 5 volt circuits. Notice though, that rather than using two separate over-voltage circuits, the two voltages are "OR'ed" into the SCR circuit.

4-2.11.4 -5 Volt Regulator. The -5 volt output is generated using a simple linear regulator. Since the regulator is essentially an emitter follower with no feedback involved, load regulation is marginal. Short-circuit protection is provided by a 4 ohm, 5 watt resistor. Because of the high power dissipation in the resistor during short-circuit conditions, it is only possible to guarantee shorts of ten seconds or less duration. The -15 volt output is taken directly from the rectifiers. It will of course change with the line voltage. With -15 volts and -5 volts under full load, a maximum ripple of .75 volts can be expected on the -15 volt output.

$+V_{lamp}$ is non-filtered and is used only by the console lamps. In order to avoid coupling lamp current into logic ground, a separate ground is provided between the supply and the Console along with $+V_{lamp}$ line.

4-2.11.5 Control Signal Generation. Chip U1 monitors power supply voltages to verify that all voltages are within specified limits. Power for this chip is provided from the +5 volt line. $\overline{\text{POWER FAIL}}$ will go to ground when + 30 VNR is equal to approximately 23 volts. MEM OK drops when + 30 VNR is at 21.5 volts. The time between $\overline{\text{POWER FAIL}}$ dropping and MEM OK dropping is important as this is the time in which the power fail service routine must store the state of the machine before all power is lost. The machine is forced to halt once MEM OK has gone low. The guaranteed worst time between the two signals dropping is 3 milliseconds.

SECTION V

MAINTENANCE

5-1 PERIODIC INSPECTION

The Nova 800 has few mechanically operational components, hence it requires a negligible amount of mechanical inspection. Any lubrication schedules are confined to any mechanically operational peripherals used with the Nova 800, e.g., card punch, line printer, tape punch or reader, etc. The Nova 800 Console switches should be checked periodically for proper operation and switch spring tension. The Console indicators of the various displays should also be observed to detect any burned out indicators. The Static Tests described in paragraph 2-3.2.1 of this manual may be performed to check out the Console controls and indicators.

5-2 PREVENTIVE MAINTENANCE

It should be pointed out that it is impossible to compile a schedule of Preventive Maintenance routines which will satisfy the requirements of all customer applications. The routines outlined in this paragraph may be scheduled against two critical factors: the minimum down-time that can be tolerated by the installation, and the periods of least activity when these procedures may be performed. Obviously these two factors will vary from installation to installation, however, the smaller the minimum down-time becomes, the more frequent preventative maintenance is required, and this must be distributed over the periods of least activity.

IO devices, because of their general mechanical nature, benefit the most from a scheduled preventative maintenance program. In addition a certain percentage of malfunctions can be detected while in the process of occurring. Diagnostic routines should play a major role in preventive maintenance programs. Suggested items that should be included are as follows:

a) Diagnostics - Run exerciser daily for a reliability check of the entire system. All other diagnostics should be run at least once weekly.

b) IO devices - clean daily, removing the dust that normally accumulates as the device is used. Check for excessive vibration, overheating of bearings, and signs of excessive mechanical play or wear. Check punch and teletype for wear and fraying. Empty the punch chad box and remove chad from within the device itself. Clear the type face of the teletype. Look for and remove excess oil and grease from within the devices.

c) General - Check all power and IO cables for fraying or wear. Check all plugs and connectors; tighten if necessary. Check the cooling fan in the Computer power supply for proper operation.

d) Lubrication - Follow the lubrication schedules as set forth in the IO device pamphlets. This requirement is perhaps the most important phase of a preventive maintenance program.

5-3 SPECIAL TOOLS AND TEST EQUIPMENT

The following is a list of special tools and test equipment recommended for efficient maintenance of the Nova 800.

MULTIMETER	SIMPSON MODEL 260 OR EQUIVALENT
OSCILLOSCOPE	TEKTRONIX 453 OR EQUIVALENT
LONG LEAD PROBES	TEKTRONIX P6010-10X OR EQUIVALENT
CURRENT PROBE	TEKTRONIX P6022 OR EQUIVALENT
EXTENDER BOARD	DGC 107-000007-02
WIRE WRAP TOOL (24 GAUGE)	GARDNER DENVER Model 14AX2 OR EQUIVALENT
IC TEST CLIP	MANUFACTURED BY AP INC. Plainesville, Ohio (Part No. 923700)
SOLDERING IRON	WELLER ISOLATED MODEL W-TCP OR EQUIVALENT

5-4 DIAGNOSTIC PROGRAMS

The Nova 800 Diagnostics are individual programs which together test all logical operations of the Computer system. Individually the programs test various logic areas of the Computer and IO. The majority of the diagnostic routines are capable of diagnosing malfunctions down to the logic level. The diagnostics provide a means of measuring the performance of the system on a repeatable basis. Copies of the diagnostic tapes as well as individual program documentation are part of the software package delivered with the Nova 800. Individual program documentation provides information as to operating procedures, error interpretation, console switch settings and logical areas tested. Certain diagnostics are normally part of the daily and weekly preventive maintenance routines.

NOVA 800 DIAGNOSTIC PROGRAMS

PROGRAM	DESCRIPTION
Address Test	Routine to test the memory address section logic.
Checkerboard III	Worst case memory noise test. Program verifies proper operation of sense amps, inhibit drivers, and memory currents.
Nova 800 Logic Test	Gate by gate test of CPU Logic (less IO).
Nova 800 Instruction Timer	Routine to test CPU clock logic, prints instruction times of basic Nova 800 instruction set.
Exerciser	Reliability test - tests CPU logic, TTY reader, punch, high speed paper tape reader, paper tape punch and real time clock. Halts on error.
Arithmetic Test	Exercises the arithmetic and logical instructions of the Nova Computers.

NOVA 800 DIAGNOSTIC PROGRAMS (Continued)

PROGRAM	DESCRIPTION
Nova 800 Teletype Test	Gate by gate test of TTY logic, PI system and IO Bus logic.
Reader/Punch Test	Routine to test high speed paper tape reader and punch.
Real Time Clock Test	Routine to test real time clock logic.
Nova 800/1200 Power Shut Down Test	Test retention of memory data on power loss. Tests power monitor auto restart option.

5-5 TROUBLESHOOTING PHILOSOPHY

Effective troubleshooting is accomplished in a minimum of time by following a series of logical steps. The ultimate aim is to effectively pinpoint the actual problem using all information available. Locating the malfunction is then the next logical step. The following is a suggested plan for effective casualty analysis:

a) Investigation - record the state of the machine on error occurrence. Look for obvious symtoms including operator error, loose plugs or connectors, blown fuses or tripped circuit breaker.

b) Isolation - through the use of diagnostic programs or console trouble-shooting techniques attempt to isolate the malfunction to a particular board.

c) Component Isolation - Isolate the faulty component using an oscilloscope and short diagnostic loops either toggled in at the console or as part of a diagnostic. Selecting the correct external synch is of importance at this point.

d) Replace the faulty component and retest by running the diagnostic that originally failed.

e) Record for future reference, the symptoms, cause, unique trouble-shooting method (or methods) used to isolate the malfunction.

5-5.1 Memory

Address decoding and data word transfer failures are the types of memory malfunctions most frequently encountered. The inability to store or fetch a word from or into a selected core location is usually an indication of the former, while storing or fetching a word which is modified by one or two bits is an indication of the latter. Address test and checkerboard are memory diagnostics designed to verify memory reliability. The two programs will detect and, in most cases, identify the cause of a malfunction. Address test is primarily intended to test address selection logic and verifies the ability to uniquely address all core locations. Checkerboard is a worst case noise test designed to detect the picking up or dropping of bits in a data word transfer. In the case of intermittent failures it may be desirable to revert to console troubleshooting, utilizing short closed loop routines which are toggled in. Programs such as the one illustrated below are valuable in resolving failures.

SAMPLE DIAGNOSTIC LOOP PROCEDURE

1. Deposit data word in AC2
2. Deposit program in core
3. Start - Program halts - Load address in console switches and continue

```
LOC
0000    063077    DOC 0, CPU      ; Halt Inst.
0001    060477    DIA 1, CPU      ; Reads Switches
0002    044011    STA 1, 11       ; Store Addr
0003    052011    STA 2, @ 11     ; Data to Addr
0004    000001    JMP. -3         ; Loop
```

Note: The address can be varied by changing the contents of the Console switches. The above routine will store the contents of AC2 (Data word) into the address in AC1. It is useful in monitoring Read/Write currents and the Inhibit current. Current loops are provided on the memory assembly to facilitate the use of a suitable current probe (Tektronix P6022 or equivalent) for current measurements. However, only one current loop is provided for measuring the Inhibit current. The physical locations of the Read/Write and Inhibit

Current loops on the memory assembly may be referenced in Section VI of this manual, the Illustrated Parts List for the Nova 800 (under separate cover). The Nova 800 power supply is considered a factory repairable unit only, and if the memory currents measured differ extensively from the values listed in Table 1-2 notify Data General Field Service.

Maintenance other than lubrication, minor adjustments and part changes should be performed by DGC personnel or respective manufacturer representatives. Lubrication should be performed in accordance with the appropriate manual listed below:

Applicable Manuals

Teletype	-	Technical Manual 33 Teletype Writer sets Bulletin 310B Volume I
		Technical Manual 33 Teletype Writer sets Bulletin 310B Volume II
		33 Page Printer set ASR, KSR and RO Parts Bulletin 1184B
High Speed Punch	-	Technical Manual High Speed Tape Punch set (BRPE) Bulletin 215B
		High Speed Tape Punch set (BRPE) Parts Bulletin 1154B
High Speed Reader	-	(Digitronics Model) (2540EP)
		Perforated Tape Reader Operation and Maintenance Manual

Lubricating Materials

Teletype-Keyboard	KS7470 (oil) KS7471 (grease)
Typing unit	KS7470 (oil) KS7471 (grease)
Reader	KS7470 (oil) KS7471 (grease) Lupriplate 105
High Speed Punch -	(BRPF11) KS7470 (oil) 145867 (grease)
High Speed Reader -	SAELO (oil)

Recommended spares one each

High Speed Reader -	Lamp incandescent Digitronics TLNBF009 GE (08805) (P/N 1638)
High Speed Punch -	Drivebelt #135097

5-6 COMPONENT REPLACEMENT

The replacement of a component requires care to prevent damage to circuit board etch. Clipping a component from the circuit board rather than unsoldering is the preferred method. Excessive heat from a soldering iron may result in damage to the component being replaced. The use of a soldering iron with an isolation transformer, a small copper alligator clip as a heat sink and a delay between the soldering of individual pins of a chip are recommended. When the extender board is used, the weight of the board under test should be supported by a non-conductive material. Replacing a Console Switch or Indicator requires the removal of the Console subassembly. The following is the procedure to be followed when replacing a Console (Data) switch:

1. Remove the four 6/32 nuts attaching the Console subassembly to the enclosure frame.

2. Remove the eight 2/56 screws holding the circuit board assembly to the console casting.

3. Replace the defective switch and reassemble in reverse order.

To replace a Console indicator follow steps 1 & 2 above and in addition remove the three phillips panhead screws holding the Benelex to the circuit board. Replace and reassemble in reverse order.

5-7 IC IDENTIFICATION

Table 5-1 is included in this section to facilitate any troubleshooting procedures that require identification between any Nova 800 IC reference number (U1-Un) and the original manufacturer's part number. This list may be referenced as a bridge between the logic configurations shown on the various CPU and Memory logic diagrams and the IC (package) pin definitions summarized in Appendix A of this manual.

Table 5-1. IC Identification List

CHIP	CPU 1	CPU 2	MEMORY
U1	9601***	MUL/DVD OP*	MC3062
U2	9002***	MUL/DVD OP*	7438
U3	7474***	MUL/DVD OP*	8840/9005
U4	9005	MUL/DVD OP*	7438
U5	9009	MUL/DVD OP*	8840/9005
U6	9002	MUL/DVD OP*	7438
U7	3026	MUL/DVD OP*	8840/9005
U8	3026	MUL/DVD OP*	7438
U9	8H90	MUL/DVD OP*	8840/9005
U10	7475	8H90	7438
U11	7475	9002	8840/9005
U12	9009	9005	7438
U13	7475	9003	8840/9005
U14	7475	3026	7438
U15	9009	3061	8840/9005
U16	3061	8885	7438
U17	3061	7475	8840/9005
U18	3026	3026	8H90
U19	9009	8H90	MC3026
U20	3061	3026	75451
U21	9005	MUL/DVD OP*	8828/7474
U22	9003	MUL/DVD OP*	7475
U23	9003	MUL/DVD OP*	8828/7474
U24	8H90	MUL/DVD OP*	8828/7474
U25	3026	MUL/DVD OP*	7475
U26	3026	MUL/DVD OP*	MC3026
U27	3026	MUL/DVD OP*	8828/7474
U28	9003	MUL/DVD OP*	8828/7474
U29	9008	MUL/DVD OP*	7475
U30	7438	7488**	7437
U31	9009	7488**	8828/7474

Table 5-1. IC Identification List (Continued)

CHIP	CPU 1	CPU 2	MEMORY
U32	9002	8H90	8828/7474
U33	9009	7438	7475
U34	3061	9002	8828/7474
U35	9002	7438	9007
U36	8885	8271	MC3026
U37	3061	9005	8889/9002
U38	8H90	9002	7524
U39	8271	9005	8889/9002
U40	8234	9008	7524
U41	8885	7475	MC3026
U42	9002	8271	75451
U43	8234	MUL/DVD OP*	75451
U44	9008	74161/9316	8H90
U45	8234	8234	8889/9002
U46	8234	7475	7524
U47	9002	8271	75324
U48	8H90	MUL/DVD OP*	8889/9002
U49	3026	74161/9316	7524
U50	9009	8234	75324
U51	8885	7475	75451
U52	9009	8271	75324
U53	9005	MUL/DVD OP*	75451
U54	8271	74161/9316	75324
U55	3061	8234	8889/9002
U56	9002	7475	7524
U57	8885	8271	75324
U58	9003	MUL/DVD OP*	8889/9002
U59	8H90	74161/9316	7524
U60	7438	8234	75324
U61	7438	8264	75451
U62	8885	74170	75324
U63	7438	74170	75451
U64	8885	8264	8889/9002
U65	8885	8264	7524
U66	8H90	74170	75324
U67	9005	74170	8H90
U68	8271	8264	8889/9002
U69	8271	8264	7524
U70	8885	74170	75451

Table 5-1. IC Identification List (Continued)

CHIP	CPU 1	CPU 2	MEMORY
U71	9003	74170	8H90
U72	74H55	8264	75324
U73	9003	8264	75324
U74	9009	74170	75324
U75	8885	74170	75324
U76	9005	8264	75324
U77	8H90	7438	75324
U78	8271	9002	75324
U79	9009	7438	75324
U80	9008	9002	8H90
U81	9002	7438	
U82	7474	9002	
U83	9008	7438	
U84	8H90	9002	
U85	9008	9008	
U86	9005	9008	
U87	9002	9008	
U88	9003	9008	
U89	9002	8260	
U90	8885	9008	
U91	9003	9008	
U92	9005	9008	
U93	9008	9008	
U94	8885	8260	
U95	9002	9008	
U96	9008	9008	
U97	9002	9008	
U98	8885	9008	
U99	9005	8260	
U100	8H90	9008	
U101	74H55	9008	
U102	3061	9008	
U103	74H55	9008	
U104	74H55	8260	
U105	9002		
U106	9003		
U107	9008		
U108	7438		
U109	7438		
U110	8885		

*** Power Monitor Option
** Program Load Option
* Multiply/Divide Option

5-8 POWER SUPPLY SPECIFICATION

The Nova 800 power supply provides all of the power required to operate the Nova 800 Processor logic and memory circuits. The electrical specification for the power supply voltages are listed in Table 1-2 and are not repeated here. Each side of the ac power line is fused by a 10 amp, 250 volt glass tube fuse (bus type). Each fuse is mounted in individual panel mounted fuseholders, with the two fuseholders mounted just above the convenience outlet in the rear of the power supply. The fuses should be checked first following any interruption of output power. Maintenance or repair beyond this point should be performed at the factory.

5-9 NOVA 800 INTERNAL INTERCONNECTIONS

As shown in Figure 1-3, the Nova 800 major assembly components are interconnectors which electrically join the PCB assemblies in each component together in the final assembly.

AC power is connected into the power supply fuses via the power cord. The load side of the fuses are connected to J1 (Reference Drawing #001-000091). P1 connects into J1 and electrically incorporates the Power switch (at the Console) in series with the convenience outlet and the power transformer primary.

The PCB assembly of the Power supply terminates in a 52 pin edge type connector P2, which connects into J2 on the Resistor Board subassembly. (Reference Drawing #001-000087). This subassembly also terminates in 52 pin edge type connector P3, which connects into J3 of the Back Panel (of the Multiple Printed Circuit Board Connector).

The RINH signals, power monitor signals, and voltages connected into the Back Panel are

routed to the various 15 inch PCB Assemblies, e.g., CPU, Memory, I/O boards by the Back Panel etch (Reference Drawing #001-000094 for the Back Panel signal distribution.)

Power and Console signals are interconnected between the Back Panel and the Console by etched edge connectors P4 (of the Back Panel) and receptacle connector J4 mounted on the Console PCB Assembly. Table 5-2 lists the various signals routed through P4 & J4 and their respective Back Panel terminations. The electrical terminations from connector J4 are shown on the Nova 800 and 1200 Console drawing #001-000089.

Table 5-2 Console/Back Panel Connections

Connector J4/P4 PIN	SIGNAL	Back Panel PIN	Connector J4/P4 PIN	SIGNAL	Back Panel PIN
1	GND	B1	27	+5	B4
2	MEM15	B18	28	MBO15	A41
3	MEM14	B76	29	MEM13	A35
4	MBO13	A37	30	MBO12	A39
5	MEM12	A36	31	MEM11	A51
6	MBO11	B5	32	MEM10	A45
7	MEM9	A53	33	LIGHTS	N/A (Bus to Pwr Supply)
8	MBO9	B9	34	MEM8	A55
9	MBO7	B14	35	MBO6	B16
10	MEM6	B22	36	MEM5	B26
11	MBO5	B32	37	MEM4	B28
12	MBO14	A43	38	MBO3	B43
13	MEM2	B47	39	MEM0	B71
14	MBO1	B77	40	LIGHT	GND
15	MBO2	B44	41	MEM1	B70
16	MBO4	B42	42	MEM7	B24
17	GND	B2	43	MEM3	B68
18	MBO8	B12	44	MBO10	B8
19	Restart Enable	A32	45	STOP	A31
20	RST	A30	46	CONT DATA	A28
21	CON RQ	A27	47	Cont+Istp+Mstp	A25
22	CON INST	A22	48	MSTP	A20
23	PL	A19	49	CARRY	A15
24	ISTP	A17	50	FETCH	A13
25	ION	A16	51	EXEC	A11
26	RUN	A14	52	DEFER	A12

SECTION VI

ILLUSTRATED PARTS LIST

The Illustrated Parts List for the Nova 800 Central Processor, Nova 800 4K Memory, and the Basic I/O Control are published under composite Document No. 005-000631-00. For illustrations and descriptions of the parts complement for the equipment mentioned above reference this separate publication.

SECTION VII

REFERENCE DRAWINGS

7-1 INTRODUCTION

All of the Data General electrical reference drawings for the Nova 800 Central Processor, Nova 800 4K Memory, and the Basic I/O Control are contained in this section. The actual drawing complement is bound under separate cover, however all of the drawings and wire lists comprising this section are listed in Table 7-1 for reference purposes.

Table 7-1. Reference Drawings

Title	Drawing No.
Nova 800 Timing and Flow Chart (Sheet 1-Flow)	001-000121 (Sheet 1)
Nova 800 Timing and Flow Chart (Sheet 2-Memory/IO Timing)	001-000121 (Sheet 2)
Nova 800 Timing and Flow Chart (Sheet 3-DCH Timing)	001-000121 (Sheet 3)
Nova 800 & 1200 Console	001-000089
Nova 800 Timing CPU-1	001-000092 (Sheet 1)
Nova 800 IR and States	001-000092 (Sheet 2)
Nova 800 ALU Control	001-000092 (Sheet 3)
Nova 800 IO	001-000092 (Sheet 4)
Nova 800 CRY CPU-2	001-000093 (Sheet 1)
Nova 800 ALU Bits 0-7	001-000093 (Sheet 2)
Nova 800 ALU Bits 8-15	001-000093 (Sheet 3)
4K Memory-MA & MB Register & Control	001-000104 (Sheet 1)
4K Memory-Sense & Inhibit	001-000104 (Sheet 2)
4K Memory-X Drivers	001-000104 (Sheet 3)
4K Memory-Y Drivers	001-000104 (Sheet 4)
Nova 800, 1200 Power Supply	001-000091
Nova 800 Back Panel	001-000094
Nova 800 or 1200 Rack Installation	010-000006

Table 7-1. Reference Drawings (Continued)

Title	Drawing No.
Power Monitor*	001-000111
Resistor Board	001-000087
I/O Bus Receivers & Common Select	001-000070
Teletype Control*	001-000071
Paper Tape Reader Control 4011*	001-000072
Paper Tape Punch Control 4012*	001-000073
Real Time Clock*	001-000074
I/O External Cable	008-000044
Nova 800/1200 Internal I/O Cable Wire List	008-000053
Nova 800/1200 Hi Speed* Reader Internal Cable Wire List	008-000054
Nova 800/1200 Punch* Internal Cable Wire List	008-000055

*Option drawings are supplied only with equipment which includes that particular option.

APPENDIX A

LOGIC DIAGRAMS

AND

TRUTH TABLES

FOR

NOVA 800 INTEGRATED

CIRCUIT PACKAGES

INTRODUCTION

This Appendix is a compilation of the Logic Diagrams and Truth Tables for the Integrated Circuit (IC) Packages used in the Nova 800 logic. Information presented in this section is intended to supplement the electrical (assembly logic diagrams) drawings and the Integrated Circuits listed in Section V of this manual. The data presented herein is cataloged first alphabetically then by number, where the alphabetical prefixes of the number are germane to the manufacturer's identification of the part rather than defining operational parameters (e.g., temperature, case construction, etc.). Table A-1 is an index listing the types of IC's cataloged in this section with the corresponding page number location. All of the logical elements listed use positive logic, i.e., the highest voltage equals a logic 1.

Table A-1. IC INDEX

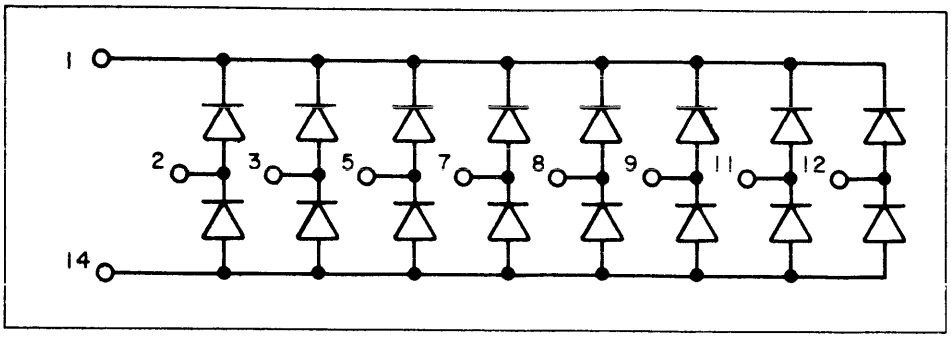
Table A-1. IC INDEX (Continued)

Table A-1. IC INDEX (Continued)

BC728

16 Diode Array

LOGIC DIAGRAM

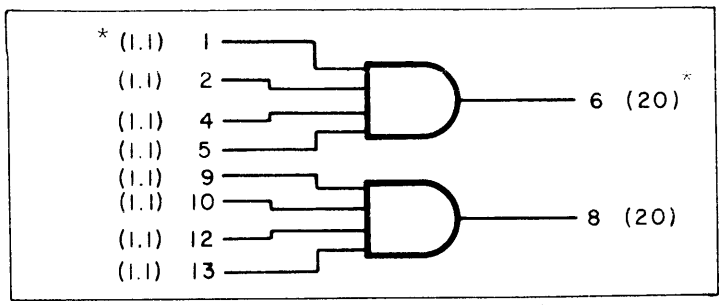


TRUTH TABLE N/A For BC728

MC3026

Dual 4-Input AND Power Gate

LOGIC DIAGRAM/PIN DESIGNATIONS



*Loading Max. Shown in Parenthesis

t_{pd} = 9.0 ns typ

P_D = 90 mW typ/pkg

TRUTH TABLE

$6 = 1 \cdot 2 \cdot 4 \cdot 5$

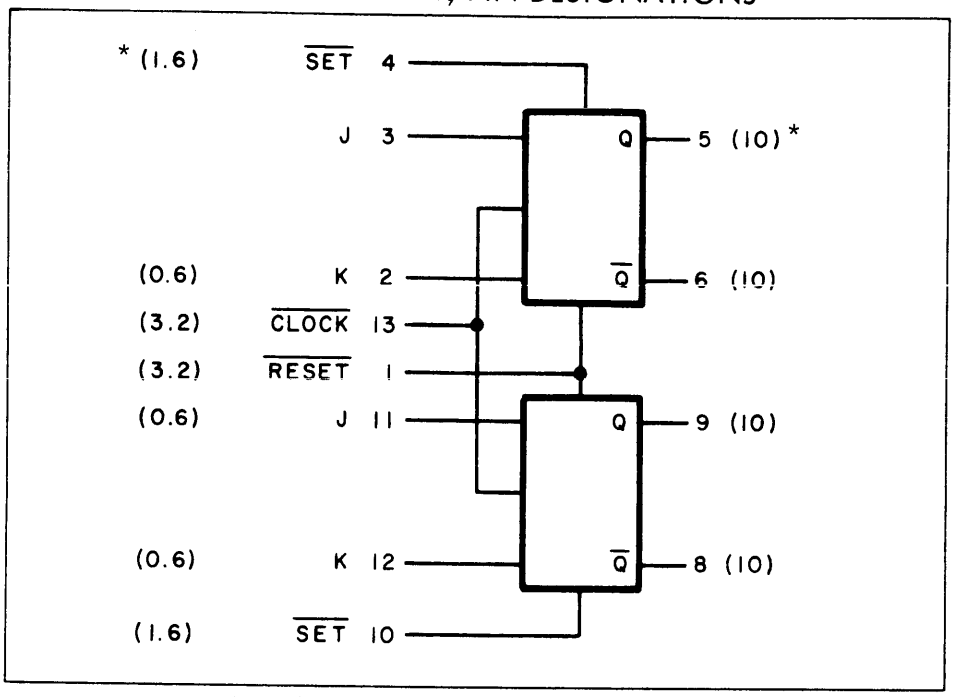
$8 = 9 \cdot 10 \cdot 12 \cdot 13$

V_{CC} = Pin 14, GND = Pin 7

MC3061

Dual J-K Flip-Flop

LOGIC DIAGRAM/PIN DESIGNATIONS



*Loading Max. Shown in Parenthesis

t = 50 MHz

P_D = 100 mW typ/pkg

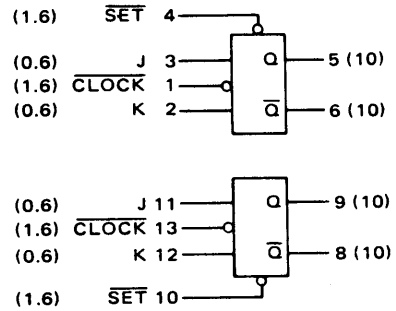
TRUTH TABLE

J	K	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

V_{CC} = Pin 14, GND = Pin 7

MC3062
Dual J-K Flip-Flop

J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

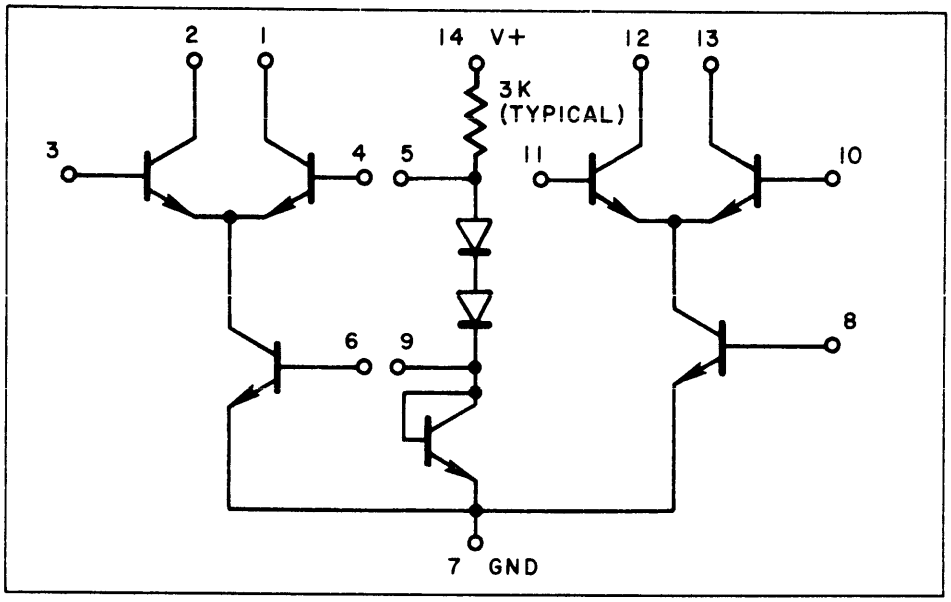


f = 50 MHz
P_D = 100 mW typ/pkg

NE510A

Amplifier

BASIC CIRCUIT SCHEMATIC



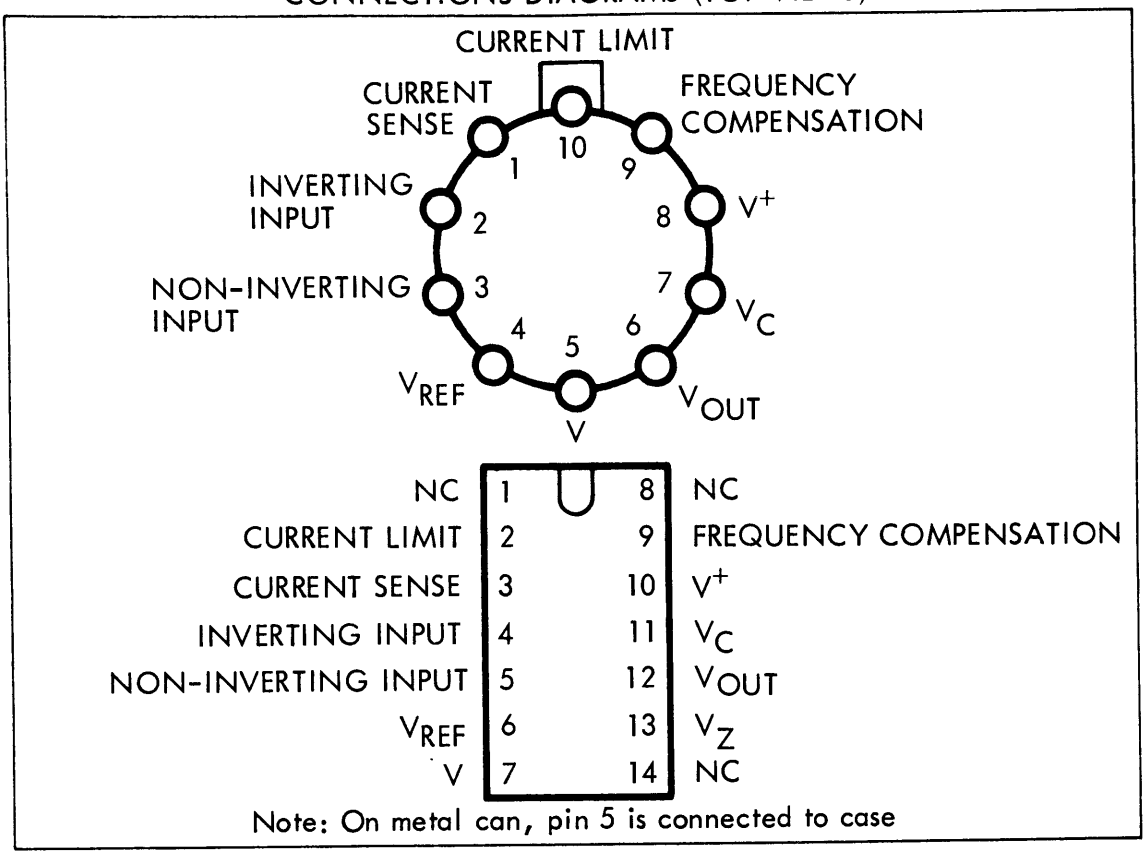
TRUTH TABLE N/A For NE 510A

NOTE: NE510 may be connected as either a high-gain, common-emitter, common-base, cascode amplifier or a common-collector, common-base, differential amplifier that is useful in critical limiter applications. Automatic gain control may be applied to either circuit.

μA723C

Precision Voltage Regulator

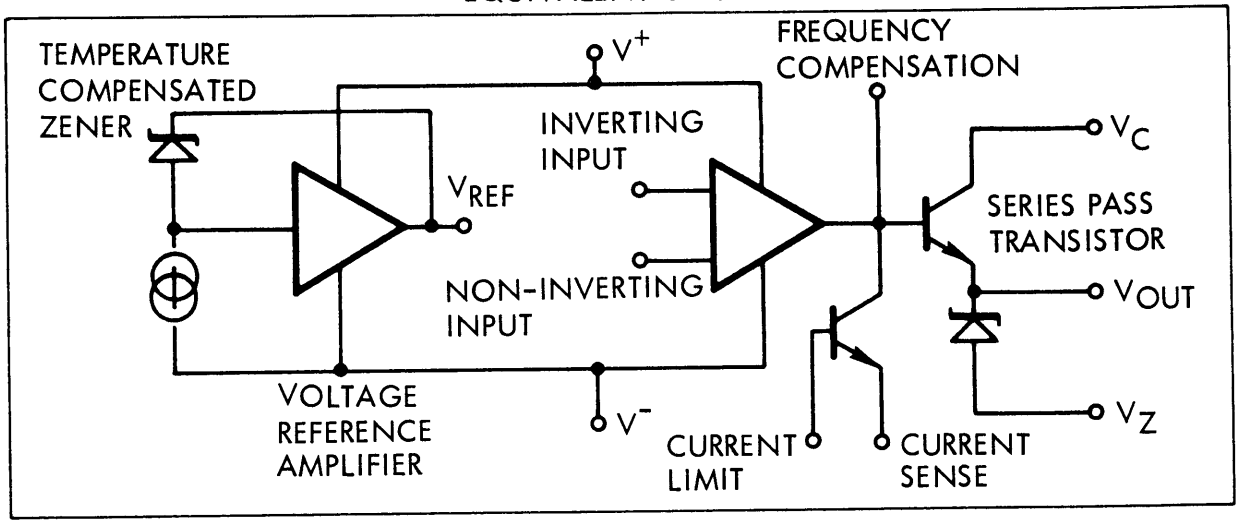
CONNECTIONS DIAGRAMS (TOP VIEWS)



NC	1	8	NC
CURRENT LIMIT	2	9	FREQUENCY COMPENSATION
CURRENT SENSE	3	10	V^+
INVERTING INPUT	4	11	V_C
NON-INVERTING INPUT	5	12	V_{OUT}
V_{REF}	6	13	V_Z
V	7	14	NC

Note: On metal can, pin 5 is connected to case

EQUIVALENT CIRCUIT

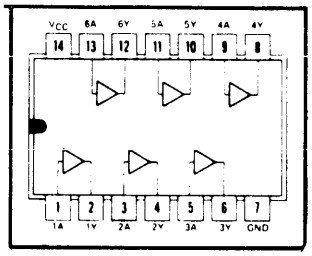


TRUTH TABLE N/A For μA723C

7407

HEX BUFFERS/DRIVERS WITH OPEN-COLLECTOR HIGH-VOLTAGE OUTPUTS

J OR N
DUAL-IN-LINE PACKAGE
(TOP VIEW)



positive logic: $Y = \overline{A}$

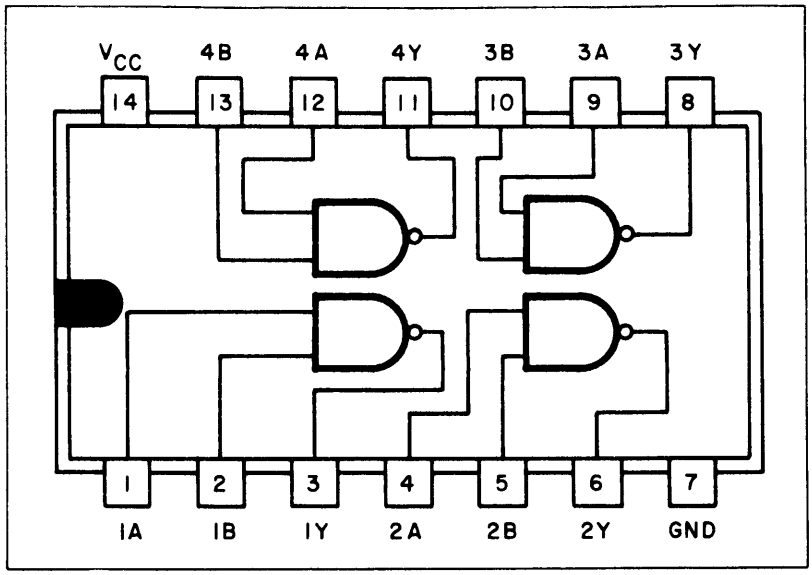
These monolithic TTL hex buffers/drivers feature high-voltage open-collector outputs for interfacing with high-level circuits (such as MOS), or for driving high-current loads (such as lamps or relays), and are also characterized for use as buffers for driving TTL inputs. For increased fan-out, several buffers in a single package may be paralleled. The SN5407 and SN7407 have minimum breakdown voltages of 30 volts and the SN5417 and SN7417 have minimum breakdown voltages of 15 volts. The maximum sink current is 30 milliamperes for the SN5407 and SN5417, and 40 milliamperes for the SN7407 and SN7417.

These circuits are completely compatible with most TTL and DTL families. Inputs are diode-clamped to minimize transmission-line effects which simplifies design. Typical power dissipation is 145 milliwatts and average propagation delay time is 14 nanoseconds.

7437
7438

Quadruple 2 - Input Positive NAND Gates
(With Open - Collector Outputs)

LOGIC DIAGRAM/PIN DESIGNATIONS



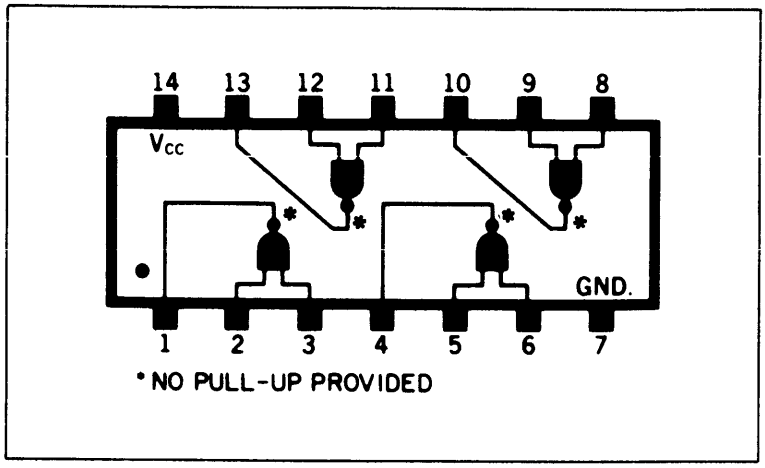
TRUTH TABLE

(positive logic) $Y = \overline{AB}$

7439 / 8881

Quad 2 - Input NAND Gate

LOGIC DIAGRAM



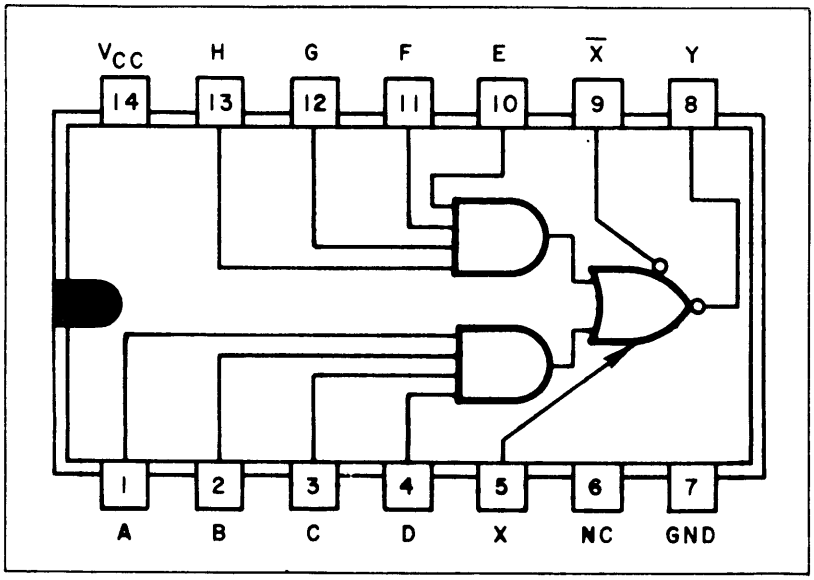
TRUTH TABLE

V_{IN}	V_{IN}	V_{OUT}
L	L	H
L	H	H
H	L	H
H	H	L

74H55

Expandable 4-Input AND-OR-INVERT Gates

LOGIC DIAGRAM



Both expander inputs are used simultaneously for expanding.
If expander is not used leave X and $\overline{X}$ pins open.

TRUTH TABLE

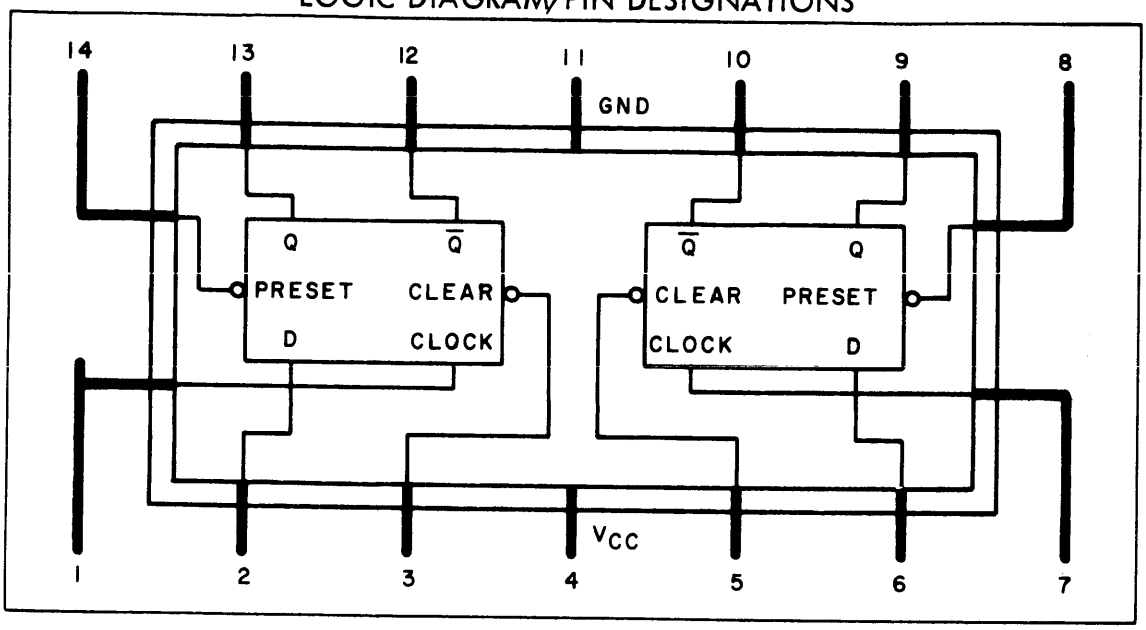
(positive logic) $Y = (ABCD) + (EFGH) + (X)$

X = Output of SN54H60/SN74H60 or SN54H62/SN74H62 circuit.

7474

Dual D-Type edge-triggered flip-flop

LOGIC DIAGRAM/PIN DESIGNATIONS



Propogation delay - 24 nsec

Power dissipation - 84 mW total for two flip-flops (42 mW per flip-flop)

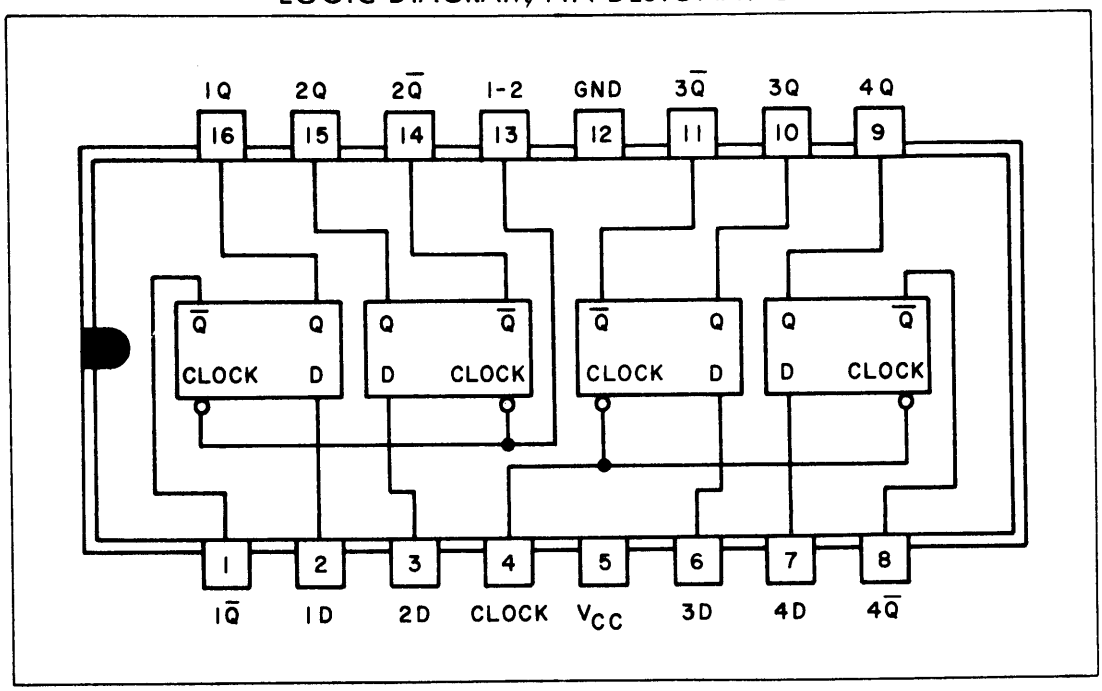
TRUTH TABLE

t_n	$t_n + 1$			
D	Q	$\overline{Q}$	PRESET	CLEAR
0 1	0 1	1 0		
	1	0	0	
	0	1		0

7475

4 - Bit Bistable Latch

LOGIC DIAGRAM/PIN DESIGNATIONS



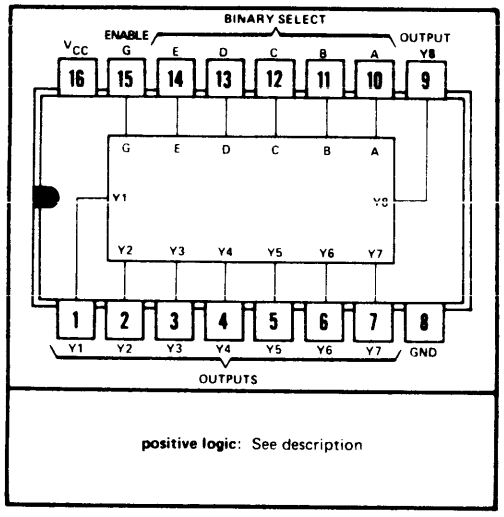
TRUTH TABLE
(EACH LATCH)

t_n	t_{n+1}
D	Q
1	1
0	0

7488

256-Bit Read-Only Memory

LOGIC DIAGRAM/PIN DESIGNATIONS

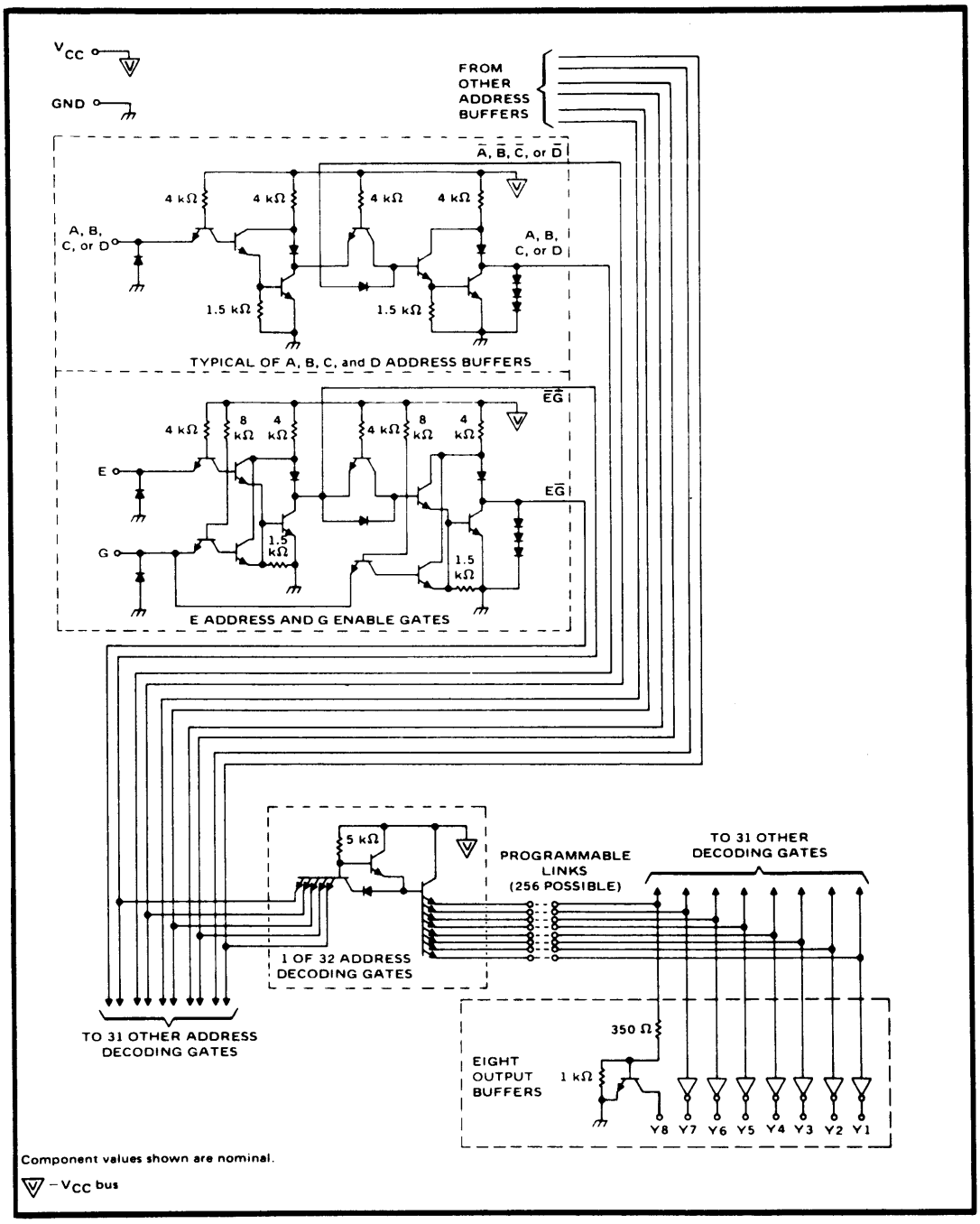


The SN7488 circuit is a custom-programmed, 256-bit, read-only memory organized as 32 words of eight bits each. This monolithic, high-speed, transistor-transistor logic (TTL), 32-word memory array is addressed in stright 5-bit binary with full on-chip decoding. An overriding memory-enable input is provided which, when taken high, will inhibit the 32 address gates and cause all eight outputs to remain high. Data, as specified by the customer are permanently programmed into the monolithic structure for the 256 bit locations. The address of an eight-bit word is accomplished through the buffered, binary select inputs which are decoded by the 32 five-input address gates. When the memory-enable input is high, all 32 gate outputs are low, turning off the eight output buffers.

7488 (cont.)

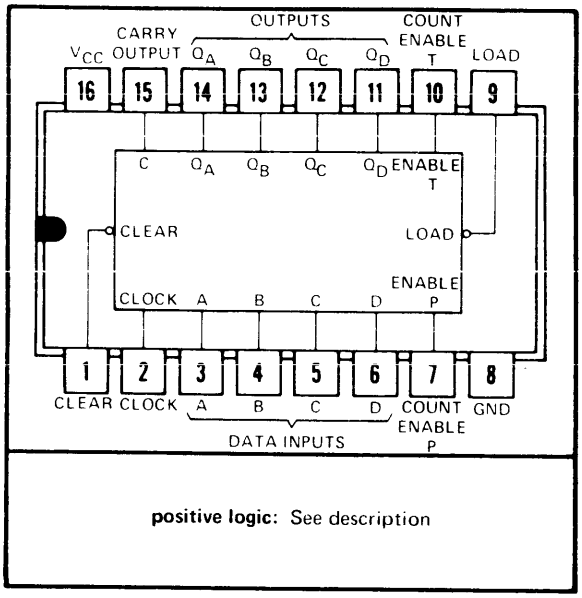
256-Bit Read-Only Memory

SIMPLIFIED SCHEMATIC DIAGRAM



SN74161
SYNCHRONOUS COUNTER

J OR N
DUAL-IN-LINE PACKAGE (TOP VIEW)

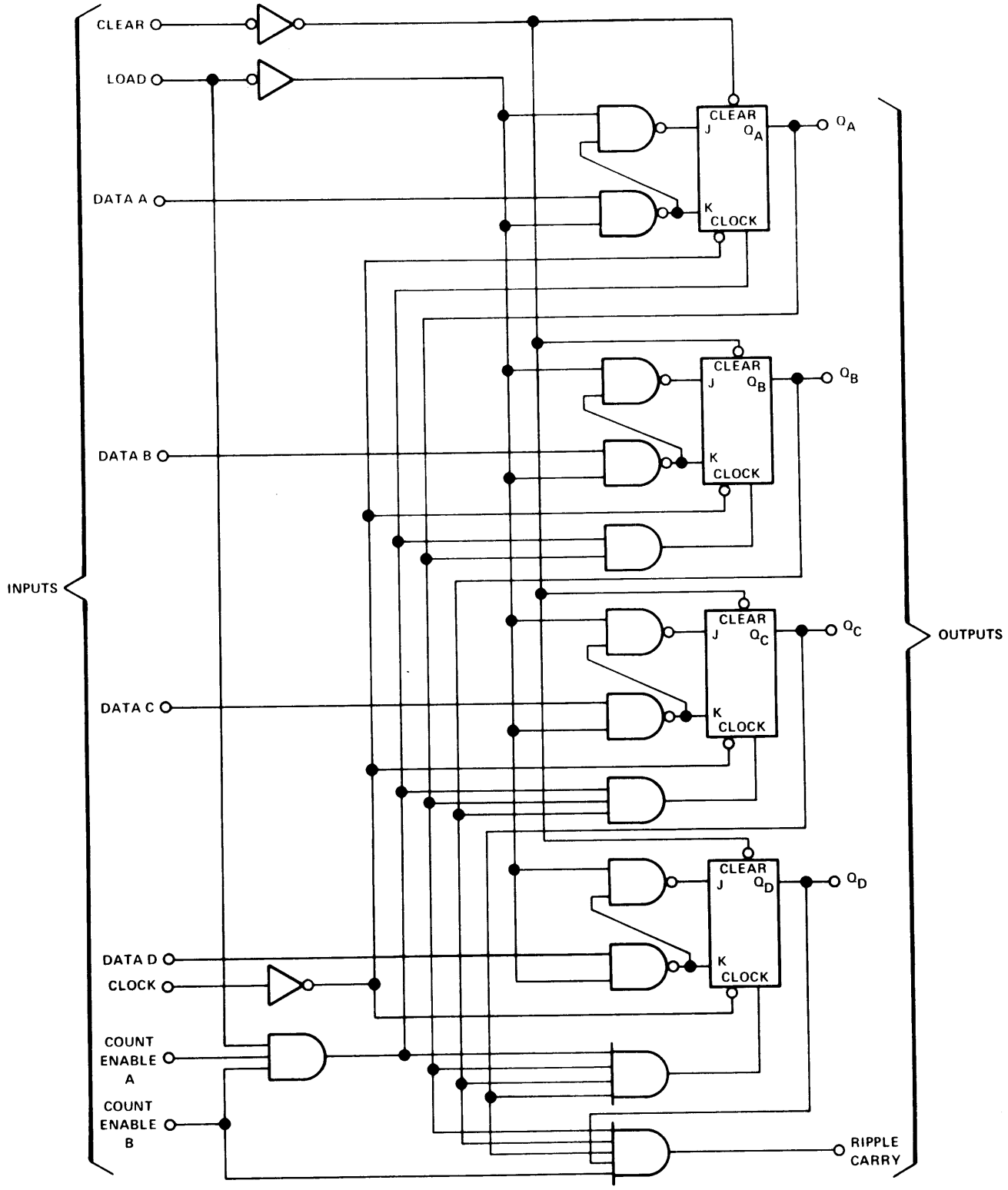


positive logic: See description

The SN74161 is a synchronous, presettable counter which has a direct clear and utilizes an internal carry look-ahead counting technique. The SN74161 is a 4-bit binary counter. Synchronous operation is provided by having all flip-flops clocked simultaneously so that the outputs change coincident with each other when so instructed by the count-enable inputs and internal gating. This mode of operation eliminates the output counting spikes which are normally associated with asynchronous (ripple clock) counters. A buffered clock input triggers the four J-K master-slave flip-flops on the rising (positive-going) edge of the clock input waveform. The counter is fully programmable; that is, the outputs may be preset to either state. As presetting is synchronous, placing a low level on the load input disables the counter and causes the outputs to agree with the data inputs after the next clock pulse. The clear function is asynchronous and a low level at this input sets all four of the flip-flop outputs low regardless of the state of the clock.

The carry look-ahead circuitry provides for cascading counters for n-bit synchronous applications without additional gating. Instrumental in accomplishing this function are two count-enable inputs and a carry output. Both count-enable inputs (P and T) must be high to count, and input T is fed forward to enable the carry output. The carry output thus enabled will produce a positive output pulse with a duration approximately equal to the positive portion of the Q_A output. This positive overflow carry pulse can be used to enable successive cascaded stages.

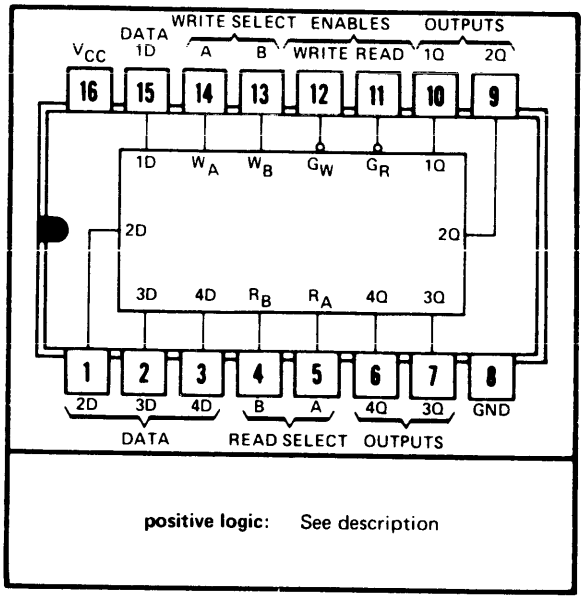
functional block diagram



74170

4-By-4 Register Files

LOGIC DIAGRAM

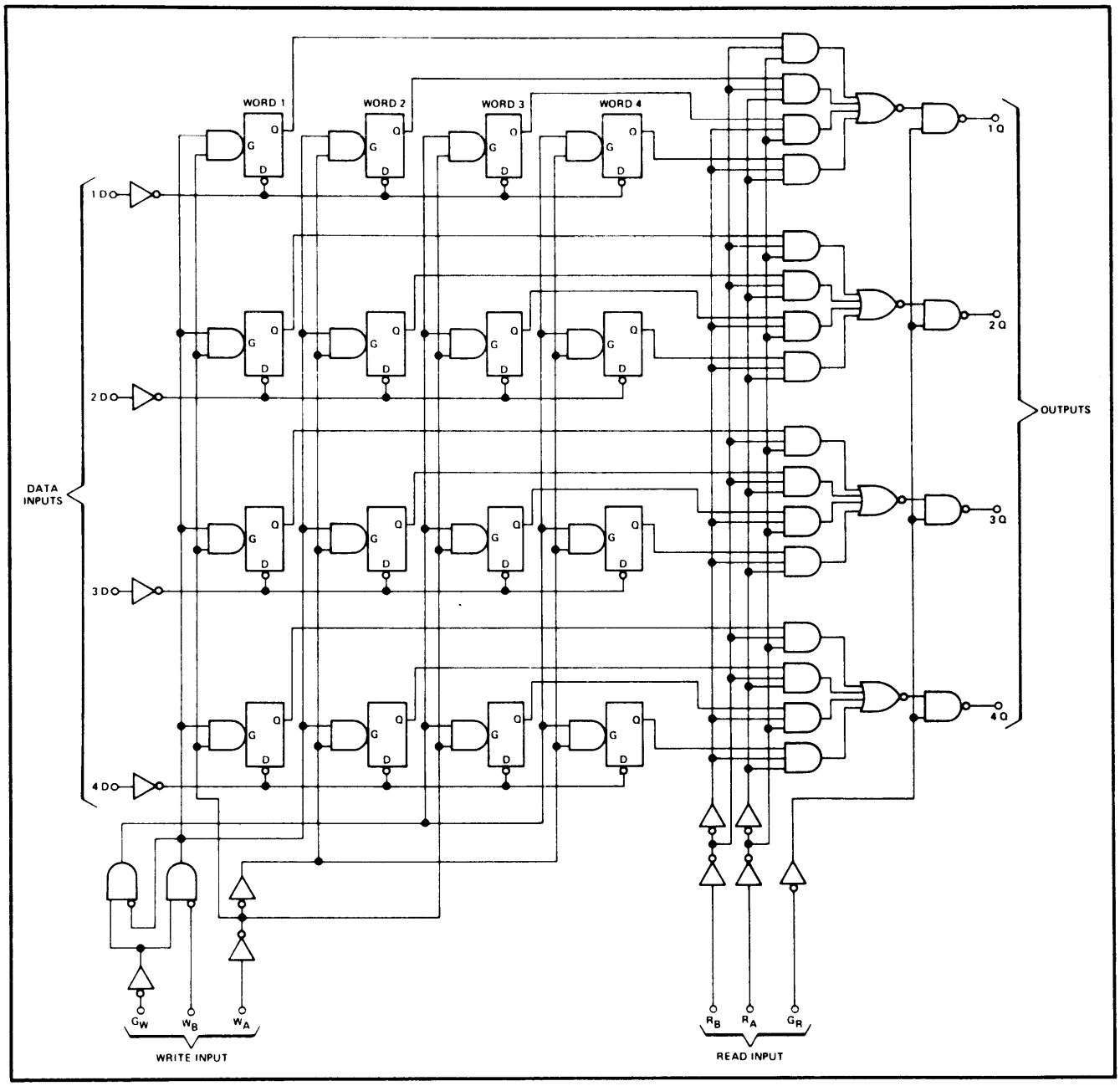


Four data inputs are available which are used to supply the 4-bit word to be stored. Location of the word is determined by the write address inputs A and B in conjunction with a write-enable signal. Data applied at the inputs should be in its true form. That is, if a high-level signal is desired from the output, a high-level is applied at the data input for that particular bit location. The latch inputs are arranged so that new data will be accepted only if both internal address (T) gate inputs are high. When this condition exists, data at the D input is transferred to the latch output. The individual address lines permit direct acquisition of data stored in any four of the latches. Four individual decoding gates are used to complete the address for teading a word. When the read address is made in conjunction with the read-enable signal, the word appears at the four outputs.

74170 (cont.)

4-By-4 Register Files

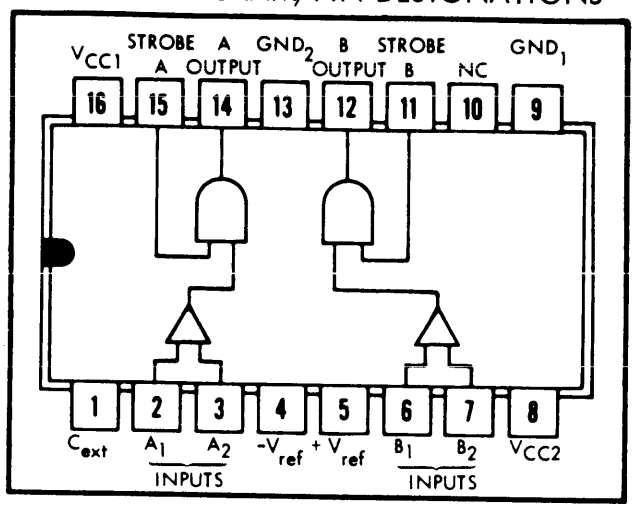
FUNCTIONAL LOGIC DIAGRAM



7524

Dual Sense Amplifiers

LOGIC DIAGRAM/PIN DESIGNATIONS



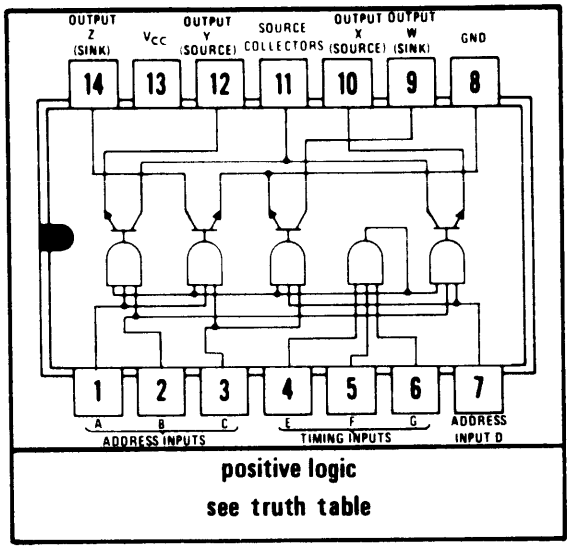
TRUTH TABLE

$IN_A \cdot$ STROBE A = OUT A
$\overline{IN_A} \cdot$ STROBE A = $\overline{\text{OUT A}}$
$IN_B \cdot$ STROBE B = OUT B
$\overline{IN_B} \cdot$ STROBE B = $\overline{\text{OUT B}}$

75324

Memory Driver with Decode Inputs (400 M A)

LOGIC DIAGRAM / PIN DESIGNATIONS



TRUTH TABLE

INPUTS							OUTPUTS			
ADDRESS				TIMING			SINK	SOURCES		SINK
A	B	C	D	E	F	G	W	X	Y	Z
0	0	1	1	1	1	1	ON	OFF	OFF	OFF
0	1	0	1	1	1	1	OFF	ON	OFF	OFF
1	1	0	0	1	1	1	OFF	OFF	ON	OFF
1	0	1	0	1	1	1	OFF	OFF	OFF	ON
X	X	X	X	0	X	X	OFF	OFF	OFF	OFF
X	X	X	X	X	0	X	OFF	OFF	OFF	OFF
X	X	X	X	X	X	0	OFF	OFF	OFF	OFF

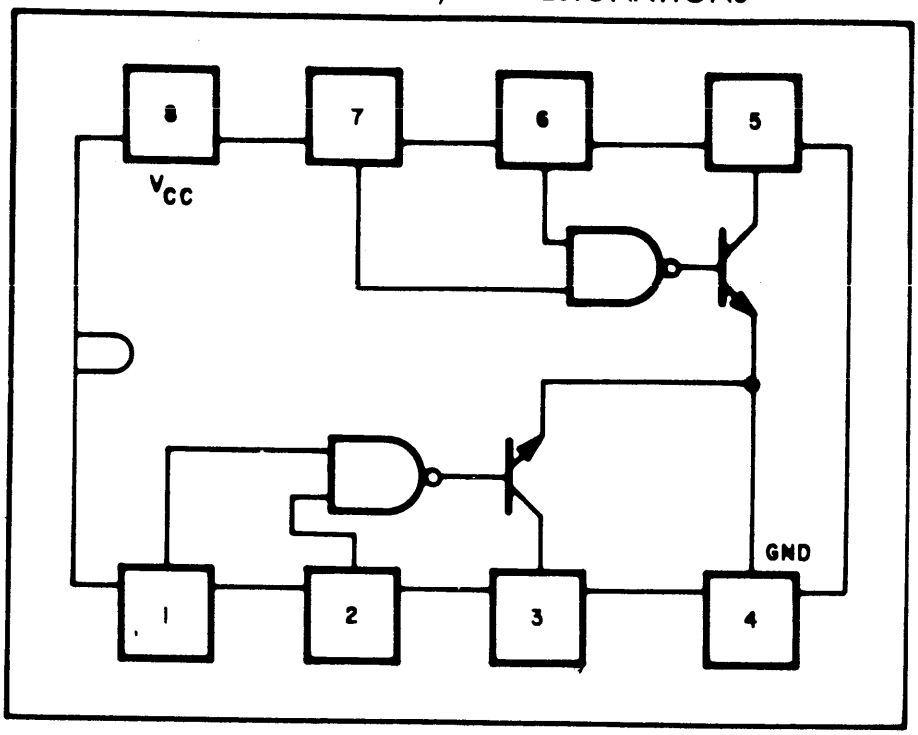
NOTES: 1. X = Logical 1 or logical 0.
2. Not more than one output is to be allowed to be ON at one time: When all timing inputs are at a logical 1, two of the address inputs must be at a logical 0.

The SN75 324 is a monolithic memory driver with decode inputs designed for use with magnetic memories. The device contains two 400-milliampere (source/sink) switch pairs, with decoding capability from four address lines. Two address inputs (B and C) are used for mode selection, i.e., source or sink. The other two address inputs (A and D) are used for switch-pair selection, i.e., output switch-pair Y/Z or W/X respectively.

75451

Interface Circuit

LOGIC DIAGRAM/PIN DESIGNATIONS



TRUTH TABLE

3 = 1 · 2
5 = 6 · 7

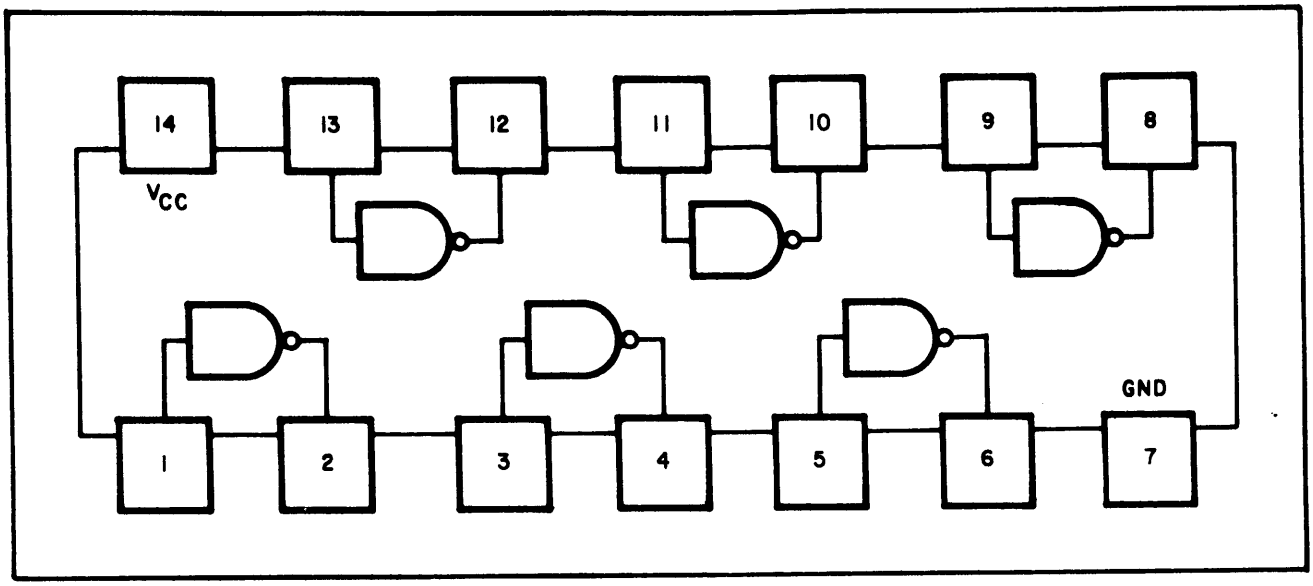
SN75 451 interface circuit - typical characteristics

Gate input current and voltage	TTL
Gate output current and voltage	TTL
Transistor collector-emitter voltage (I_C = 0.1 mA)	35V
Transistor collector substrate breakdown voltage (I_{CS} = 0.1 mA)	50V
Transistor saturated collector-emitter forward voltage (I_C = 0.1 mA)	0.3V
(I_C = 300 mA)	0.5V
Overall turn-on delay	16ns
Overall turn-off delay	17ns

8H90

Hex Inverter

LOGIC DIAGRAM/PIN DESIGNATIONS



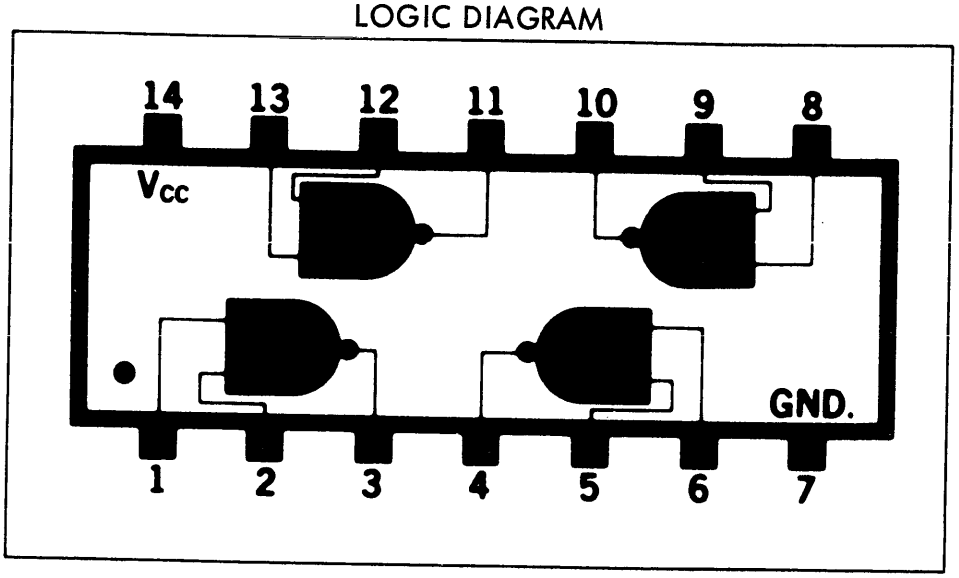
TRUTH TABLE

ANY INPUT LOW = HIGH OUT
ANY INPUT HIGH = LOW OUT

8T80

Quad 2 - Input NAND Interface Gate

LOGIC DIAGRAM



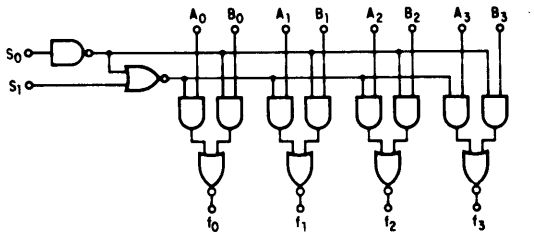
TRUTH TABLE

V_{IN}	V_{IN}	V_{OUT}
L	L	H
L	H	H
H	L	H
H	H	L

8234

2-INPUT 4-BIT DIGITAL MULTIPLEXERS

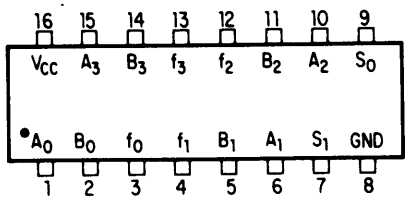
8234 LOGIC DIAGRAM AND TRUTH TABLE



TRUTH TABLE

S_0	S_1	f_n
0	0	$\overline{B}$
1	0	$\overline{A}$
0	1	$\overline{B}$
1	1	1

PIN CONFIGURATIONS

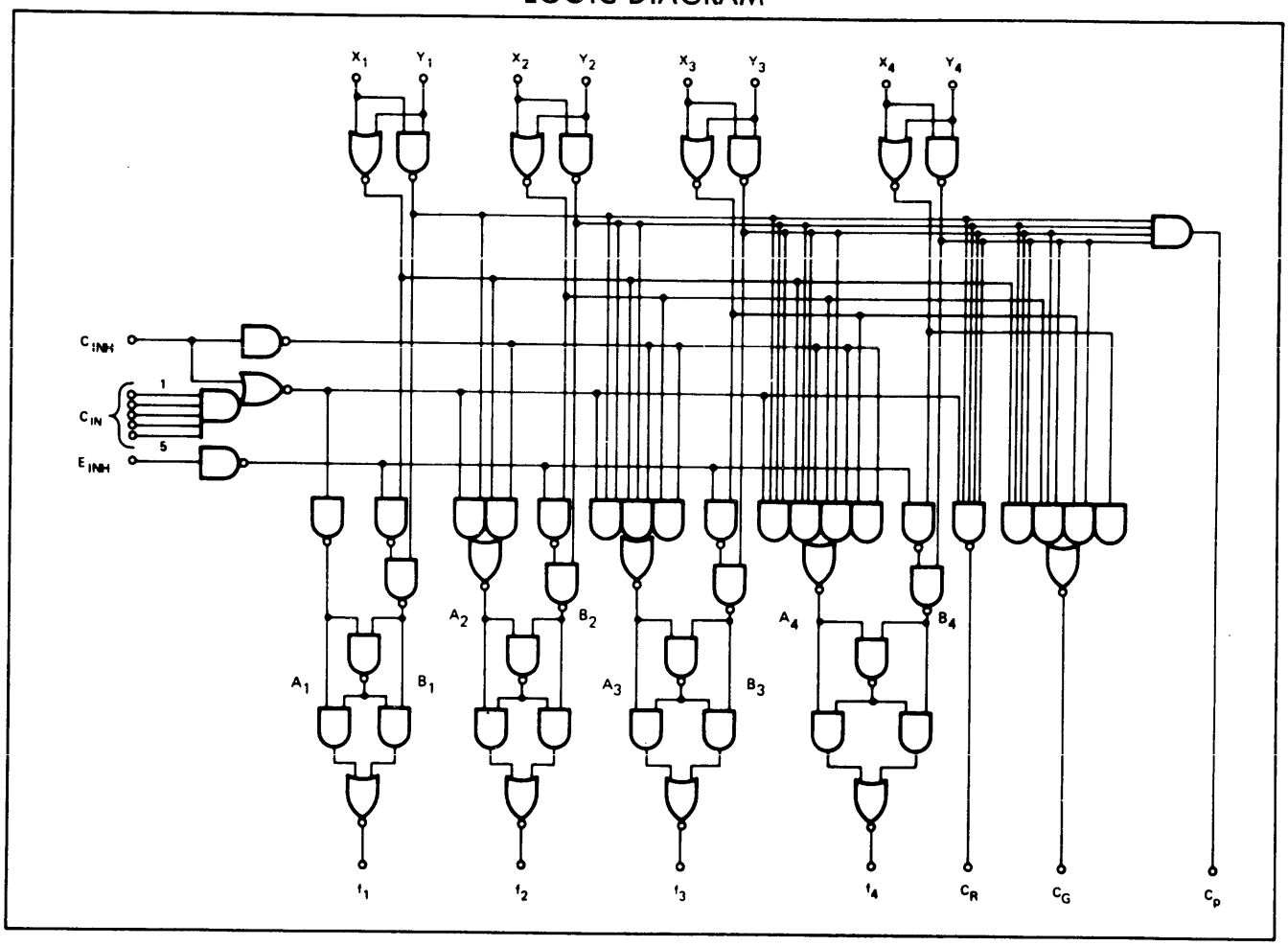


R and B PACKAGE

8260

Arithmetic Logic Element

LOGIC DIAGRAM



A and B refer to functional block diagram

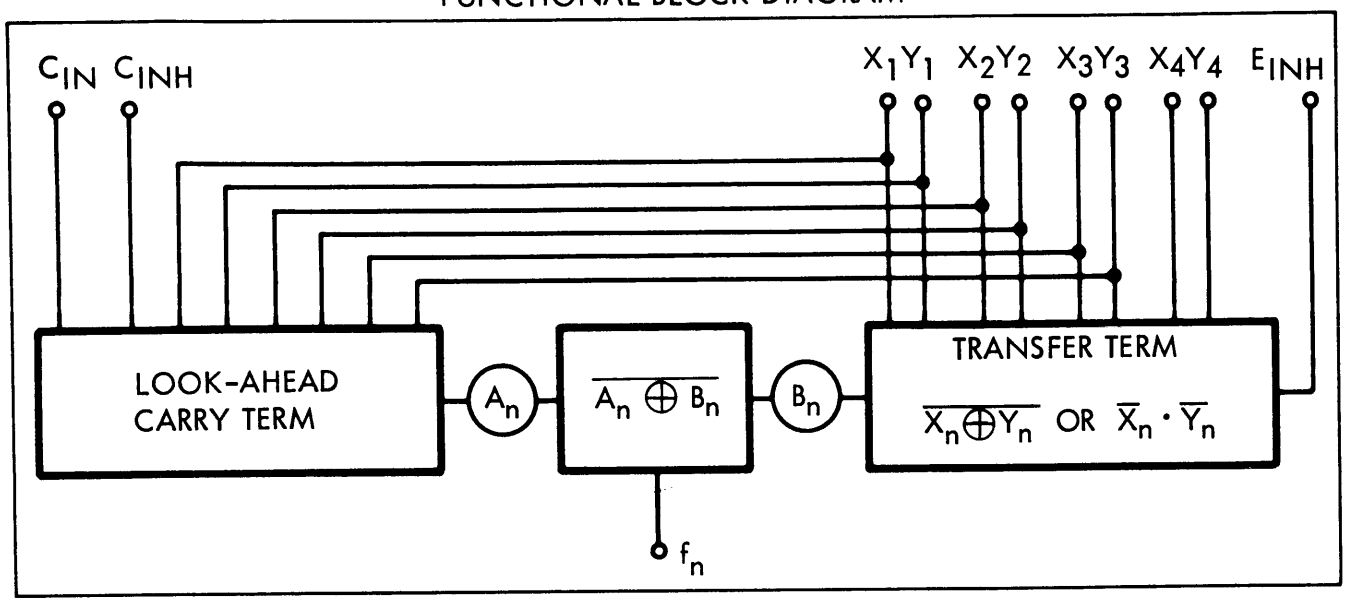
C_G = Internally Generated Carry

C_R = Ripple Carry

C_P = Propogated Carry

8260(cont.)

FUNCTIONAL BLOCK DIAGRAM



TRUTH TABLES

$C_{INH} = 1 \longrightarrow A_n = 1$

$C_{INH} = 0 \longrightarrow A_n =$

C_{IN}	A_1	A_1	X_1	Y_1	A_2	A_2	X_2	Y_2	A_3	A_3	X_3	Y_3	A_4
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	1	0	0	0	1	0
		0	1	0	0	0	1	0	0	0	1	0	0
		0	1	1	1	0	1	1	1	0	1	1	1
		1	0	0	0	1	0	0	0	1	0	0	0
		1	0	1	1	1	0	1	1	1	0	1	1
		1	1	0	1	1	1	0	1	1	1	0	1
		1	1	1	1	1	1	1	1	1	1	1	1

A_n	B_n	f_n
0	0	1
0	1	0
1	0	0
1	1	1

E_{INH}	X_n	Y_n	B_n
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

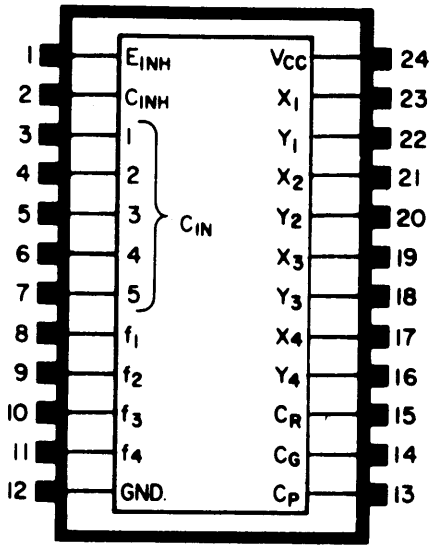
8260(cont.)

MODE OF OPERATION

INPUTS	Least Significant C_{IN} Inputs to be: *	CONTROLS		f	FUNCTION
		C_{INH}	E_{INH}		
X_n, Y_n ↓	0	0	0	Σ_n	Add
	0	0	1	- -	Not Used
	0	1	0	$X_nY_n + \overline{X_n}\overline{Y_n}$	Coincidence
	0	1	1	X_nY_n	AND
$\overline{X_n}$, $\overline{Y_n}$ ↓	1	C	0	$\overline{\Sigma}_n$	Add
	1	0	1	- -	Not Used
	1	1	0	$\overline{X_n}\overline{Y_n} + X_nY_n$	Coincidence
	1	1	1	$\overline{X_n}\overline{Y_n}$	AND

* Least significant of a "Multiple Package" adder system.

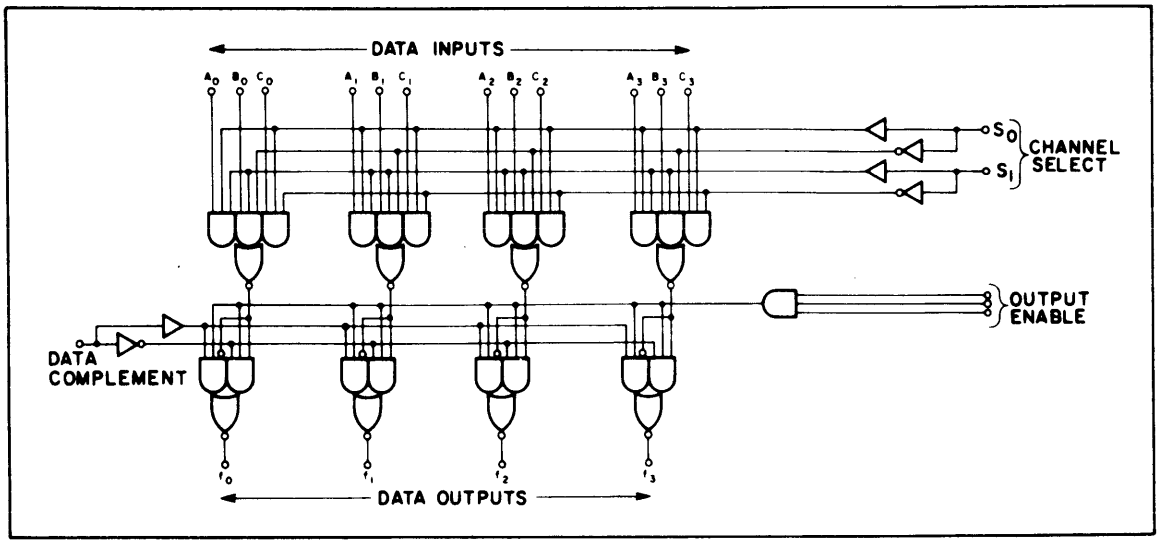
P PACKAGE



8264

3 - Input, 4-Bit Digital Multiplier

LOGIC DIAGRAM

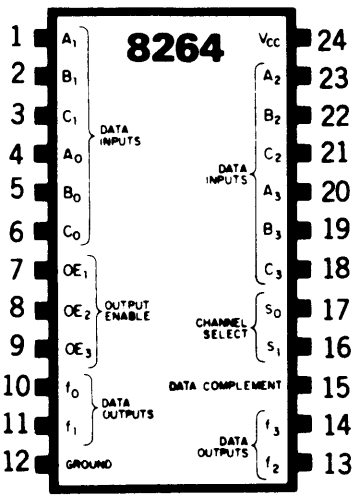


TRUTH TABLE

DATA INPUT A_n	B_n	C_n	CHANNEL SELECT S_0	S_1	DATA COMPLEMENT	OUTPUT ENABLE (8264)	DATA OUTPUTS
A_n	x	x	1	1	0	1	A_n
x	B_n	x	0	1	0	1	B_n
x	x	C_n	1	0	0	1	C_n
x	x	x	0	0	0	1	0
A_n	x	x	1	1	1	1	$\overline{A_n}$
x	B_n	x	0	1	1	1	$\overline{B_n}$
x	x	C_n	1	0	1	1	$\overline{C_n}$
x	x	x	0	**0**	1	1	1
x	x	x	x	x	x	0	1

x = Either State

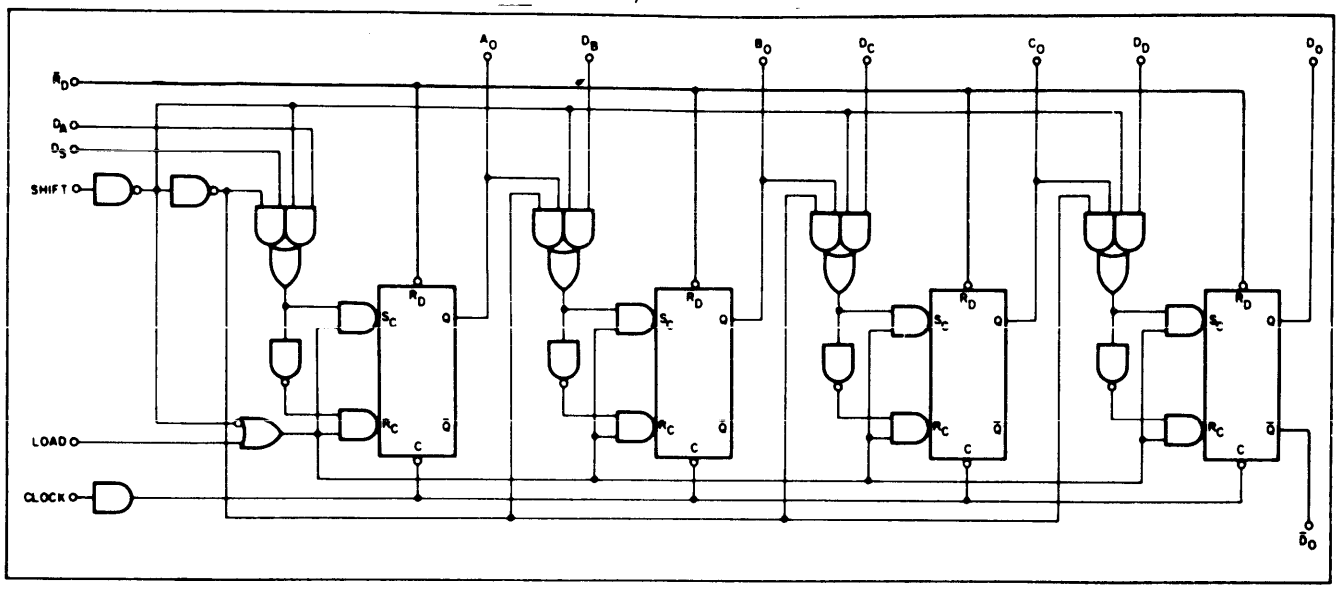
P, Y PACKAGE



8271

4 - Bit Shift Registers

LOGIC DIAGRAM/PIN DESIGNATIONS

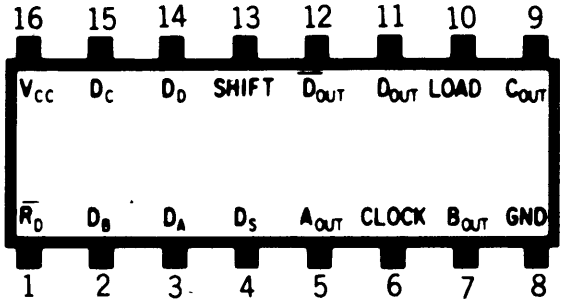


TRUTH TABLE

CONTROL STATE	LOAD	SHIFT
Hold	0	0
Parallel Entry	1	0
Shift Right	0	1
Shift Right	1	1

B PACKAGE

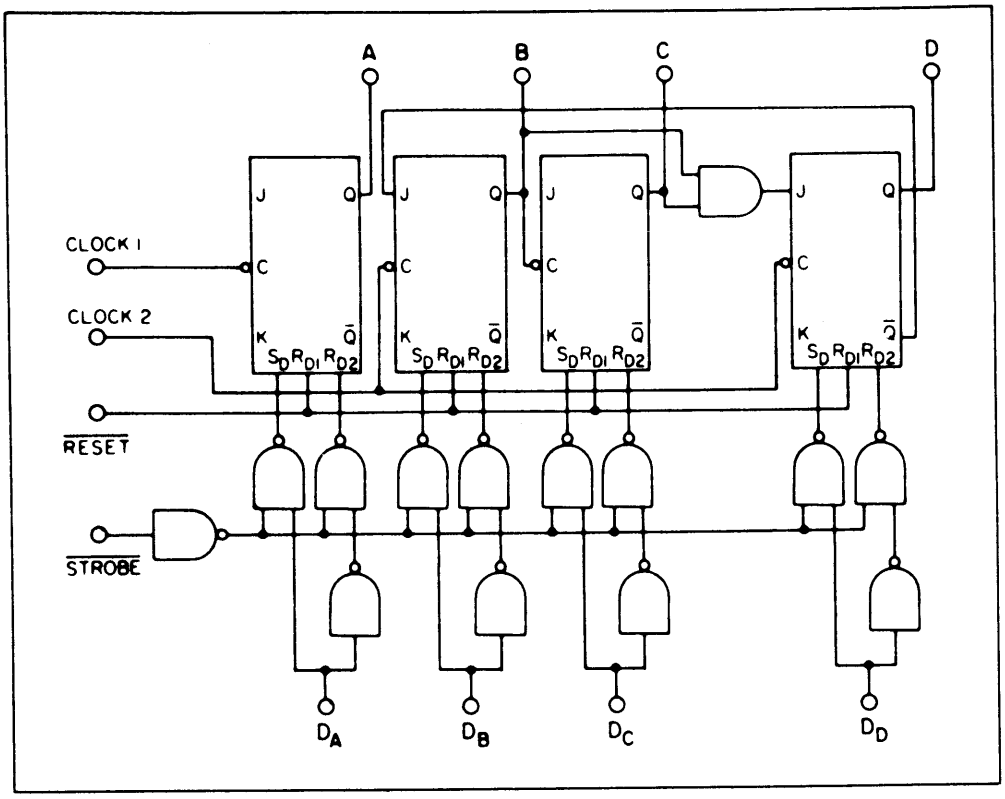
8271B



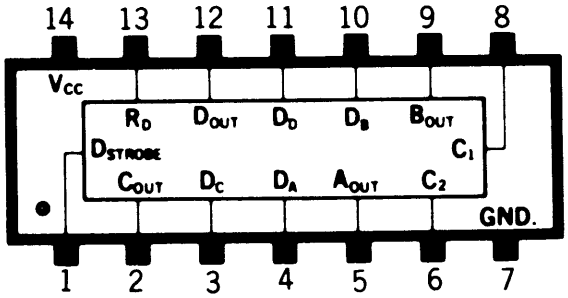
8280

BCD Decade Counter/Storage Element

LOGIC DIAGRAM



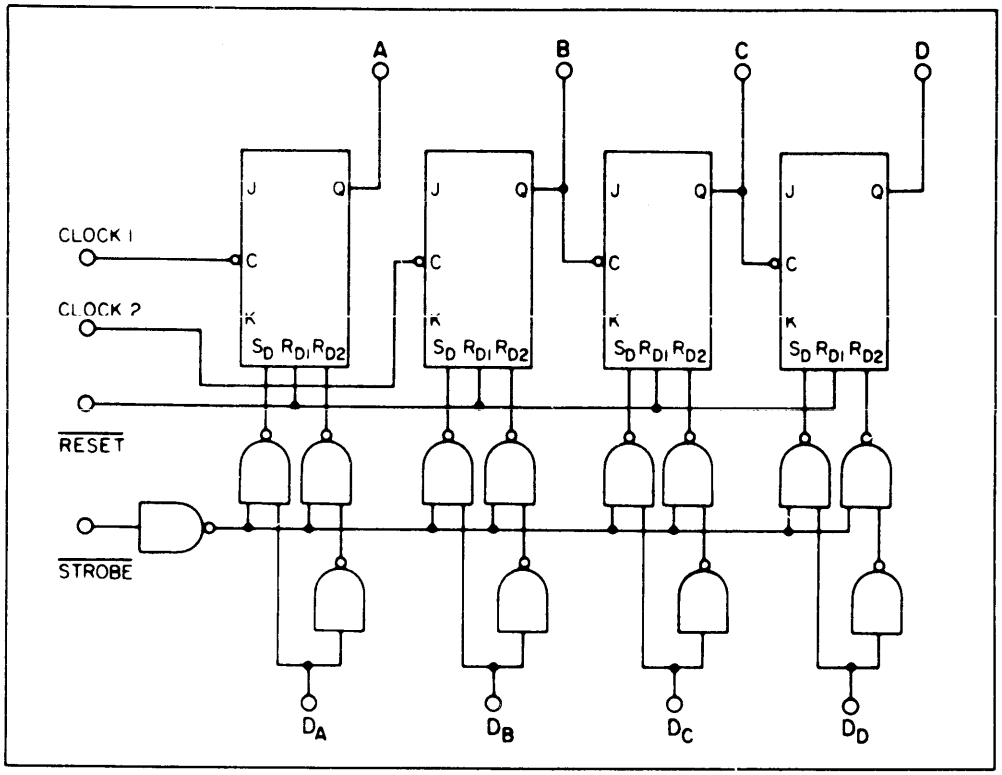
8280 has strobed parallel-entry for setting to any output state. A "1" or "0" at a data input will be transferred to the associated output when the strobe input is put at the "0" level. For additional flexibility, both units are provided with a reset input which is common to all four bits. A "0" on the reset line produces "0" at all four outputs. The counting operation is performed on the falling (negative-going) edge of the input clock pulse.



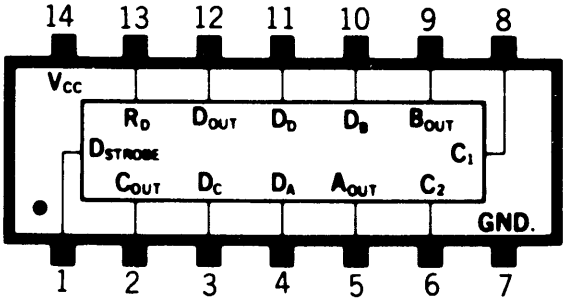
8281

4-Bit Binary Counter/Storage Element

LOGIC DIAGRAM



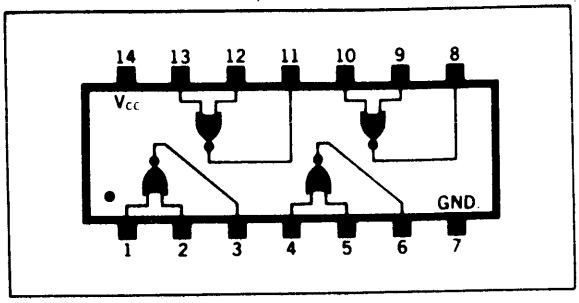
8281 has strobed parallel-entry for setting to any output state. A "1" or "0" at a data input will be transferred to the associated output when the strobe input is put at the "0" level. For additional flexibility, both units are provided with a reset input which is common to all four bits. A "0" on the reset line produces "0" at all four outputs. The counting operation is performed on the falling (negative-going) edge of the input clock pulse.



8885

Quad 2 - Input NOR Gate

LOGIC DIAGRAM/PIN DESIGNATIONS



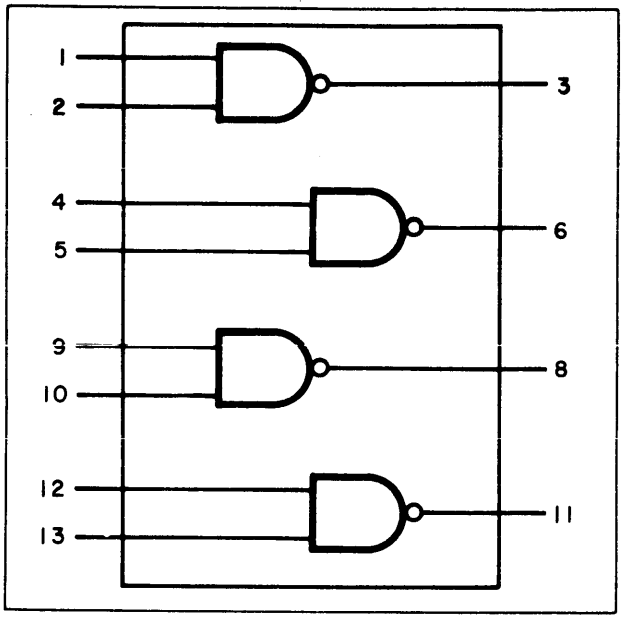
TRUTH TABLE

V_{IN}	V_{IN}	V_{OUT}
H	H	L
H	L	L
L	H	L
L	L	H

9002

Quad 2 - Input NAND Gates

LOGIC DIAGRAM/PIN DESIGNATIONS



V_{CC} = Pin 14

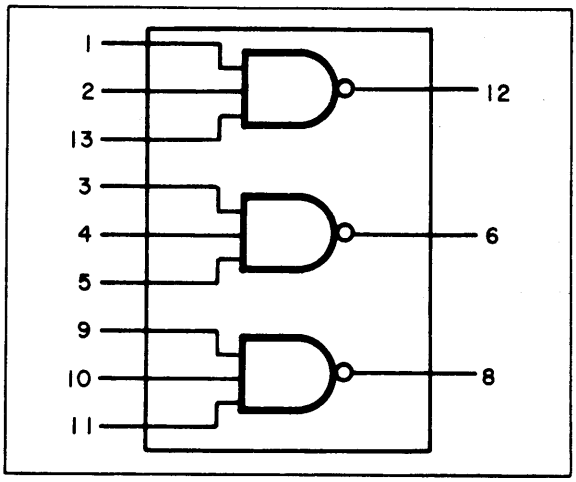
Gnd = Pin 7

TRUTH TABLE LISTED BELOW

9003

Triple 3 - Input NAND Gates

LOGIC DIAGRAM/PIN DESIGNATIONS



V_{CC} = Pin 14

Gnd = Pin 7

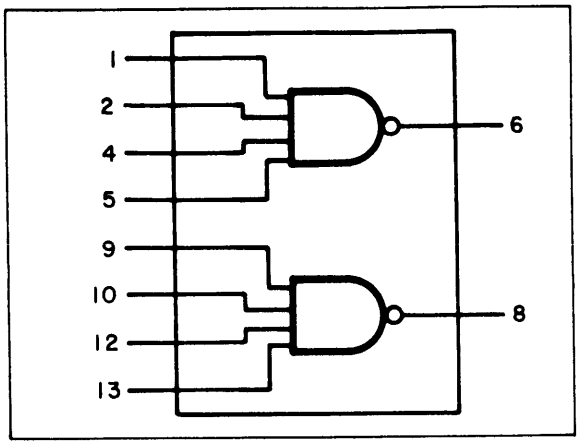
9002 & 9003 TRUTH TABLE

All Inputs High = Low Out
Any Input Low = High Out

9004/9009*

Dual 4 - Input NAND Gates

LOGIC DIAGRAM/PIN DESIGNATIONS



V_{CC} = Pin 14

Gnd = Pin 7

*9009 Has Higher Input-Output Loading Parameters Than 9004

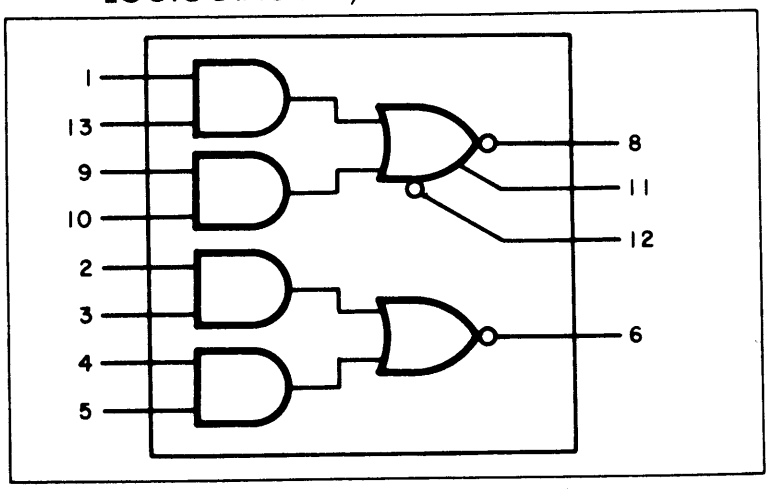
TRUTH TABLE

All Inputs High = Low Out
Any Input Low = High Out

9005

Dual Extendable AND-OR-INVERT Gates

LOGIC DIAGRAM/PIN DESIGNATIONS



*Four Extenders (9006) may be tied to these terminals

V_{CC} = Pin 14

Gnd = Pin 7

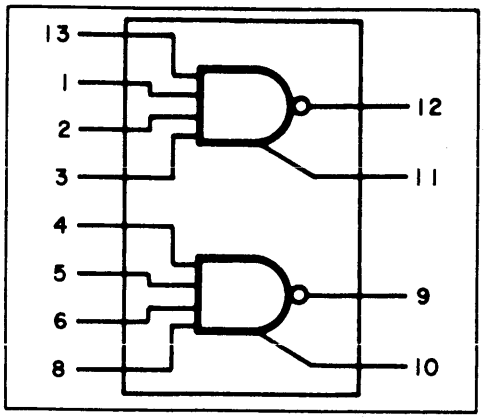
TRUTH TABLE

$(\underline{2} \cdot 3) + (4 \cdot \underline{5}) = \overline{6}$
$(2 + 3) \cdot (\overline{4} + 5) = 6$

9006

Dual Extender AND-OR-INVERT Gates

LOGIC DIAGRAM



Extender for use with 9005 & 9008

V_{CC} = Pin 14

Gnd = Pin 7

TRUTH TABLE

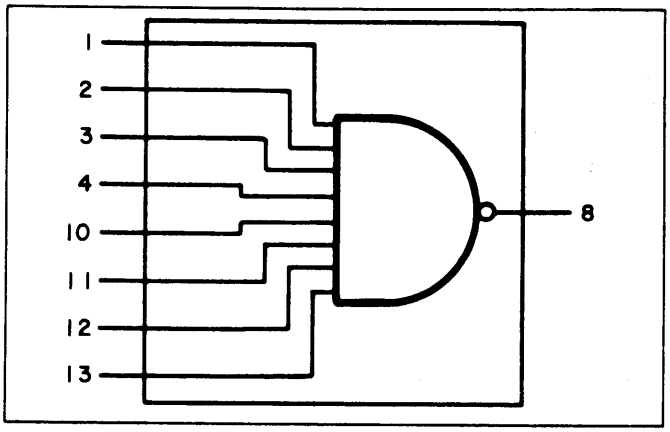
$\overline{4 \cdot 5 \cdot 6 \cdot 8} = \overline{9}$

$\overline{4} + \overline{5} + \overline{6} + \overline{8} = 9$

9007

8 - Input NAND Gate

LOGIC DIAGRAM



V_{CC} = Pin 14

Gnd = Pin 7

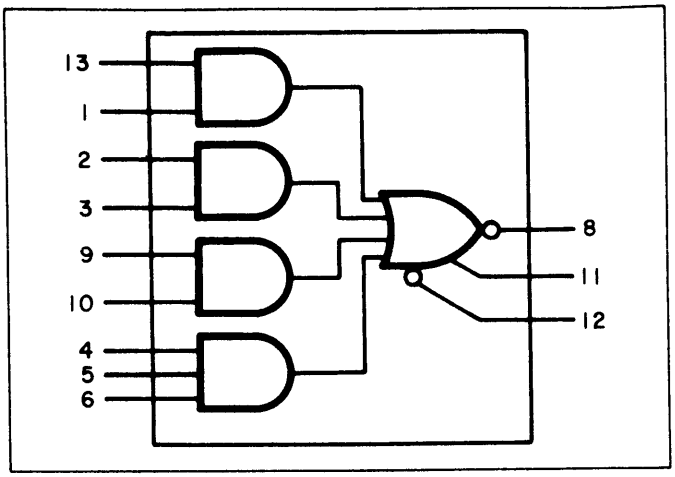
TRUTH TABLE

ALL INPUTS HIGH = LOW OUT
ANY INPUT LOW = HIGH OUT

9008

Single Extendable AND-OR-INVERT Gate

LOGIC DIAGRAM/PIN DESIGNATIONS



*Four Extenders (9006) may be tied to these terminals

V_{CC} = Pin 14
Gnd = Pin 7

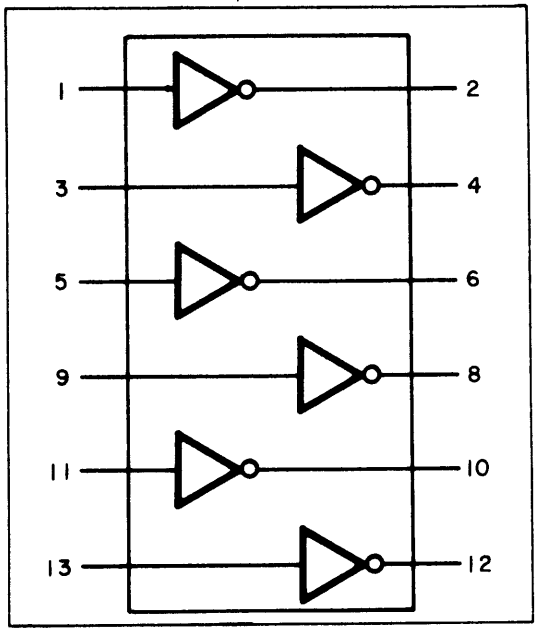
TRUTH TABLE

$$(\overline{1} \cdot 13) + (\overline{2} \cdot 3) + (\overline{9} \cdot \overline{10}) + (\overline{4} \cdot \overline{5} \cdot \overline{6}) = \overline{8}$$
$$(\overline{1} + 13) \cdot (\overline{2} + 3) \cdot (\overline{9} + \overline{10}) \cdot (\overline{4} + \overline{5} + \overline{6}) = 8$$

9016

Quad Hex Inverter

LOGIC DIAGRAM/PIN DESIGNATIONS



V_{CC} = Pin 14
Gnd = Pin 7

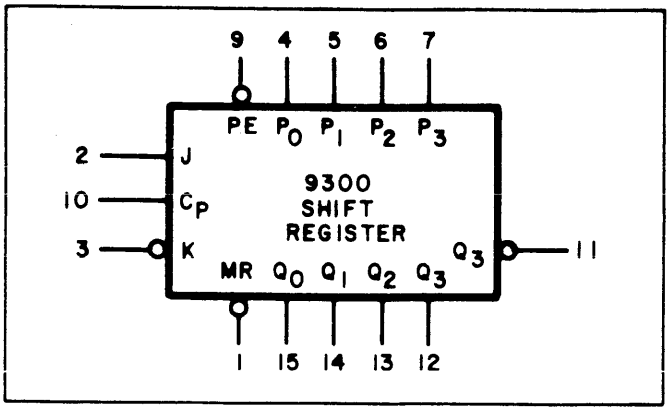
TRUTH TABLE

ANY INPUT LOW = HIGH OUT
ANY INPUT HIGH = LOW OUT

9300

4 - Bit Shift Register

LOGIC DIAGRAM



V_{CC} = Pin 14
Gnd = Pin 8

PIN NOMENCLATURE

$\overline{PE}$	Parallel Enable (Active Low) Input
P_0, P_1, P_2, P_3	Parallel Inputs
J	First Stage J (Active High) Input
$\overline{K}$	First Stage K (Active Low) Input
Cp	Clock Active High Going Edge Input
$\overline{MR}$	Master Reset (Active High) Input
Q_0, Q_1, Q_2, Q_3	Parallel Outputs
$\overline{Q_3}$	Complementary Last Stage Output

Data entry is synchronous with the registers changing state after each low to high transition of the clock. With the parallel enable low the parallel inputs determine the next condition of the shift register. When the parallel enable input is high the shift register performs a one bit shift to the right, with data entering the first stage flip-flop through $J\overline{K}$ inputs. By tying the two inputs together D type entry is obtained.

The asynchronous active low master reset when activated overrides all other input conditions and clears the register.

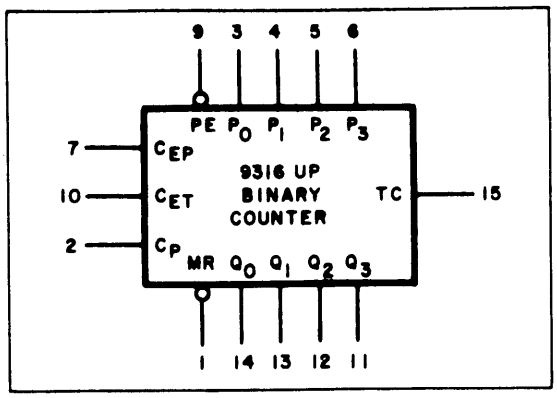
TRUTH TABLE FOR SERIAL ENTRY

J	$\overline{K}$	Q_o at t_{n+1}
L	L	L
L	H	Q_o at t_n (no change)
H	L	$\overline{Q}_o$ at t_n (toggles)
H	H	H

$\overline{PE}$ = HIGH, $\overline{MR}$ = HIGH (n + 1) indicates state after next clock

9316
Binary Up Counter

LOGIC DIAGRAM



V_{CC} = Pin 16
Gnd = Pin 8

PIN NOMENCLATURE

$\overline{PE}$	Parallel Enable (Active Low) Input
P_0, P_1, P_2, P_3	Parallel Inputs
CEP	Count Enable Parallel Input
CET	Count Enable Trickle Input
$\underline{Cp}$	Clock Active High Going Edge Input
$\overline{MR}$	Master Reset (Active Low) Input
Q_0, Q_1, Q_2, Q_3	Parallel Outputs
TC	Terminal Count Output

The counter is synchronous with the counter outputs changing state after the low to high transition of the clock. When conditions are satisfied for counting, a clock pulse will change the counter to the next state of a binary sequence. When the parallel enable is low the parallel inputs determine the next state of the counter synchronously with the clock.

Mode selection is accomplished as shown below. However a restriction is placed on the manner of selection. The transition of CEP or CET from high to low or of $\overline{PE}$ from low to high may only be done when CP is high.

By utilizing the two count enable inputs and terminal count output, multi-stage synchronous counting is obtained, with operating speeds equivalent to that of a single stage.

When the asynchronous master reset is active outputs Q_{0-3} will be forced low regardless of all other input conditions.

MODE SELECTION

$\overline{PE}$	CE (Count Enable)	MODE
H	H	Count Up
H	L	No Change
L	X	Presetting

Where CE (Count Enable) = CEP • CET

H = High Voltage Level
L = Low Voltage Level
X = Don't Care Condition

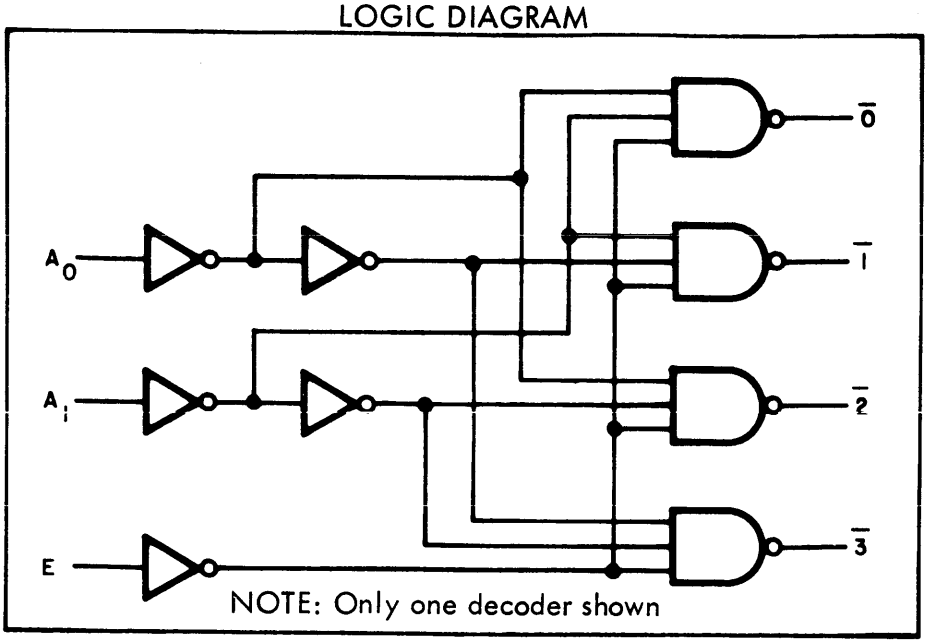
LOGIC EQUATION FOR TERMINAL COUNT

$$TC = CET \cdot Q_0 \cdot Q_1 \cdot Q_2 \cdot Q_3$$

9321

Dual One-Of-Four Decoder

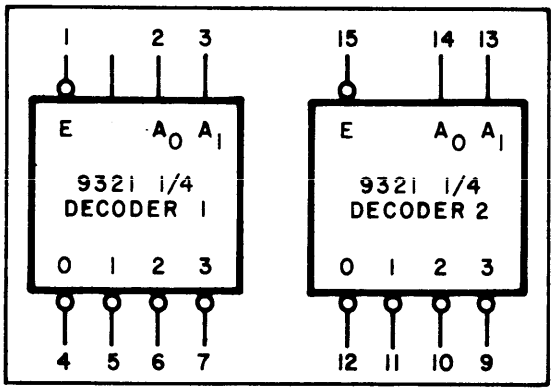
LOGIC DIAGRAM



NOTE: Only one decoder shown

V_{CC} = Pin 16
GND = Pin 8

PIN DESIGNATIONS



H = High Voltage Level
L = Low Voltage Level
X = Don't Care Condition

TRUTH TABLE

$\overline{E}$	A_0	A_1	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	H	L	H	L	H	H
L	L	H	H	H	L	H
L	H	H	H	H	H	L

PIN NOMENCLATURE

Decoder 1 and 2

$\overline{E}$ Enable (Active Low) Input

A_0, A_1 Address Inputs

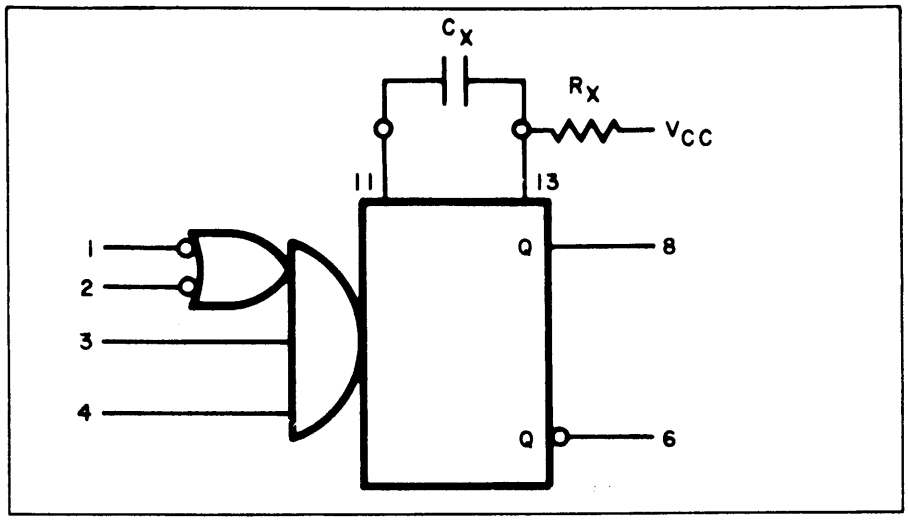
$\overline{0}$, $\overline{1}$, $\overline{2}$, $\overline{3}$ (Active Low) Outputs

The 9321 consists of two independent one-of-four decoders, each with an active low enable. Each decoder accepts two inputs and provides one of four mutually active low outputs.

9601

Monostable Multivibrator (One Shot)

LOGIC DIAGRAM



V_{CC} = Pin 14
Gnd = Pin 7

The inputs are dc coupled hence triggering is independent of input transition times. If the input signal is applied to an active high input, triggering will occur on the rising edge of the waveform. By applying the input signal to an active low input, triggering will occur on the falling edge of the waveform.

Each time the input conditions for triggering are met, the external capacitor is discharged and a new cycle is started. Successive inputs with a period shorter than the delay time (R_XC_X) retrigger the monostable resulting in a continuous true output. Retriggering may be inhibited by tying the negation ($\overline{Q}$) output back to an active level low input.

The formula for calculating the delay time constant is:

0.36 x R(in ohms) x C(in Farads) = T(in seconds)

APPENDIX B

NOVA 800 SIGNAL LIST

SIGNAL ORIGIN

CENTRAL PROCESSOR

AND

MEMORY

NOTE: Blank entries in level column denote flip-flop outputs which may be in either logic state.
Accumulators, Adders, I/O DATA lines, memory INHIBIT lines, MEMORY ADDRESS (MA) flip-flops, MEMORY BUFFER (MB) flip-flops, PROGRAM COUNTER (PC) flip-flops, and RINH flip-flops output levels are not defined in the level column.

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
ACD Ø	74	1Ø		93-2	D-MULT Ø	73	6		93-2
ACD 1	74	9		"	D-MULT 1	73	3		"
ACD 2	74	7		"	D-MULT 2	73	18		"
ACD 3	74	6		"	D-MULT 3	73	21		"
ACD 4	7Ø	1Ø		"	D-MULT 4	69	6		"
ACD 5	7Ø	9		"	D-MULT 5	69	3		"
ACD 6	7Ø	7		"	D-MULT 6	69	18		"
ACD 7	7Ø	6		"	D-MULT 7	69	21		"
ACD 8	66	1Ø		93-3	D-MULT 8	65	6		93-3
ACD 9	66	9		"	D-MULT 9	65	3		"
ACD1Ø	66	7		"	D-MULT1Ø	65	18		"
ACD11	66	6		"	D-MULT11	65	21		"
ACD12	62	1Ø		"	D-MULT12	61	6		"
ACD13	62	9		"	D-MULT13	61	3		"
ACD14	62	7		"	D-MULT14	61	18		"
ACD15	62	6		"	D-MULT15	61	21		"
AC-H-WRITE	36	5		93-1	ACSXØ-3	75	13		93-2
					ACSX 4-7	71	13		"
					ACSX 8-11	67	13		93-3
					ACSX12-15	63	13		"
					ACD Ø-3	74	13		93-2
					ACD 4-7	7Ø	13		"
					ACD 8-11	66	13		93-3
					ACD12-15	62	13		"
AC-L-WRITE	36	7		93-1		34	4,6	H	93-1
					ACSX Ø-3	75	14		93-2
					ACSX 4-7	71	14		"
					ACSX 8-11	67	14		93-3
					ACSX12-15	63	14		"
					ACD Ø-3	74	14		93-2
					ACD 4-7	7Ø	14		"
					ACD 8-11	66	14		"
					ACD12-15	62	14		"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
ACSX Ø	75	1Ø		93-2	SX-MULT Ø	76	6		93-2
ACSX 1	75	9		"	SX-MULT 1	76	3		"
ACSX 2	75	7		"	SX-MULT 2	76	18		"
ACSX 3	75	6		"	SX-MULT 3	76	21		"
ACSX 4	71	1Ø		"	SX-MULT 4	72	6		"
ACSX 5	71	9		"	SX-MULT 5	72	3		"
ACSX 6	71	7		"	SX-MULT 6	72	18		"
ACSX 7	71	6		"	SX-MULT 7	72	21		"
ACSX 8	67	1Ø		93-3	SX-MULT 8	68	6		93-3
ACSX 9	67	9		"	SX-MULT 9	68	3		"
ACSX1Ø	67	7		"	SX-MULT1Ø	68	18		"
ACSX11	67	6		"	SX-MULT11	68	21		"
ACSX12	63	1Ø		"	SX-MULT12	64	6		"
ACSX13	63	9		"	SX-MULT13	64	3		"
ACSX14	63	7		"	SX-MULT14	64	18		"
ACSX15	63	6		"	SX-MULT15	64	21		"
ADDER=Ø	A77			93-1					
	83	6,3	(OC)	93-2	AUTO INC+DEC	42	2	H	92-2
	83	11,8	(OC)	"	FETCHSKIP*	4	4	H	92-3
	81	6,3	(OC)	"	INC PC	96	9	H	"
	81	11,8	(OC)	"					
	79	6,3	(OC)	93-3					
	79	11,8	(OC)	"					
	77	6,3	(OC)	"					
	77	11,8	(OC)	"					
ADDER TO MEM*	1Ø3	8	L	92-3	(B34)				93-1
					MBO Ø-3*	6Ø	7	L	93-2
					MBO 4-7*	55	7	L	"
					MBO 8-11*	5Ø	7	L	93-3
					MBO12-15*	45	7	L	"

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
ADDERØ	1Ø4	11		93-2	MBOØ*	6Ø	1		93-2
					SUMØ	1Ø3	13		"
					SUM1	1Ø2	1Ø		"
					SUM8	93	5		93-3
					CRYTEST*	4Ø	1		93-1
ADDER1	1Ø4	1Ø		93-2	MBO1*	6Ø	6		93-2
					SUM1	1Ø2	13		"
					SUM2	1Ø1	1Ø		"
					SUMØ	1Ø3	3		"
					SUM9	92	5		93-3
ADDER2	1Ø4	9		93-2	MBO2*	6Ø	1Ø		93-2
					SUM1	1Ø2	3		"
					SUM2	1Ø1	13		"
					SUM3	1ØØ	1Ø		"
					SUM1Ø	91	5		93-3
ADDER3	1Ø4	8		93-2	MBO3*	6Ø	15		93-2
					SUM2	1Ø1	3		"
					SUM3	1ØØ	13		"
					SUM4	98	1Ø		"
					SUM11	9Ø	5		93-3
ADDER4	99	11		93-2	MBO4*	55	1		93-2
					SUM3	1ØØ	3		"
					SUM4	98	13		"
					SUM5	97	1Ø		"
					SUM12	88	5		93-3
ADDER5	99	1Ø		93-2	MBO5*	55	6		93-2
					SUM4	98	3		"
					SUM5	97	13		"
					SUM6	96	1Ø		"
					SUM13	87	5		93-3
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
ADDER6	99	9		93-2	MBO6*	55	1Ø		93-2
					SUM5	97	3		"
					SUM6	96	13		"
					SUM7	95	1Ø		"
					SUM14	86	5		93-3
ADDER7	99	8		93-2	MBO7*	55	15		93-2
					SUM6	96	3		"
					SUM7	95	13		"
					SUM8	93	1Ø		93-3
					SUM15	85	5		"
ADDER8	94	11		93-3	MBO8*	50	1		93-3
					SUM7	95	3		93-2
					SUM8	93	13		93-3
					SUM9	92	1Ø		"
					SUMØ	1Ø3	5		93-2
ADDER9	94	1Ø		93-3	MBO9*	5Ø	6		93-3
					SUM8	93	3		"
					SUM9	92	13		"
					SUM1Ø	91	1Ø		"
					SUM1	1Ø2	5		93-2
ADDER1Ø	94	9		93-3	MBO1Ø*	5Ø	1Ø		93-3
					SUM9	92	3		"
					SUM1Ø	91	13		"
					SUM11	9Ø	1Ø		"
					SUM2	1Ø1	5		93-2
ADDER11	94	8		93-3	MBO11*	5Ø	15		93-3
					SUM1Ø	91	3		"
					SUM11	9Ø	13		"
					SUM12	88	1Ø		"
					SUM3	1ØØ	5		93-2

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
ADDER12	89	11		93-3	MBO12*	45	1		93-3
					SUM11	9Ø	3		"
					SUM12	88	13		"
					SUM13	87	1Ø		"
					SUM4	98	5		93-2
ADDER13	89	1Ø		93-3	MBO13*	45	6		93-3
					SUM12	88	3		"
					SUM13	87	13		"
					SUM14	86	1Ø		"
					SUM15	97	5		93-2
ADDER14	89	9		93-3	MBO14*	45	1Ø		93-3
					SUM13	87	3		"
					SUM14	86	13		"
					SUM15	85	1Ø		"
					SUM6	96	5		93-2
ADDER15	89	8		93-3	MBO15*	45	15		93-3
					SUM14	86	3		"
					SUM15	85	13		"
					SUM7	95	5		93-2
					CRYTEST*	4Ø	1Ø		93-1
ALC*	56	8	L	92-2	ALC	77	5	L	92-2
					AND	57	5	L	"
					HAS E CYCLE*	58	5	L	"
					(MULT)	46	9	L	92-3
					(MULT)	43	9	L	"
ALC	77	6	H	92-2	FORCE SEL X*	1Ø9	2	H	92-3
					FORCE SELY*	1Ø9	4	H	"
					LOAD CRY*	1Ø6	9	H	"
					FETCH SKIP*	1Ø1 (A57)	4	H	"
AND	57	6	H	92-2	(CARRY LOGIC)	32	11	H	93-1
					ADDER Ø-3	1Ø4	1,2	H	93-2
					ADDER 4-7	99	1,2	H	"
					ADDER 8-11	94	1,2	H	93-3
					ADDER12-15	89	1,2	H	"

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
AND ENABLE*	28	12	L	92-2	IO SKIP	57	2	L	92-2
					WRITE AC*	1Ø7	5	H	92-3
ALC+IO SET	21	8	H	92-2	LOAD PC*	93	6	H	92-3
ACWRITE*	11	8	L	93-1	ACSX Ø-3	75	12	L	93-2
					ACSX 4-7	71	12	L	"
					ACSX 8-11	67	12	L	93-3
					ACSX 12-15	63	12	L	"
					ACDØ-3	74	12	L	93-2
					ACD4-7	7Ø	12	L	"
					ACD8-11	66	12	L	93-3
					ACD12-15	62	12	L	"
ARØ	57	11		93-2	ACSXØ	75	15		93-2
					ACDØ	74	15		"
					SX-MULTØ	76	4		"
					SUM CRY	39	1	H	93-1
AR1	57	9		93-2	ACSX1	75	1		93-2
					ACD1	74	1		"
					SX-MULT1	76	1		"
AR2	57	7		93-2	ACSX2	75	2		93-2
					ACD2	74	2		"
					SX-MULT2	76	2Ø		"
AR3	57	5		93-2	ACSX3	75	3		93-2
					ACD3	74	3		"
					SX-MULT3	76	23		"
AR4	52	11		93-2	ACSX4	71	15		93-2
					ACD4	7Ø	15		"
					SX-MULT4	72	4		"
					ARØ-4	57	4	H	"
AR5	52	9		93-2	ACSX5	71	1		93-2
					ACD5	7Ø	1		"
					SX-MULT5	72	1		"
AR6	52	7		93-2	ACSX6	71	2		"
					ACD6	7Ø	2		"
					SX-MULT6	72	2Ø		"
AR7	52	5		93-2	ACSX7	71	3		93-2
					ACD7	7Ø	3		"
					SX-MULT7	72	23		"
AR8	47	11		93-3	ACSX8	67	15		93-3

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
AR8(continued)					SX-MULT8	68	4		93-3
					AR4-7	52	4	H	"
AR9	47	9		93-3	ACSX9	67	1		"
					SX-MULT9	68	1		"
AR1Ø	47	7		"	ACSX1Ø	67	2		93-3
					ACD1Ø	66	2		"
					SX-MULT1Ø	68	2Ø	"	"
AR11	47	5		93-3	ACSX11	67	3		93-3
					ACD11	66	3		"
					SX-MULT11	68	23		"
AR12	42	11		93-3	ACSX12	63	15		93-3
					ACD12	62	15		"
					SX-MULT12	64	4		"
					AR8-11	47	4	H	"
AR13	42	9		93-3	ACSX13	63	1		93-3
					ACD13	62	1		"
					SX-MULT13	64	1		"
AR14	42	7		93-3	ACSX14	63	2		93-3
					ACD14	62	2		"
					SX-MULT14	64	2Ø		"
AR15	42	5		93-3	ACSX15	63	3		93-3
					ACD15	62	3		"
					SX-MULT15	64	23		"
AUT DEC	39	5	H	92-2	FORCE SX COM*	63	1	H	92-3
AUT INC+DEC	39	7	H	92-2	FORCE PLUS ONE*	73	2	H	92-3
					MB LD EN*	1Ø8	12	H	"
CARRY*	Back Plane	A15	L	92-2	CARRY*	Con-nector	J4 49		to→
					(Indicator light)	6	5		89-1
CG	1Ø4	14	H	"	CR-CG*	13	2	H	"
CLKA	37	5		92-1	CLK B	37	12		92-1
					MEM CLK	18	12,13		"
					MEM CLK	18	9,1Ø		"
					PC CLK	5Ø	3		"
					CPU CLK	49	5, 9		"

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
CLKA (continued)					MA LOAD*	38	13		92-1
CLKA*	37	6		92-1	CLK-B	51	5	L	92-1
CLKB	37	9		92-1	DCHA*	71	13		92-4
					DCHO	72	1		"
					MA LOAD*	52	9		92-1
						38	3		"
CLKB*	37	8		93-1					
CLR	26	6	H	92-4		(A5Ø)			92-4
					ION*	92	3	H	"
CON DATA*	29	8		92-4		(A28)	(P46)	L	89-1
					[CONØ*] (S11)	6	3	L	"
					[CON1*] (S12)	6	1	L	"
					[CON2*] (S13)	6	9	L	"
					[CON3*] (S14)	6	13	L	"
					[CON4*] (S15)	3	9	L	"
					[CON5*] (S16)	3	11	L	"
					[CON6*] (S17)	3	5	L	"
					[CON7*] (S18)	3	3	L	"
					[CON8*] (S19)	3	1	L	"
					[CON9*] (S2Ø)	3	13	L	"
					[CON1Ø*] (S21)	4	9	L	"
					[CON11*] (S22)	4	11	L	"
					[CON12*] (S23)	4	13	L	"
					[CON13*] (S24)	4	5	L	"
					[CON14*] (S25)	4	3	L	"
					[CON15*] (S26)	4	1	L	"
					[CON DATA]	48	3	L	92-4
[CON DATA]	48	4	H	92-4	INH TRANS*	3Ø	9	H	"
					CON DATA*	29	1Ø	H	"
CON INST*	73	6	L	92-2		(A22)	(P22)		
					PRESET*	12	2	L	92-2
					(IR LOGIC)	5Ø	12	L	"
					[CON INST]	5	9	L	89-1
[CON INST]	5	8	H	89-1	MEMØ*	1	2	H	"
					MEM1*	1	4	H	"
					MEM2*	2	1Ø	H	"
					MEM3*	1	12	H	"
					MEM4*	1	1Ø	H	"

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
[CON INST] (continued)					MEM5*	2	12	H	89-1
					MEM6*	2	2	H	"
					MEM7*	2	4	H	"
CON RQ*	U5	6		89-1	→ Back Plane		A27		
	Back Plane		A27→		Key Seen	81	13	L	92-2
CONT+ISTP+MSTP*	Con-								
	nector	J4 pin 47 →			Back Plane		A25		
	Back Plane		A25 →		KEY·PRESET*	97	1		92-2
[CONØ*](S11)	6	4	L	89-1	MEMØ*	(B71)	(391)	L	89-1
					(CON IND)	7	9	L	"
[CON1*] (S12)	6	2	L	89-1	MEM1*	(B7Ø)	(P41)	L	"
					(CON IND)	7	13	L	"
[CON2*] (S13)	6	8	L	89-1	MEM2*	(B47)	(P13)	L	"
					(CON IND)	7	3	L	"
[CON3*] (S14)	6	12	L	89-1	MEM3*	(B68)	(P43)	L	"
					(CON IND)	7	1	L	"
[CON4*] (S15)	3	8	L	89-1	MEM4*	(B28)	(P37)	L	"
					(CON IND)	8	13	L	"
[CON5*] (S16)	3	1Ø	L	89-1	MEM5*	(B26)	(P36)	L	"
					(CON IND)	8	3	L	"
[CON6*] (S17)	3	6	L	89-1	MEM6*	(B22)	(P1Ø)	L	"
					(CON IND)	8	1	L	"
[CON7*] (S18)	3	4	L	89-1	MEM7*	(B24)	(P42)	L	"
					(CON IND)	9	13	L	"
[CON8*] (S19)	3	2	L	89-1	MEM8*	(A55)	(P34)	L	"
					(CON IND)	9	3	L	"
[CON9*] (S2Ø)	3	12	L	89-1	MEM9*	(A53)	(P7)	L	"
					(CON IND)	9	1	L	"
[CON1Ø*] (S21)	4	8	L	89-1	MEM1Ø*	(A45)	(P32)	L	"
					(CON IND)	1Ø	13	L	"
[CON11*] (S22)	4	1Ø	L	89-1	MEM11*	(A51)	(P31)	L	89-1
					(CON IND)	1Ø	3	L	"
[CON12*] (S23)	4	12	L	89-1	MEM12*	(A36)	(P5)	L	"
					(CON IND)	1Ø	1	L	"
[CON13*] (S24)	4	6	L	89-1	MEM13*	(A35)	(P29)	L	89-1
					(CON IND)	11	13	L	"
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
[CON14*] (S25)	4	4	L	89-1	MEM14*	(B76)	(P3)	L	89-1
					(CON IND)	11	3	L	"
[CON15*] (S26)	4	2	L	89-1	MEM15*	(B18)	(P2)	L	"
					(CON IND)	12	13	L	"
CPU CLK	49	6,8		92-1	MB LOAD	18	5	H	92-1
					INHIBIT F/F	35	13	H	"
						34	13	L	"
					READ CY F/F	52	5	H	"
					IRØ→IR15	73	1Ø	H	92-2
					STATES	78	6	L	"
					"	39	6	L	"
					DCH LOGIC	69	6	L	92-4
					RUN SET "	68	6	L	"
	(B31)				CRY LOGIC	36	6	L	93-1
					OVFLO F/F	15	13	L	"
					AC WRITE*	11	1Ø	H	"
					ARØ-AR3	57	6	L	93-2
					AR4-AR7	52	6	L	"
					AR8-AR11	47	6	L	93-3
					AR12-AR15	42	6	L	"
CPU INST	65	3	H	92-4	CON DATA*	29	13	H	92-4
					INTA	25	1	H	"
					IORST	6	4	H	"
					HALT	31	9	H	"
					(IO SKP LOGIC)	1Ø9	12	H	"
					"	1Ø9	1Ø	H	"
					MSKO	31	1	H	"
					[ION]	89	12	H	"
					ION*	92	2	H	"
					WRITE AC*	76	13	H	92-3
CR	1Ø4	15	H	92-4	GR-CG*	13	13	H	93-1
CR·CG*	13	12	L	93-1	CRY TO AR	38	12	L	93-1
					"	39	2	H	"
					OVFLO	15	2	H	"
					"	19	1	L	"
					OVFLO	16	1	L	"
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
CRY TO AR	39	6	H	93-1	CRY TEST*	4Ø	4	H	93-1
					"	4Ø	3	H	"
					SUM CRY	39	1Ø	H	"
					AR12	42	4	H	93-3
CRY	35	6,3	(OC)	93-1	CRY TO AR	38	4	L	93-1
					"	37	1	H	"
CRY*	32	8	L	93-1		(A59)			92-2
	(A59)				CARRY*	(A15)			92-2
CRY SET	35	11,8	(OC)	93-1	CRY*	36	14	H	93-1
CRY TEST*	4Ø	8	L	93-1		(A76)			
					PREV CRY	19	11	L	93-1
					FETCH SKIP*	4	3	H	92-3
12CR	89	15	H	93-3	AR8-11	94	3	H	93-3
12CG	89	14	H	"	"	94	4	H	"
8CR	94	15	H	"	AR4-7	99	3	H	93-2
8CG	94	14	H	"	"	99	4	H	"
4CR	99	15	H	93-2	ARØ-3	1Ø4	3	H	"
4CG	99	14	H	"	"	1Ø4	4	H	"
DATAØ*	16	11	(OC)	1Ø4-1	TERMINATOR				93-3
	17	1		"					
	(B62)								
DATA1*	16	8	(OC)	1Ø4-1	TERMINATOR				93-3
	17	3		"					
	(B65)								
DATA2*	14	11	(OC)	1Ø4-1	TERMINATOR				93-3
	15	1		"					
	(B82)								
DATA3*	14	8	(OC)	104-1	TERMINATOR				93-3
	15	3		"					
	(B73)								
DATA4*	12	11	(OC)	104-1	TERMINATOR				93-3
	13	1		"					
	(B61)								
DATA5*	12	8	(OC)	104-1	TERMINATOR				93-3
	13	3		"					
	(B57)								
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
DATA6*	1Ø 11 (B95)	11 1	(OC)	104-1 "	TERMINATOR				93-3
DATA7*	1Ø 11 (B55)	8 3	(OC)	104-1 "	TERMINATOR				93-3
DATA8*	8 9 (B6Ø)	11 1	(OC)	1Ø4-1 "	TERMINATOR				93-3
DATA9*	8 9 (B63)	8 3	(OC)	1Ø4-1 "	TERMINATOR				93-3
DATA1Ø*	6 7 (B75)	11 1	(OC)	1Ø4-1 "	TERMINATOR				93-3
DATA11*	6 7 (B58)	8 3	(OC)	1Ø4-1 "	TERMINATOR				93-3
DATA12*	4 5 (B59)	11 1	(OC)	1Ø4-1 "	TERMINATOR				93-3
DATA13*	4 5 (B64)	8 3	(OC)	1Ø4-1 "	TERMINATOR				93-3
DATA14*	2 3 (B56)	11 1	(OC)	1Ø4-1 "	TERMINATOR				93-3
DATA15*	2 3 (B66)	8 3	(OC)	1Ø4-1 "	TERMINATOR				93-3
D*	23	12	L	92-2	DEFER*		(A12)		92-2
					D	24	3	L	"
D	24	4	H	92-2	(States Logic)	83	6	H	92-2
						73	1	H	92-3
					MEM OUT*	44	2, 3	H	"
					MB LD EN*	1Ø8	13	H	"
					=Ø ENABLE	11Ø	13	H	"
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
DATIA	7	6	H	92-4		(A44)			
					CON DATA*	29	1	H	92-4
DATIB	25	8	H	"		(A42)			
					INTA	25	5	H	"
DATIC	8	6	H	"		(A54)			
					IORST	6	5	H	"
DATOA	27	6	H	"		(A58)			
DATOB	8	*	H	"		(A56)			
					MSKO*	31	5	H	92-4
DATOC	7	8	H	"		(A48)			
DCH	78	9	H	92-2	DCH*	77	9	H	92-2
					MEM OUT*	44	9	H	92-3
					IO OUT EN*	67	5	H	92-4
					DCHO	72	2,12	H	"
DCH*	77	8	L	92-2		(B51)			
					STATE SUPPRESS	79	4	L	92-2
					LOAD AR	1Ø5	5	H	92-3
DCHA*	74	8	L	92-4		(A6Ø)			
					DCHA*	71	9	L	92-4
DCHA STUTTER	7Ø	11	H	92-4	(CPU CLK LOGIC)	53	2	H	92-1
DCHI	15	13	H	92-4		(B37)			
DCH INC EN	69	11	H	92-4	DCHO	7Ø	1Ø	L	92-4
					FORCE PLUS ONE*	6Ø	9	H	92-3
					MB LD EN*	1Ø8	1	H	"
[DCH INC EN]	69	12	L	92-4	IO OUT EN*	67	4	H	92-4
DCHMØ*	(B17)		L	92-4	DCHI	51	9	L	92-4
					DCH INC EN	32	4	H	"
DCHM1*	(B21)		L	92-4	DCHI	32	1	H	"
					DCH INC EN	51	12	L	"
DCHO	15	6	H	92-4		(B33)			
					OVFLO	14	9	H	93-1
DCHP TEST	(A95)			92-4	FAST DCH	68	14	H	92-4
DCHR*	(B36)			92-4		7Ø	5	L	92-4
DCH SYNC	69	5	H	92-4	DCHA*	71	1	H	92-4
					DCH Stutter	35	1	H	"
*Indicates "NOT"					DCH INC EN	32	5	H	"
					"	32	2	H	"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
DCH SYNC (continued)					RUN SET	51	2	H	92-4
					DCH	78	15	H	92-2
					MEM CY SET	75	13	H	"
DEFER*	Back Plane		A12→		CONNECTOR	J4,	Pin	52 to →	
					(Indicator lamp)	12	5	L	89-1
D-H-READ	33	6	(OC)	93-1	ACDØ-3	74	4		93-2
					ACD4-7	7Ø	4		"
					ACD8-11	66	4		"
					ACD12-15	62	4		"
DIV LOAD CRY*				93-1	SUM CRY	39	9	H	93-1
					"	32	1	L	"
						13	3	L	"
					CRY SET	38	1Ø	L	"
DIV ADD*					CRY TO AR	37	9	H	93-1
					"	38	5	L	"
DIV FIRST*					CRY SET	16	2	L	93-1
D-L-H-SEL	13	8	H	93-1	D-MULT8-11	65	16	H	93-3
					D-MULT12-15	61	16	H	"
D-L-READ	33	3	(OC)	93-1	ACDØ-3	74	5		93-2
					ACD4-7	7Ø	5		"
					ACD8-11	66	5		93-3
					ACD12-15	62	5		"
D-L-SEL	11	6	H	93-1	D-MULTØ-3	73	17	H	93-2
					D-MULT4-7	69	17	H	"
					D-MULT8-11	65	17	H	93-3
					D-MULT12-15	61	17	H	"
D-M-COM	16	11	H	93-1	D-MULTØ-3	73	15	H	93-2
					D-MULT4-7	69	15	H	"
D-M-H-SEL	11	11	H	93-1	D-MULTØ-3	73	16	H	93-2
					D-MULT4-7	69	16	H	"
D-MULT Ø	73	1Ø		93-2	ADDER Ø	1Ø4	16		93-2
D-MULT 1	73	11		"	ADDER 1	1Ø4	18		"
D-MULT 2	73	14		"	ADDER 2	1Ø4	2Ø		"
D-MULT 3	73	13		"	ADDER 3	1Ø4	22		"
D-MULT 4	69	1Ø		"	ADDER 4	99	16		"
D-MULT 5	69	11		"	ADDER 5	99	18		"
D-MULT 6	69	14		"	ADDER 6	99	2Ø		"
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
D-MULT 7	69	13		93-2	ADDER 7	99	22		93-2
D-MULT 8	65	1Ø		93-3	ADDER 8	94	16		93-3
D-MULT 9	65	11		"	ADDER 9	94	18		"
D-MULT1Ø	65	14		"	ADDER1Ø	94	2Ø		"
D-MULT11	65	13		"	ADDER11	94	22		"
D-MULT12	61	1Ø		"	ADDER12	89	16		"
D-MULT13	61	11		"	ADDER13	89	18		"
D-MULT14	61	14		"	ADDER14	89	2Ø		"
D-MULT15	61	13		"	ADDER15	89	22		"
DRIVE IO*	(B88)			92-4	[DRIVE IO]	18	1	L	1Ø4-1
[DRIVE IO]	18	2	H	1Ø4-1	[Drive IO· Select]	26	9,1Ø	12 H	"
[DRIVE IO· Select]	26	8	H	"	DATAØ*	16	12	H	"
					DATA 1*	16	1Ø	H	"
					DATA 2*	14	12	H	"
					DATA 3*	14	1Ø	H	"
					DATA 4*	12	12	H	"
					DATA 5*	12	1Ø	H	"
					DATA 6*	1Ø	12	H	"
					DATA 7*	1Ø	1Ø	H	"
					DATA 8*	8	12	H	"
					DATA 9*	8	1Ø	H	1Ø3-1
					DATA1Ø*	6	12	H	"
					DATA11*	6	1Ø	H	"
					DATA12*	4	12	H	"
					DATA13*	4	1Ø	H	"
					DATA14*	2	12	H	"
					DATA15*	2	1Ø	H	"
DSZ	64	3	H	92-2	FORCE SX Com*	43 (A72)	1Ø	H	92-3
DSØ*	9	8	L	92-4		(A68)			
DS1*	9	1Ø	L	"		(A66)			
DS2*	9	12	L	"		(A46)			
DS3*	9	2	L	"		(A62)			
DS4*	9	4	L	"		(A64)			
DS5*	9	6	L	"					
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
E*	22	8	L	92-2	E	24	5	L	92-2
					EXECUTE*	(A11)			
					E·IO	75	5	L	92-2
					(MULT)	43	7	L	92-3
					INC PC	98	2	L	"
					INC PC	94	13	L	"
E	24	6	H	92-2	WRITE AC*	1Ø7	13	H	92-3
					MEM OUT*	44	1	H	"
					MB LD EN*	1Ø8	1Ø	H	"
EFA*	58	8	L	92-2		(A87)			
					EFA	77	3	L	92-2
					(MULT)	46	7	L	92-3
					D-L-H-SEL	13	9	L	93-1
					D-M-COM	16	12	L	"
EFA	77	4	H	92-2	(STATES Logic)	83	13	H	92-2
					EFA·JSR*	1Ø5	9	H	92-3
					=Ø ENABLE	11Ø	12	H	"
EFA·JSR*	1Ø5	8	L	92-3		(A92)			
					LOAD AR	1Ø5	4	H	92-3
					WRITE AC*	1Ø5	12	L	"
					AC-H-WRITE	37	4	L	93-1
E·IO	75	6	H	92-2	OMIT STROBE*	45	1Ø	H	92-3
					[E·IO*]	59	9	H	"
					(IO DECODE)	88	13	H	92-4
					(IO DECODE)	89	1Ø	H	"
					HALT	31	1Ø	H	"
					WRITE AC*	1Ø7	6	H	92-3
					IO Stutter*	88	4	H	92-4
					IO Stutter*	66	9	H	"
					IO OUT EN*	67	3	H	92-4
[E·IO*]	59	8	L	92-3	FORCE D-L-SEL*	62	1Ø	L	92-3
EXEC*	Back Plane		A-11	→	Connector (Indicator Lamp)	J4 Pin 12	19 9	to → L	89-1
EXT SELECT*	Back Plane		B8Ø →		SELECT	35	9	H	1Ø4-1
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
F*	23	8	L	92-2		(A47)			92-2
					FETCH*	(A13)		L	"
					F	24	13	L	"
					ALC*	41	4	L	"
					MBB8, 1Ø, 11	16	9	L	93-1
F	24	12	H	92-2	(MULT)	45	1,6	H	92-3
					LOAD PC*	93	5	H	"
					ION SYNC	92	9	H	92-4
FAST DCH	68	11	H	92-4	DCHO	72	4	H	92-4
					IO STUTTER*	87	1	H	"
FETCH SKIP*	1Ø1	8	L	92-3	INC PC	95	9	L	92-3
					CPU CLK	49	2	H	92-1
					" "	49	12	H	"
FETCH*	Back	Plane	A13→		Connector	J4,	Pin	50 to →	
					(Indicator Lamp)	12	11	L	89-1
◇FORCE AR Shift*				93-1	SHIFT AR Left	1Ø	3	L	93-1
FORCE D-L-SEL*	45	13	L	92-3		(A84)			
	61	8	(OC)	"	D-L-SEL	11	4	L	93-1
	43	3	L	"	FORCE-SX-H-SEL*	74	2	H	92-3
					MEM OUT*	44	13	H	"
					MEM OUT*	44	6	H	"
FORCE PLUS One*	45	4	L	92-3		(B49)			
	43	4	L	"	PLUS ONE	1Ø	1	L	93-1
FORCE MEM CY*	(B27)				READ CY	35	5	L	92-1
◇FORCE MQ OUT &AC-L-WRITE*					D-M-H-SEL	11	13	L	93-1
					D-L-H-SEL	13	11	L	"
					D-L-SEL	11	5	L	"
						13	4	L	"
FORCE SX COM*	63	3	(OC)	92-3		(A67)			
	43	12	L	"	SX-COM	1Ø	5	L	93-1
FORCE SX-H-SEL*	46	12	L	92-3		(A89)			
	45	3	L	"	SX-H-SEL	1Ø	9	L	93-1
◇MUL/DIV OPTION									
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
FORCE SX-H-READ*	46	3	H	92-3		(A86)			
					SX-H-READ	32	13	L	93-1
FORCE SX-L-READ*	46	4	L	92-3		(A88)			
					SX-L-READ	1Ø	13	L	93-1
FORCE SX-L-SEL*	46	13	L	92-3		(A91)			
					SX-L-SEL	1Ø	11		93-1
FORCE SEL X*	1Ø9	3	(OC)	92-3		(B67)			
					SEL-N	18	1Ø		93-1
					SEL-L	2Ø	5		"
					SEL-R	19	5		"
FORCE SEL Y*	1Ø9	6	(OC)	92-3		(B69)			
					SEL-N	18	13		93-1
					SEL-S	19	3		"
					SEL-R	18	8		"
F-SET	4Ø	12,13		92-2	F+PI SET*	75	1	H	92-2
					PC TO MEM*	1Ø6	4	H	92-3
					STOP RQ	6	12	H	92-2
F+PI	41	11	H	92-2	(IR LOGIC)	73	9	H	"
F+PI SET*	75	3	L	92-2	MEM CY SET	76	3	L	92-2
					ION SYNC	11Ø	9	L	92-4
HALT	9Ø	6	H	92-4	RUN SET	7Ø	1	H	92-4
					FORCE D-L-SEL*	62	9	L	92-3
HAS E CYCLE*	58	6	H	92-2	(STATES LOGIC)	42	4	L	92-2
					(STATES LOGIC)	42	12	L	"
INC PC	95	8	H	92-3		(A81)			93-1
					PCØ-3	59	7	H	93-2
					4-7	54	7	H	"
					8-11	49	7	H	93-3
					12-15	44	7	H	"
INH DCH	1Ø6	12	H	92-4		7Ø	4	L	92-4
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
INHØ	34	9		1Ø4-1	MEMØ*	16	1	H	1Ø4-1
					DATAØ*	16	13	H	"
INHØ*	34	8		1Ø4-1	(INHBØ) (Q15)	68	12	H	1Ø4-1
INH1	34	5		"	MEM1*	16	5	H	1Ø4-1
					DATA1*	16	9	H	1Ø4-1
INH1*	34	6		1Ø4-1	(INHB1) (Q16)	68	2	H	1Ø4-1
INH2	32	5		1Ø4-1	MEM2*	14	1	H	1Ø4-1
					DATA2*	14	13	H	"
INH2*	32	6		1Ø4-1	(INHB2) (Q13)	64	2	H	1Ø4-1
INH3	32	9		1Ø4-1	MEM3*	14	5	H	1Ø4-1
					DATA3*	14	9	H	1Ø4-1
INH3*	32	8		1Ø4-1	(INHB3) (Q14)	64	12	H	1Ø4-1
INH4	31	9		1Ø4-1	MEM4*	12	1	H	1Ø4-1
					DATA4*	12	13	H	"
INH4*	31	8		1Ø4-1	(INHB4) (Q11)	58	12	H	1Ø4-1
INH5	31	5		"	MEM5*	12	5	H	1Ø4-1
					DATA5*	12	9	H	"
INH5*	31	6		1Ø4-1	(INHB5) (Q12)	58	2	H	1Ø4-1
INH6	28	5		"	MEM6*	1Ø	1	H	1Ø4-1
					DATA6*	1Ø	13	H	"
INH6*	28	6		1Ø4-1	(INHB6) (Q9)	55	2	H	1Ø4-1
INH7	28	9		"	MEM7*	1Ø	5	H	1Ø4-1
					DATA7*	1Ø	9	H	"
INH7*	28	8		1Ø4-1	(INHB7) (Q1Ø)	55	12	H	1Ø4-1
INH8	27	9		"	MEM8*	8	1	H	1Ø4-1
					DATA8*	8	13	H	1Ø4-1
INH8*	27	8		1Ø4-1	(INHB8) (Q7)	48	12	H	1Ø4-1
INH9	27	5		"	MEM9*	8	5	H	1Ø4-1
					DATA9*	8	9	H	"
INH9*	27	6		1Ø4-1	(INHB9) (Q8)	48	2	H	1Ø4-1
INH1Ø	24	5		"	MEM1Ø*	6	1	H	1Ø4-1
					DATA1Ø*	6	13	H	1Ø4-1
INH1Ø*	24	6		1Ø4-1	(INHB1Ø) (Q5)	45	2	H	1Ø4-1
INH11	24	9		"	MEM11*	6	5	H	1Ø4-1
					DATA11*	6	9	H	1Ø4-1
INH11*	24	8		1Ø4-1	(INHB11) (Q6)	45	12	H	1Ø4-1

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
INH12	23	9		1Ø4-1	MEM12*	4	1	H	1Ø4-1
					DATA12*	4	13	H	"
INH12*	23	8		1Ø4-1	(INHB12) (Q3)	39	12	H	1Ø4-1
INH13	23	5		"	MEM13*	4	5	H	1Ø4-1
					DATA13*	4	9	H	1Ø4-1
INH13*	23	6		1Ø4-1	(INHB13) (Q4)	39	2	H	1Ø4-1
INH14	21	5		"	MEM14*	2	1	H	1Ø4-1
					DATA14*	2	13	H	1Ø4-1
INH14*	21	6		1Ø4-1	(INHB14)	37	2	H	1Ø4-1
INH15	21	9		1Ø4-1	MEM15*	2	5	H	1Ø4-1
					DATA15*	2	9	H	"
INH15*	21	8		1Ø4-1	(INHB15) (Q2)	37	12	H	1Ø4-1
INHBØ	7Ø	3		1Ø4-2	Q15				
INHB1	7Ø	5		"	Q16				
INHB2	63	3		"	Q13				
INHB3	63	5		"	Q14				
INHB4	61	3		"	Q11				
INHB5	61	5		"	Q12				
INHB6	53	3		1Ø4-1	Q9				
INHB7	53	5		1Ø4-1	Q1Ø				
INHB8	51	3		"	Q7				
INHB9	51	5		"	Q8				
INHB1Ø	43	3		"	Q5				
INHB11	43	5		"	Q6				
INHB12	42	3		"	Q3				
INHB13	42	5		"	Q4				
INHB14	2Ø	3		"	Q1				
INHB15	2Ø	5		"	Q2				
INH GATE A	26	6	H	1Ø4-1	(INHBØ)(Q15)	68	13	H	1Ø4-1
					(INHB1) (Q16)	68	1	H	"
					(INHB2) (Q13)	64	1	H	"
					(INHB3) (Q14)	64	13	H	"
					(INHB4) (Q11)	58	13	H	1Ø4-1
					(INHB5) (Q12)	58	1	H	"
					(INHB6) (Q9)	55	1	H	"
					(INHB7) (Q1Ø)	55	13	H	"

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
INH GATE B	26	6	H	1Ø4-1	(INHB8) (Q7)	48	13	H	1Ø4-1
					(INHB9) (Q8)	48	1	H	"
					(INHB1Ø) (Q5)	45	1	H	"
					(INHB11) (Q6)	45	13	H	"
					(INHB12) (Q3)	39	13	H	"
					(INHB13) (Q4)	39	1	H	"
					(INHB14) (Q1)	37	1	H	"
					(INHB15) (Q2)	37	13	H	"
INHIBIT	17	5	H	92-1		B3Ø)			1Ø4-1
					INH GATE A, B	41	9	H	"
					WRITE MEM	41	2	H	"
					WRITE F/F	17	11	H	92-1
INHIBIT*	17	6	L	92-1					
INHIBIT SELECT*	(B85)				SELECT	35	5	L	1Ø4-1
INH TRANS*	3Ø	3, 11, 8	L	92-4		B45)		L	1Ø4-1
					[INH TRANS· SEL]*	36	2, 5, 4	L	"
[INH TRANS· SEL]	36	6	H	1Ø4-1	MEMØ*	16	2	H	1Ø4-1
					MEM1*	16	4	H	"
					MEM2*	14	2	H	"
					MEM3*	14	4	H	"
					MEM4*	12	2	H	"
					MEM5*	12	4	H	"
					MEM6*	1Ø	2	H	"
					MEM7*	1Ø	4	H	"
					MEM8*	8	2	H	"
					MEM9*	8	4	H	"
					MEM1Ø*	6	2	H	"
					MEM11*	6	4	H	"
					MEM12*	4	2	H	"
					MEM13*	4	4	H	"
					MEM14*	2	2	H	"
					MEM15*	2	4	H	"
INTA	25	6	H	92-4		(A4Ø)			
INTR*	(B29)				INTRQ*	89	1	L	92-4
INTRQ*	88	8	H	92-4		89	4	L	92-4
						42	13	L	"
ION	91	12	H	92-4	ION*	91	11	H	92-4
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
ION*	91	8	L	92-4		(A16)			
					ION	91	13	L	92-4
					INTRQ*	9Ø	12	L	"
ION*	Back	Panel	A16→		Connector	J4,	Pin	25to→	
					(Indicator Lamp)	12	3	L	89-1
ION SYNC	11Ø	8	H	92-4		(B53)			
IO OUT EN*	67	6	L	92-4	IO STUTTER*	9Ø	1	L	92-4
					DCHA STUTTER	7Ø	13	L	"
					DCHA*	71	2	H	"
IOPLS	27	8	H	92-4		(A74)			
IRØ	14	1	H	92-2	ALC*	56	9	H	92-2
					FORCE D-L-SEL*	62	4	L	92-3
IRØ*	14	16	L	92-2	IO UNPROTECTED*	22	2	H	92-2
					STA	47	2	H	"
					LDA	47	4	H	"
					EFA*	58	9	H	"
					STOP RQ	8Ø	1	H	"
IR1	14	14	H	92-2	IO UNPROTECTED*	22	1	H	92-2
					STA	47	1	H	"
					FORCE-SX-H-READ*	46	2	H	92-3
					LOAD PC*	95	4	H	"
IR1*	14	15	L	92-2	LDA	47	5	H	92-2
					FORCE-D-L-SEL*	62	5	L	92-3
					WRITE AC*	1Ø7	1Ø	H	"
IR2	14	8	H	92-2	NON ACD INST	65	9	L	92-2
					STA	65	4	L	"
					IO UNPROTECTED*	22	13	H	"
					FORCE-SX-L-READ*	46	5	H	92-3
					OMIT STROBE*	56	2	L	"
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
IR2*	14	9	L	92-2	LDA	65	13	L	92-2
IR3	14	11	H	92-2	DSZ/ISZ	47	13	H	92-2
IR3*	14	1Ø	L	92-2	JMP+JSR	47	9	H	92-2
	(B15)				D-H-READ	33	5	L	93-1
					AC-H-WRITE	37	5	L	"
IR4	13	1	H	92-2	DSZ	64	1	L	92-2
					ISZ	64	5	L	"
IR4*	13	16	L	92-2	JSR	64	1Ø	L	"
	(B11)				D-L-READ	33	1	L	93-1
					AC-L-WRITE	12	9	L	"
IR5	13	14	H	92-2	AND ENABLE*	28	13	H	92-2
					STATES LOGIC	83	1	H	"
					FORCE-D-L-SEL*	43	2	H	92-3
					LOAD PC*	97	5	L	"
					DATIC	8	4	H	92-4
					DATOB	8	9	H	"
					DATOC	7	9	H	"
					HALT	31	12	H	"
					WRITE AC*	76	1	H	92-3
IR5*	13	15	L	92-2	DATIA	7	1	H	92-4
					DATIB	25	12	H	"
					DATOA	27	4	H	"
IR6	13	8	H	92-2	AND ENABLE*	28	2	H	92-2
					FORCE-SX-H-SEL*	62	12	L	92-3
					FORCE-SX-L-SEL*	46	15	H	"
					HALT	31	13	H	92-4
					DATIA	7	4	H	"
					DATIC	8	2	H	"
					DATOB	8	1Ø	H	"
					DATIB	25	13	H	"
					DATOA	27	2	H	92-4
					DATOC	7	13	H	"

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
IR6*	13	9	L	92-2	KEYM	97	12	H	92-2
					FORCE SX COM*	43	11	H	92-3
IR7	13	11	H	92-2	AND ENABLE*	28	1	H	92-2
					FORCE-SX-L-READ*	46	6	H	92-3
					FORCE PLUS ONE*	43	5	H	"
					READ IO	88	2	H	92-4
					HALT	90	4	H	"
IR7*	13	1Ø	L	92-2	FORCE-SX-H-SEL*	62	13	L	92-3
					WRITE AC*	76	9,10	H	"
					INC PC	96	5	H	"
					IO OUT EN*	67	2	H	92-4
IR8	1Ø	1	H	92-2	FORCE SEL X*	1Ø9	1	H	92-3
					CLR	26	4	H	92-4
					IOPLS	27	9	H	"
					IO SKIP PEND	86	2	H	"
IR8*	1Ø	16	L	92-2	IO STUTTER*	87	5	L	92-4
					STRT	26	9	H	"
					IO SKIP PEND	86	4	H	"
IR9	1Ø	14	H	92-2	FORCE SEL Y*	1Ø9	5	H	92-3
					STRT	26	13	H	92-4
					IOPLS	27	1Ø	H	"
					IO SKIP PEND	86	13	H	"
IR9*	1Ø	15	L	92-2	IO STUTTER*	87	4	L	92-4
					CLR	26	5	H	"
					IO SKIP PEND	86	1Ø	H	"
IR1Ø	1Ø	8	H	92-2	DSØ*	9	9	H	92-4
					CPU INST	28	11	H	"
IR1Ø*	1Ø	9	L	92-2					
IR11	1Ø	11	H	92-2	DS1*	9	11	H	92-4
					CPU INST	28	1Ø	H	"
IR11*	1Ø	1Ø	L	92-2					
IR12	11	1	H	92-2	DS2*	9	13	H	92-4
					CPU INST	28	9	H	"
IR12*	11	16	L	92-2	LOAD CRY*	1Ø6	11	H	92-3

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
IR13	11	14	H	92-2	FETCH SKIP*	4	5	H	92-3
					DS3*	9	1	H	92-4
					CPU INST	28	5	H	"
IR13*	11	15	L	92-2					
IR14	11	8	H	92-2	FETCH SKIP*	4	2	H	92-3
					DS4*	9	3	H	92-4
					CPU INST	28	4	H	"
IR14*	11	9	L	92-2					
IR15	11	11	H	92-2	FETCH SKIP*	99	9	H	92-3
					DS5*	9	5	H	92-4
					CPU INST	28	3	H	"
IR15*	11	10	L	92-2		99	1	H	92-3
IORST	5	6	H	92-4		(A70)			
IO SKIP	57	3	H	92-2	HAS E CYCLE*	57	13	H	92-2
					FETCH SKIP*	101	13	H	92-3
					IO STUTTER*	85	2	H	92-4
IO SKIP PEND	86	8	L	92-4	IO SKIP SYNC	68	15	H	92-4
IO SKIP SYNC	68	9	H	92-4	FETCH SKIP*	101	10	H	92-3
IO STUTTER*	85	8	L	92-4	CPU CLK	53	1	L	92-1
						48	1	L	"
IO UNPROTECTED*	22	12	L	92-2		(A61)			
					REAL IO INST	75	9	L	92-2
					MEM CY SET	75	12	H	"
					EFA*	58	10	H	"
ISTP*	Connector Back Plane		J4 → A17→		Back Plane		A17	L	89-1
						4	1	L	92-2
ISZ	64	6	H	92-2		(A61)			
JMP+JSR	56	6	H	92-2	PC TO MEM*	110	5	H	92-3
					PC TO MEM*	99	4	H	"
					HAS E CYCLE*	57	9	H	92-2
					LOAD PC*	93	1	H	92-3
JSR	64	8	H	92-2	EFA·JSR*	105	10	H	92-3
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
KEY	78	5	H	92-2	KEY*	77	13	H	92-2
					STOP RQ	98	10	L	"
					KEY· PRESET*	97	2	H	"
					LOAD AR	60	2	H	92-3
					PC TO MEM*	99	2	H	"
					CON DATA*	29	2	H	92-4
KEY*	77	12	L	92-2	STATE SUPPRESS	79	2	L	92-2
					RQENB*	79	12	H	"
					(STOP RQ LOGIC)	6	13	H	"
					PRESET*	73	3	H	"
					INH DCH	106	13	L	92-4
KEYM	78	11	H	92-2	STOP RQ	80	10	H	92-2
					OMIT STROBE*	61	5	H	92-3
						95	12	H	"
KEYM*	78	12	L	92-2	STATE SUPPRESS	79	5	L	92-2
					INH DCH	106	1	L	92-4
KEYM· PL*	95	11	L	92-3	ROM ENABLE*	90	10	L	92-3
					PL LAST	110	1	L	"
					PL· LAST WORD*	38	4	L	"
					ADDER TO MEM*	103	3,11	H	"
KEY SEEN + RESTART	81	3	H	92-2	LOAD PC*	94	9	H	92-3
					INH TRANS*	30	13	H	92-4
KEY· PRESET*	100	2	H	92-2	KEYM	97	13	H	92-2
					STOP RQ	80	13	H	"
					OMIT STROBE*	61	12, 13	H	92-3
					FORCE D-L-SEL*	61	9	H	"
					WRITE AC*	107	9	H	"
					MEM OUT*	44	5	H	"
					INC PC	96	6	H	"
					LOAD PC*	85	5	H	"

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
LOAD AR	6Ø	3,11, 6	OC	92-3		(A83)			93-1
					ARØ-3	57	1Ø	H	93-2
					AR4-7	52	1Ø	H	"
					AR8-11	47	1Ø	H	93-3
					AR12-15	42	1Ø	H	"
LOAD PC*	93	8	L	92-3		(A79)			93-1
					PCØ-3	59	9	L	93-2
					PC4-7	54	9		
					PC8-11	49	9	L	93-3
					PC11-15	44	9	L	"
LDA	65	11	H	92-2	WRITE AC*	1Ø7	1	H	92-3
LOAD CRY*	1Ø6	8	L	92-3		(B38)			
					WRITE AC*	1Ø5	13	L	92-3
						38	9	L	93-1
MA1	33	15		1Ø4-1	[SARD1] (Jumper)	35	4		1Ø4-1
MA1*	33	14		1Ø4-1	[SARD1] "	35	4		1Ø4-1
MA2	33	1Ø		"	[SARD2] "	35	1		"
MA2*	33	11		"	[SARD2] "	35	1		"
MA3	33	9		"	[SARD2] "	35	2		"
MA3*	33	8		"	[SARD3] "	35	2		"
MA4	29	16		"	MA4 B*	67	3	H	1Ø4-4
MA4*	29	1		"					
MA4 B*	67	4		1Ø4-4	MA4B	67	11	L	1Ø4-4
					Y ADDR DCDR	52	5,4		"
					"	66	5,4		"
MA4B	67	1Ø		1Ø4-4	"	54	5,4		"
					"	62	5,4		"
MA5	29	15		1Ø4-1					
MA5*	29	14		"	MA5B	67	5	H	"
MA5 B	67	6		1Ø4-4	MA5 B*	67	9	L	"
					Y ADDR DCDR	54	7		"
					"	62	7		"
					"	52	7		"
					"	66	7		"

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
MA5 B*	67	8	H	1Ø4-4	Y ADDR DCDR	54	1		1Ø4-4
					"	62	1		"
					"	52	1		"
					"	66	1		"
MA6	29	1Ø	H	1Ø4-1	MA6 B*	67	1	H	"
MA6*	29	11	L	"					
MA6 B*	67	2	L	1Ø4-4	MA6 B	67	13	L	1Ø4-4
					Y ADDR DCDR	62	6		"
					"	66	6		"
MA6 B	67	12	H	1Ø4-4	"	54	6		"
					"	52	6		"
MA7	29	9	H	1Ø4-1	MA7 B*	44	11	H	"
MA7*	29	8	L	"					
MA7 B*	44	1Ø	L	1Ø4-4	MA7 B	44	3	L	1Ø4-4
					Y ADDR DCDR	6Ø	5,4		"
					"	5Ø	5,4		"
MA7 B	44	4	H	1Ø4-4	"	57	5,4		
					"	47	5,4		
MA8	25	16	H	1Ø4-1	MA8 B*	44	9	H	1Ø4-4
MA8*	25	1	L	"					
MA8 B*	44	8	L	1Ø4-4	MA8 B	44	5	L	1Ø4-4
					Y ADDR DCDR	6Ø	7		"
					"	5Ø	7		"
					"	57	7		"
					"	47	7		"
MA8 B	44	6	H	1Ø4-4	"	6Ø	1		"
					"	5Ø	1		"
					"	57	1		"
					"	47	1		"
MA9	25	15	H	1Ø4-1	MA9 B*	44	13	H	"
MA9*	25	14	L	"					
MA9 B*	44	12	L	1Ø4-4	MA9 B	44	1	L	1Ø4-4
					Y ADDR DCDR	6Ø	6		"
					"	57	6		"
MA9 B	44	2	H	1Ø4-4	"	5Ø	6		"

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
MA1Ø	25	1Ø	H	1Ø4-1	MA1Ø B*	71	3	H	1Ø4-4
MA1Ø*	25	11	L	"					
MA1Ø B*	71	4	L	1Ø4-3	MA1ØB	71	11	L	1Ø4-3
					X ADDR DCDR	73	5,4		"
					"	77	5,4		"
MA1Ø B	71	1Ø	H	1Ø4-3	"	72	5,4		"
					"	76	5,4		"
MA11	25	9	H	1Ø4-1					
MA11*	25	8	L	"	MA11B	71	5	H	1Ø4-3
MA11 B	71	6	L	1Ø4-3	MA11 B*	71	9		"
					X ADDR DCDR	72	7		"
					"	76	7		"
					"	73	7		"
					"	77	7		"
MA11 B*	71	8	H	1Ø4-3	"	72	1		"
					"	76	1		"
					"	73	1		"
					"	77	1		"
MA12	22	16	H	1Ø4-1	MA12 B*	71	1	H	"
MA12*	22	1	L	"					
MA12 B*	71	2	L	1Ø4-3	MA12 B	71	13	L	1Ø4-3
					X ADDR DCDR	76	6		"
					"	77	6		"
MA12 B	71	12	H	1Ø4-3	"	72	6		"
					"	73	6		"
MA13	22	15	H	1Ø4-1	MA13 B*	8Ø	11	H	"
MA13*	22	14	L	"					
MA13 B*	8Ø	1Ø	L	1Ø4-3	MA13 B	8Ø	3	L	1Ø4-3
					X ADDR DCDR	79	5,4		"
					"	74	5,4		"
MA13 B	8Ø	4	H	1Ø4-3	X ADDR DCDR	78	5,4		"
					"	75	5,4		"
MA14	22	1Ø	H	1Ø4-1	MA14 B*	8Ø	9	H	"
MA14*	22	11	L	"					
MA14 B*	8Ø	8	L	1Ø4-3	MA14 B	8Ø	5	L	"
					X ADDR DCDR	79	7		"
					"	74	7		"
					"	78	7		"

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
MA14 B* (Cont'd)					X ADDR DCDR	75	7		1Ø4-3
MA14 B	8Ø	6	H	1Ø4-3	"	79	1		"
					"	74	1		"
					"	78	1		"
					"	75	1		"
MA15	22	9	H	1Ø4-1	MA15 B*	84	13	H	"
MA15*	22	8	L	"					
MA15 B*	8Ø	12	L	1Ø4-3	MA15 B	8Ø	1	L	"
					X ADDR DCDR	79	6		"
					"	78	6		"
MA15 B	8Ø	2	H	1Ø4-3	"	74	6		"
					"	75	6		"
MA LOAD*	52	8	L	92-1		(B7)		L	
					[MA LOAD]	3Ø	9,1Ø	L	1Ø4-1
					"	3Ø	12, 13	L	"
[MA LOAD]	3Ø	8	L	1Ø4-1	MA1-3	33	13	L	1Ø4-1
						33	4	L	"
					MA4-7	29	13	L	"
						29	4	L	"
					MA8-11	25	13	L	"
						25	4	L	"
					MA12-15	22	13	L	"
						22	4	L	"
MBB8*	17	16							
MBB1Ø*	17	15			(CRY LOGIC)	35	1		93-1
MBB11*	17	8			(CRY LOGIC)	35	4		93-1
MB CLR*	19	6	L	92-1		(B86)			
					[MB CLEAR]	18	9	L	1Ø4-1
[MB CLEAR]	18	8	H	1Ø4-1		3Ø	2,4, 5	H	1Ø4-1
[MB CLEAR· SEL]	3Ø	6	L	"	INHØ F/F	34	13	L	"
					INH1 F/F	34	1	L	"
					INH2 F/F	32	1	L	"
					INH3 F/F	32	13	L	"
					INH4 F/F	31	13	L	"
					INH5 F/F	31	1	L	"
					INH6 F/F	28	1	L	"
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
[MB CLEAR· SEL] Cont'd					INH7 F/F	28	13	L	1Ø4-1
					INH8 F/F	27	13	L	"
					INH9 F/F	27	1	L	"
					INH1Ø F/F	24	1	L	"
					INH11 F/F	24	13	L	"
					INH12 F/F	23	13	L	"
					INH13 F/F	23	1	L	"
					INH14 F/F	21	1	L	"
					INH15 F/F	21	13	L	"
MB LDEN*	1Ø8	3	(OC)	92-3	MB LOAD	35	1Ø	L	92-1
	1Ø8	8	(OC)	"	CPU CLOCK	36	1	L	"
	1Ø8	11	(OC)	"					
MB LOAD	18	6	H	92-1		(B74)		H	
						36	9	H	1Ø4-1
[MB LOAD· SEL]	36	8	H	1Ø4-1	INHØ F/F	34	11	H	"
					INH1 F/F	34	3	H	"
					INH2 F/F	32	3	H	"
					INH3 F/F	32	11	H	"
					INH4 F/F	31	11	H	"
					INH5 F/F	31	3	H	"
					INH6 F/F	28	3	H	"
					INH7 F/F	28	11	H	"
					INH8 F/F	27	11	H	"
					INH9 F/F	27	3	H	"
					INH1Ø F/F	24	3	H	"
					INH11 F/F	24	11	H	"
					INH12 F/F	23	11	H	"
					INH13 F/F	23	3	H	"
					INH14 F/F	21	3	H	"
					INH15 F/F	21	11	H	"
MBO INH*	1Ø8	6	(OC)	92-3		(B48)			
					ADDER TO MEM*	1Ø3	13	H	92-3
					PC TO MEM*	1Ø4	2	H	"
					D-M-H-SEL	11	12	L	93-1
					D-L-H-SEL	13	1Ø	L	"
MBOØ*	6Ø	3		93-2		(B79)			
					[MDØ]	17	9		1Ø4-1

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 8 00

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
MBO1*	6Ø	4		93-2		(B77)			
					MD1 (CON IND)	17	5		1Ø4-1
					(P14)	7	11		89-1
MBO2*	6Ø	12		93-2		(B44)			
					MD2 (CON IND)	15	9		1Ø4-1
					(P15)	7	5		89-1
MBO3*	6Ø	13		93-2		(B43)			
					MD3 (CON IND)	15	5		1Ø4-1
					(P38)	8	9		89-1
MBO4*	55	3		93-2		(B42)			
					MD4 (CON IND)	13	9		1Ø4-1
					(P16)	8	11		89-1
MBO5*	55	4		93-2		(B32)			
					MD5 (CON IND)	13	5		1Ø4-1
					(P11)	8	5		89-1
MBO6*	55	12		93-2		(B16)			
					MD6 (CON IND)	11	9		1Ø4-1
					(P35)	9	9		89-1
MBO7*	55	13		93-2		(B14)			
					MD7 (CON IND)	11	5		1Ø4-1
					(P9)	9	11		89-1
MBO8*	5Ø	3		93-3		(B12)			
					MD8 (CON IND)	9	9		1Ø4-1
					(P18)	9	5		89-1
MBO9*	5Ø	4		93-3		(B9)			
					MD9 (CON IND)	9	5		1Ø4-1
					(P8)	1Ø	9		89-1
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 8 00

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
MBO1Ø*	5Ø	12		93-3		(B8)			
					MD1Ø	7	9		1Ø4-1
					(CON IND)				
					(P44)	1Ø	11		89-1
MBO11*	5Ø	13		93-3		(B5)			
					MD11	7	5		1Ø4-1
					(CON IND)				
					(P6)	1Ø	5		89-1
MBO12*	45	3		93-3		(A39)			
					MD12	5	9		1Ø4-1
					(CON IND)				
					(P3Ø)	11	9		89-1
MBO13*	45	4		93-3		(A37)			
					MD13	5	5		1Ø4-1
					(CON IND)				
					(P4)	11	11		89-1
MBO14*	45	12		93-3		(A43)			
					MD14	3	9		1Ø4-1
					(CON IND)				
					(P12)	11	5		89-1
MBO15*	45	13		93-3		(A41)			
					MD15	3	5		1Ø4-1
					(CON IND)				
					(P28)	11	1		89-1
[MDØ]	17	8		1Ø4-1	INHØ	34	12		1Ø4-1
MD1	17	6		"	INH1	34	2		"
					MA1	33	3		"
MD2	15	8		"	INH2	32	2		"
					MA2	33	6		"
MD3	15	6		"	INH3	32	12		"
					MA3	33	7		"
MD4	13	8		"	INH4	31	12		"
					MA4	29	2		"
MD5	13	6		"	INH5	31	2		"
					MA5	39	3		"
MD6	11	8		"	INH6	28	2		"
					MA6	29	6		"
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
MD7	11	6		1Ø4-1	INH7	28	12		1Ø4-1
					MA7	29	7		"
MD8	9	8		"	INH8	27	12		"
					MA8	25	2		"
MD9	9	6		"	INH9	27	2		"
					MA9	25	3		"
MD1Ø	7	8		"	INH1Ø	24	2		"
					MA1Ø	25	6		"
MD11	7	6		"	INH11	24	12		"
					MA11	25	7		"
MD12	5	8		"	INH12	23	12		"
					MA12	22	2		"
MD13	5	6		"	INH13	23	2		"
					MA13	22	3		"
MD14	3	8		"	INH14	21	2		"
					MA14	22	6		"
MD15	3	6		"	INH15	21	12		"
MBØ	56	1		93-2	D-MULTØ	73	5		93-2
MBØ*	56	16		"					
MB1	56	14		"	D-MULT1	73	2		"
MB1*	56	15		"					
MB2	56	8		"	D-MULT2	73	19		"
MB2*	56	9		"					
MB3	56	11		"	D-MULT3	73	22		"
MB3*	56	1Ø		"					
MB4	51	1		"	D-MULT4	69	5		93-2
MB4*	51	16		"					
MB5	51	14		"	D-MULT5	69	2		93-2
MB5*	51	15		"					
MB6	51	8		"	D-MULT6	69	19		93-2
MB6*	51	9		"	D-M-COM	34	13	L	93-1
MB7	51	11		"	D-MULT7	69	22		93-2
MB7*	51	1Ø		"	D-M-COM	34	12	L	93-1
MB8	46	1		93-3	D-MULT8	65	5		93-3
					D-M-COM	34	9	H	93-1
MB8*	46	16		"					
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
MB9	46	14		93-3	D-MULT9	65	2		93-3
MB9*	46	15		"					
MB1Ø	46	8		"	D-MULT1Ø	65	19		"
MB1Ø*	46	9		"					
MB11	46	11		"	D-MULT11	65	22		"
MB11*	46	1Ø		"					
MB12	41			"	D-MULT12	61	5		"
MB12*	41			"					
MB13	41			"	D-MULT13	61	2		"
MB13*	41			"					
MB14	41			"	D-MULT14	61	19		"
MB14*	41			"					
MB15	41			"	D-MULT15	61	22		93-3
MB15*	41			"					
MEM CLK	18	8		92-1		(B4Ø)			
					READ CY	2Ø	13	L	92-1
					READ 2	2Ø	13	L	"
					STROBE	16	13	L	"
					WRITE SYNC	16	13	L	"
					INHIBIT	17	13	L	"
					WRITE	17	13	L	"
MEM CY SET	76	6	H	92-2	READ CY	52	4	H	92-1
					MA LOAD*	87	13	H	"
MEM LATCH	14	6	H	93-1	MBØ-3	56	13,4	L	93-2
					MB4-7	51	13,4	L	"
					MB8-11	46	13,4	L	93-3
					MB12-15	41	13,4	L	"
MEM OK	Connector Back	P2, Pin Plane	A9→	24→	Back Plane		A9	L	91-1
					STOP RQ	U5	9	H	92-2
MEM OUT*	44	8	L	92-3		(A69)			
						74	4	H	92-3
MEMØ*	16	3	(OC)	1Ø4-1		(B71)			
(ACEX + ACDP)	1	3	(OC)	89-1	(CON IND)				
					(P39)	7	9	L	89-1
					IRØ	14	2		92-2
					MBØ	56	2		93-2
					ALC+IO SET	21	1	L	92-2

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
					ALC+IO SET	21	1Ø	L	92-2
					(STATES LOGIC)	84	3	L	"
MEM1*	16	6	(OC)	1Ø4-1		(B7Ø)			
(ACDP)	1	6	(OC)	89-1	(CON IND)				
					(P41)	7	13	L	89-1
					IR1	14	3		92-2
					MB1	56	3		93-2
					ALC+IO SET	21	13	L	92-2
MEM2*	14	3	(OC)	1Ø4-1		(B47)			
(DP+DPN)	2	8	(OC)	89-1	(CON IND)				
					(P13)	7	3	L	89-1
					IR2	14	7		92-2
					MB2	56	7		93-2
					ALC+IO SET	21	9	L	92-2
MEM3*	14	6	(OC)	1Ø4-1		(B68)			
(ACEX+ACDP)	1	11	(OC)	89-1	(CON IND)				
					(P43)	7	1	L	89-1
					IR3	14	6		92-2
					MB3	56	6		93-2
MEM4*	12	3	(OC)	1Ø4-1		(B28)			
(ACEX+ACDP)	1	8	(OC)	89-1	(CON IND)				
					(P37)	8	13	L	89-1
					IR4	13	2		92-2
					MB5	51	2		93-2
MEM5*	12	6	(OC)	1Ø4-1		(B26)			
(EX+STRT+ACDP)	2	11	(OC)	89-1	(CON IND)				
					(P36)	8	3	L	89-1
					IR5	13	3		92-2
					MB5	51	3		93-2
MEM6*	1Ø	3	(OC)	1Ø4-1		(B22)			
(EX+EXN+DP+DPN +PL)	2	3	(OC)	89-1	(CON IND)				
					(P1Ø)	8	1	L	89-1
					IR6	13	7		92-2
					MB6	51	7		93-2
MEM7*	1Ø	6	(OC)	1Ø4-1		(B24)			
(EXN+DPN)	2	6	(OC)	89-1	(CON IND)				
					(P42)	9	13	L	89-1

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 8 00

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
					IR7	13	6		92-2
					MB7	51	6		93-2
MEM8*	8	3	(OC)	1Ø4-1		(A55)			
					(CON IND)				
					(P34)	9	3	L	89-1
					IR8	1Ø	2		92-2
					MB8	46	2		93-3
					MBB8*	17	2	H	93-1
MEM9*	8	6	(OC)	1Ø4-1		(A53)			
					(CON IND)				
					(P7)	9	1	L	89-1
					IR9	1Ø	3		92-2
					MB9	46	3		93-3
MEM1Ø	6	3	(OC)	1Ø4-1		(A45)			
					(CON IND)				
					(P32)	1Ø	13	L	89-1
					IR1Ø	1Ø	7		92-2
					MB1Ø	46	7		93-3
					MBB1Ø*	17	3	H	93-1
MEM11*	6	6	(OC)	1Ø4-1		(A51)			
					(CON IND)				
					(P31)	1Ø	3	L	89-1
					IR11	1Ø	6		92-2
					MB11	46	6		93-3
					MBB11	17	7	H	93-1
MEM12*	4	3	(OC)	1Ø4-1		(A36)			
					(CON IND)				
					(P5)	1Ø	1	L	89-1
					IR12	11	2		92-2
					MB12	41	2		93-3
MEM13*	4	6	(OC)	1Ø4-1		(A35)			
					(CON IND)				
					(P29)	11	13	L	89-1
					IR13	11	3		92-2
					MB13	41	3		93-3

* Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
MEM14*	2	3	(OC)	1Ø4-1		(B76)			
					(CON IND)				
					(P3)	11	3	L	89-1
					IR14	11	7		92-2
					MB14	41	7		93-3
MEM15*	2	6	(OC)	1Ø4-1		(B18)			
					(CON IND)				
					(P2)	12	13	L	89-1
					IR15	11	6		92-2
					MB15	41	6		93-3
MID	54	5	H	92-1	(IO INST DE-				
					CODE)	89	9	H	92-4
					DCHO	72	13	H	"
MQØ					D-MULTØ	73	4		93-2
MQ1					D-MULT1	73	1		"
MQ2					D-MULT2	73	2Ø		"
MQ3					D-MULT3	73	23		"
MQ4					D-MULT4	69	4		"
MQ5					D-MULT5	69	1		"
MQ6					D-MULT6	69	2Ø		"
MQ7					D-MULT7	69	23		"
MQ8					D-MULT8	65	4		93-3
MQ9					D-MULT9	65	2		"
MQ1Ø					D-MULT1Ø	65	2Ø		"
MQ11					D-MULT11	65	23		"
MQ12					D-MULT12	61	4		"
MQ13					D-MULT13	61	1		"
MQ14					D-MULT14	61	2Ø		"
MQ15					D-MULT15	61	23		"
MUL+DIV DE-									
CODE*	(A65)				MUL+DIV DE-				
					CODE	77	1	L	92-2
					STOP RQ	8Ø	2,4	H	"
MUL+DIV DE-									
CODE	77	2	H	92-2	HAS E CYCLE*	57	1Ø	H	92-2
					REAL IO INST	75	1Ø	H	"
					MEM CY SET	76	5	L	"
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
MSKO*	31	6	L	92-4		(A38)			
MSTP*	Connector		J4, Pin	48→	Back Plane	A2Ø		L	89-1
	Back	Plane	A2Ø→			4	1Ø	L	92-2
						4	13	L	92-2
NON ACD INST	65	8	H	92-2	DSZ/ISZ	47	12	H	92-2
					JMP+JSR	47	1Ø	H	"
					INC PC	95	2	H	92-3
					MB LD EN*	1Ø8	9	H	"
NEW CRY	19	1Ø	H	93-1	CRY SET	16	5	H	93-1
OMIT STROBE*	61	3	(OC)	92-3	STROBE	35	9	L	92-1
	61	11	(OC)	"	STROBE F/F	36	12	L	"
	61	6	(OC)	"					
	43	13	L	"					
	45	12	L	"					
OVFLO	14	8	H	93-1		(B39)			
OVFLO TO DIV	16	3	H	"	CRY SET	16	4	H	93-1
PC CLK	5Ø	6		92-1		(A85)			93-1
					(STATES				
					LOGIC)	64	13	L	92-2
					FETCH SKIP*	1ØØ	9	H	92-3
					PCØ-3	59	2	H	93-2
					PC4-7	54	2	H	"
					PC8-11	49	2	H	93-3
					PC12-15	44	2	H	"
					PC CLK*	1ØØ	8	L	92-3
PC CLK*	1ØØ	8	L	92-3	KEY SYNC F. F.	1Ø2	13		92-2
PC TO MEM*	1Ø4	8	L	92-3		(B36)			93-1
					ADDER TO				
					MEM*	1Ø3	4,1Ø	H	92-3
					MBOØ-3	6Ø	9	L	93-2
					MBO4-7	55	9	L	"
					MBO8-11	5Ø	9	L	93-3
					MBO12-15	45	9	L	"
PLUS ONE	1Ø	2	H	93-1	ADDER 12-15	89	3,4	H	93-3
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
PCØ	59	11		93-2					
PC1	59	12		93-2	SX-MULT1	76	2		93-2
					MBO1*	6Ø	5		"
PC2	59	13		93-2	SX-MULT2	76	19		93-2
					MBO2*	6Ø	11		"
PC3	59	14		93-2	SX-MULT3	76	22		93-2
					MBO3*	6Ø	14		"
PC4	54	11		93-2	SX-MULT4	72	5		93-2
					MBO4*	55	2		"
PC5	54	12		93-2	SX-MULT5	72	2		93-2
					MBO5*	55	5		"
PC6	54	13		93-2	SX-MULT6	72	19		93-2
					MBO6*	55	11		"
PC7	54	14		93-2	SX-MULT7	72	22		93-2
					MBO7*	55	14		
PC8	49	11		93-3	SX-MULT8	68	5		93-3
					MBO8	5Ø	2		"
PC9	49	12		93-3	SX-MULT9	68	2		93-3
					MBO9*	5Ø	5		"
PC1Ø	49	13		93-3	SX-MULT1Ø	68	19		93-3
					MBO1Ø*	5Ø	11		"
						(A78)			
					PL• last word*	38	5	L	92-3
PC11	49	14		93-3	SX-MULT11	68	22		93-3
					MBO11*	5Ø	14		"
PC12	44	11		93-3	SX-MULT12	64	5		93-3
					MBO12*	45	2		"
PI*	23	6	L	92-2	PI	84	5	L	92-2
					(MULT)	45	9	L	92-3
PI	84	6	H	92-2		(A93)			
					(States Logic)	83	9,1Ø	H	92-2
					ION	92	4	H	92-4
					INH trans*	3Ø	1	H	"
PI SET	4Ø	3,4		92-2	F+PI SET*	75	2	H	92-2
					MBO INH*	1Ø8	4,5	H	92-3

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
PL*	Connector	J4, Pin	23→		Back Plane	A19		L	89-1
PL*	(A19)		L	89-1	PL	24	9	L	92-2
					STOP RQ	8Ø	9	H	"
						74	5	H	92-3
					OMIT STROBE*	56	1	L	"
					LOAD PC*	97	4	L	"
PL	24	8	H	92-2	KEYM·PL*	95	13	H	92-3
PL LAST	11Ø	3	H	92-3	PC TO MEM*	11Ø	4	H	92-3
					LOAD PC*	93	2	H	"
PL·LAST WORD*	1ØØ	4	L	92-3	PC TO MEM*	1Ø6	3	H	92-3
					KEYM	97	9	L	92-2
PL·LAST WORD*	38	6	H	92-3	PL LAST	11Ø	2	L	92-3
					PL·LAST WORD	1ØØ	3	H	"
					INC PC	96	13	H	"
PRESET*	12	6	L	92-2	READ CY	2Ø	1	L	92-1
					READ 2	2Ø	1	L	"
					STROBE	16	1	L	"
					WRITE SYNC	16	1	L	"
					INHIBIT	17	1	L	"
					WRITE	17	1	L	"
						54	1	L	"
					STATES	39	1	L	92-2
					DCH LOGIC	69	1	L	92-4
						68	1	L	"
PROTECT*	(A23)		H	92-2	E·IO	56	12	H	92-2
					FETCH SKIP*	1Ø1	11	H	92-3
PTGØ*	32	11	L	92-1	DCHA*	74	13	H	92-4
					CON DATA*	29	9	H	"
					STATES LOGIC	82	13	L	"
PTG2*	33	8	L	92-1	PTG2	48	9	L	92-1
					DCHO LOGIC	7Ø	9	L	92-4
					DRIVE IO*	(B88)		L	"
					IO STUTTER*	9Ø	2	L	"
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
PTG2	48	8	H	92-1	LOAD PC*	93	9	H	92-3
					INC PC	96	3,4	H	"
					RUN SET Logic	68	1Ø	H	92-4
					DATOA	27	5	H	"
					DATOB	8	12	H	"
					DATOC	7	1Ø	H	"
					INTRQ*	91	4	H	"
					FETCH SKIP*	1Ø2	3	H	92-3
					IO STUTTER*	85	3	H	92-4
PTG1*	32	8	L	92-1	ROM ENABLE*	9Ø	9	L	92-3
					PTG1	48	11	L	92-1
PTG1	48	1Ø	H	92-1	CPU CLK logic	53	5	H	92-1
					MB LOAD	18	4	H	"
					INHIBIT LOGIC	35	12	H	"
					IR LOGIC	73	11	H	92-2
					STATES LOGIC	83	5	H	"
					FORCE PLUS one*	6Ø	1Ø	(OC)	92-3
					FORCE SX COM*	73	13	H	"
					LOAD AR	62	1	H	"
					MB LD EN*	1Ø8	2	(OC)	"
					LOAD PC*	93	4	H	"
					INC PC	96	1	H	"
					INC PC	35	1	H	92-3
					DCH LOGIC	69	13	H	92-4
					DATIB	88	1	H	"
					IO STUTTER*	85	6	H	"
					ION LOGIC	92	5	H	"
PTG3*	33	6	L	92-1	PTG3	48	5	L	92-1
					STATES	41	9	L	92-2
					ADDER TO MEM*	1Ø3	1	H	92-3
					LOAD PC*	94	2	L	"
	(B13)				AC WRITE*	12	5	L	93-1
PTG3	48	6	H	92-1	READ CY F/F	52	1	H	92-1
					MA LOAD*	52	13	H	"
					STATES	78	1Ø	H	92-2
					LOAD PC*	93	3	H	92-3
					PC TO MEM*	1Ø4	4	H	"
					LOAD AR	62	2	H	"

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
					LOAD CRY*	1Ø6	1Ø	H	92-3
					DCHA STUTTER	35	2	H	92-4
					DCH LOGIC	69	1Ø	H	"
					IO STUTTER*	88	3	H	"
					STRT	26	12	H	"
					CLR	26	1	H	"
					IOPLS	27	13	H	"
PWR LOW*				92-4	IO SKIP PEND	84	9	L	92-4
					INTRQ*	89	2	L	"
READ CY	2Ø	5		92-1	READ1*	19	9,1Ø	H	92-1
					"	19	12	H	"
					MB CLEAR*	19	4,5	H	"
					MB CLEAR	19	2	H	"
					READ2	2Ø	11	H	"
READ CY*	2Ø	6		92-1					
READ IO	71	6	H	92-4	IO STUTTER*	85	5	H	92-4
					READ IO*	84	13	H	92-4
					OMIT STROBE*	61	1	H	92-3
READ IO*	84	12	L	92-4		(B83)			1Ø4-2
					ADDER TO MEM*	1Ø3	2,12	H	92-3
					PC TO MEM*	1Ø4	3	H	"
					[READ IO]	18	3	L	1Ø4-1
	18	4	H	1Ø3-1	[MDØ]	17	13	H	"
					MD1	17	2	H	"
					MD2	15	13	H	"
					MD3	15	2	H	"
					MD4	13	13	H	"
					MD5	13	2	H	"
					MD6	11	13	H	"
					MD7	11	2	H	"
					MD8	9	13	H	"
					MD9	9	2	H	"
					MD1Ø	7	13	H	"
					MD11	7	2	H	"
					MD12	5	13	H	"
					MD13	5	2	H	"
					MD14	3	13	H	"
					MD15	3	2	H	"

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
[READ IO]	18	4	L	1Ø4-1		18	5	H	1Ø4-1
	18	6	L	"					
	18	6	L	"	[MDØ]	17	1Ø	L	"
					MD1	17	4	L	"
					MD2	15	1Ø	L	"
					MD3	15	4	L	"
					MD4	13	1Ø	L	"
					MD5	13	4	L	"
					MD6	11	1Ø	L	"
					MD7	11	4	L	"
					MD8	9	1Ø	L	"
					MD9	9	4	L	"
					MD1Ø	7	1Ø	L	"
					MD11	7	4	L	"
					MD12	5	1Ø	L	"
					MD13	5	4	L	"
					MD14	3	1Ø	L	"
					MD15	3	4	L	"
READ1*	19	8	L	92-1		(B87)		L	"
					READ1B	18	13	L	"
	18	12	H	1Ø4-1	"	19	5,4	H	"
					READ 2B	19	12	H	1Ø4-1
READ1B*	19	6	H	1Ø4-1	"	19	1Ø	H	"
					(X ADDR DCDR)	72	2		"
					"	76	2		"
					"	73	2		"
					"	77	2		"
					"	79	3		"
					"	74	3		"
					"	78	3		"
					"	75	3		"
					(Y ADDR DCDR)	54	2		1Ø4-4
						62	2		"
						52	2		"
						66	2		"
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
READ2	2Ø	9	H	92-1	MB CLR*	38	9	H	92-1
					WRITE SYNC	16	11	H	"
READ2*	2Ø	8	L	92-1		(B9Ø)		L	1Ø4-1
					STROBE	36	9	L	92-1
					READ2B	18	11	L	1Ø4-1
	18	1Ø	H	1Ø4-1	READ2B	19	9	H	"
						19	13	H	"
READ2B	19	8	H	1Ø3-2	(Y ADDR DCDR)	6Ø	3	H	1Ø4-4
					"	5Ø	3	H	"
					"	57	3	H	"
					"	47	3	H	"
REAL IO INST	75	8	H	92-2	E·IO	56	13	H	92-2
					IO SKIP	58	1	H	"
					INC PC	98	1	L	93-3
RELOAD DISABLE	(B72)			92-1	STROBE	38	11	H	92-1
					(MB LOAD GATE)	36	12	H	1Ø4-1
RESTART ENABLE	Back	Plane	A32 L	→	Connector	J4 Pin	19	to →	
					Console switchS5			L	
RESET*	24	1Ø	L	92-2		6	1Ø	H	92-2
					PRESET*	12	1	L	"
					(MR)	78	1	L	"
					ION	91	1Ø	L	92-4
					IORST	5	1	L	"
RINHØ	(A5)			1Ø4-2					
RINH1	(A7)			"					
RINH2	(A9)			"					
RINH3	(A11)			"					
RINH4	(A13)			"					
RINH5	(A15)			"					
RINH6	(A18)			"					
RINH7	(A17)			"					
RINH8	(A19)			"					
RINH9	(A24)			"					
RINH1Ø	(A23)			"					
RINH11	(A21)			"					
RINH12	(A28)			"					
RINH13	(A25)			"					
RINH14	(A29)			"					

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
RINH15	(A27)			1Ø4-2					
ROM ENABLE*	66	4	L	92-3		(B19)			92-3
					ROM	3Ø	15	L	93-1
					ROM	31	15	L	"
RQENB*	79	8	L	92-2		(B41)			
RUN	78	7	H	"	RUN*	77	11	H	92-2
					RQENB*	79	9	H	"
					CON DATA*	29	3	H	92-4
						34	1	L	92-1
RUN*	77	1Ø	L	"		(A14)			
					KEY SEEN F/F	82	1,2	L	92-2
					PC TO MEM*	1Ø4	1Ø	H	92-3
					Connector J4, Pin		26	to	→
						12	1	L	89-1
[RUN SET*]	51	3	L	92-4	RUN	66	1	L	92-4
RUN SET	66	2	H	"	RUN	78	2	H	92-2
					RQENB*	79	1Ø	H	"
					MA LOAD*	87	12	H	92-1
					READ CY	52	2	H	"
RST*	Connector J4, Pin 20→				Back Plane	A30		L	89-1
	Back Plane A30→				RESET*	6	2	L	92-2
SELB	(A82)			92-4	IO SKIP PEND	86	5	L	92-4
SELO	(A8Ø)			92-4	IO SKIP PEND	86	3	L	92-4
SELECT	35	8	H	1Ø4-1	(MEM bus gate)	36	1	H	1Ø4-1
					(MB load enable)	36	13	H	"
					READ 1B	19	1	H	"
					READ 1B	19	2	H	"
					STRBA, B, C, D	1	1	H	"
					STROBE	1	1Ø	H	"
					STROBE	1	12	H	"
					STROBE	1	13	H	"
					INH GATE A&B	26	2	H	"
					INH GATE A&B	26	4	H	"
					INH GATE A&B	26	5	H	"
					DRIVE IO	26	13	H	"
					INHIBIT	41	1Ø	H	"
*Indicates "NOT"					INHIBIT	41	12	H	"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
					INHIBIT	41	13	H	1Ø4-1
					MB CLEAR*	30	1	L	"
SEL-L	2Ø	6	H	93-1	CRY TEST*	4Ø	13	H	93-1
					SUMØ	1Ø3	2	H	93-2
					SUM1	1Ø2	2	H	"
					SUM2	1Ø1	2	H	"
					SUM3	1ØØ	2	H	"
					SUM4	98	2	H	"
					SUM5	97	2	H	"
					SUM6	96	2	H	"
					SUM7	95	2	H	"
					SUM8	93	2	H	93-3
					SUM9	92	2	H	"
					SUM1Ø	91	2	H	"
					SUM11	9Ø	2	H	"
					SUM12	88	2	H	"
					SUM13	87	2	H	"
					SUM14	86	2	H	"
					SUM15	85	2	H	"
SEL-N	18	8	H	93-1	CRY TEST*	4Ø	2	H	93-1
					SUMØ	1Ø3	1	H	93-2
					SUM1	1Ø2	1	H	"
					SUM2	1Ø1	1	H	"
					SUM3	1ØØ	1	H	"
					SUM4	98	1	H	"
					SUM5	97	1	H	"
					SUM6	96	1	H	"
					SUM7	95	1	H	"
					SUM8	93	1	H	93-3
					SUM9	92	1	H	"
					SUM1Ø	91	1	H	"
					SUM11	9Ø	1	H	"
					SUM12	88	1	H	"
					SUM13	87	1	H	"
					SUM14	86	1	H	"
					SUM15	85	1	H	"

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
SEL-R	18	6	H	93-1	CRY TEST*	4Ø	9	H	93-1
					SUMØ	1Ø3	9	H	93-2
					SUM1	1Ø2	9	H	"
					SUM2	1Ø1	9	H	"
					SUM3	1ØØ	9	H	"
					SUM4	98	9	H	"
					SUM5	97	9	H	"
					SUM6	96	9	H	"
					SUM7	95	9	H	"
					SUM8	93	9	H	93-3
					SUM9	92	9	H	"
					SUM1Ø	91	9	H	"
					SUM11	9Ø	9	H	"
					SUM12	88	9	H	"
					SUM13	87	9	H	"
					SUM14	86	9	H	"
					SUM15	85	9	H	"
SEL-S	2Ø	8	H	93-1	CRY TEST*	4Ø	5	H	93-1
					SUMØ	1Ø3	4	H	93-2
					SUM1	1Ø2	4	H	"
					SUM2	1Ø1	4	H	"
					SUM3	1ØØ	4	H	"
					SUM4	98	4	H	"
					SUM5	97	4	H	"
					SUM6	96	4	H	"
					SUM7	95	4	H	"
					SUM8	93	4	H	93-3
					SUM9	92	4	H	"
					SUM1Ø	91	4	H	"
					SUM11	9Ø	4	H	"
					SUM12	88	4	H	"
					SUM13	87	4	H	"
					SUM14	86	4	H	"
					SUM15	85	4	H	"

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
SHIFT AR LEFT	1Ø	4	H	93-1	AR12-15	42	13	H	93-3
					AR8-11	47	13	H	"
					AR4-7	52	13	H	93-2
					ARØ-3	57	13	H	"
+SLØ				1Ø4-2	SNSØ	69	2		1Ø4-2
-SLØ				"	"	69	3		"
+SL1				"	SNS1	69	6		"
-SL1				"	"	69	7		"
+SL2				"	SNS2	65	2		"
-SL2				"	"	65	3		"
+SL3				"	SNS3	65	6		"
-SL3				"	"	65	7		"
+SL4				"	SNS4	59	2		"
-SL4				"	"	59	3		"
+SL5				"	SNS5	59	6		"
-SL5				"	"	59	7		"
+SL6				"	SNS6	56	2		"
-SL6				"	"	56	3		"
+SL7				"	SNS7	56	6		"
-SL7				"	"	56	7		"
+SL8				1Ø4-2	SNS8	49	2		1Ø4-2
-SL8				"	"	49	3		"
+SL9				"	SNS9	49	6		"
-SL9				"	"	49	7		"
+SL1Ø				"	SNS1Ø	46	2		"
-SL1Ø				"	"	46	3		"
+SL11				"	SNS11	46	6		"
-SL11				"	"	46	7		"
+SL12				"	SNS12	4Ø	2		"
-SL12				"	"	4Ø	3		"
+SL13				"	SNS13	4Ø	6		"
-SL13				"	"	4Ø	7		"
+SL14				"	SNS14	38	2		"
-SL14				"	"	38	3		"
+SL15				"	SNS15	38	6		"
-SL15				"	"	38	7		"
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
SNSØ	69	14	H	1Ø4-2	SNSØ*	68	9	H	1Ø4-2
SNSØ*	68	8	L	"	INHØF/F	34	1Ø	L	1Ø4-1
SNS1	69	12	H	"	SNS1*	68	5	H	1Ø4-2
SNS1*	68	6	L	"	INH1F/F	34	4	L	1Ø4-1
SNS2	65	14	H	"	SNS2*	64	5	H	1Ø4-2
SNS2*	64	6	L	"	INH2F/F	32	4	L	1Ø4-1
SNS3	65	12	H	"	SNS3*	64	9	H	1Ø4-2
SNS3*	64	8	L	"	INH3F/F	32	1Ø	L	1Ø4-1
SNS4	59	14	H	"	SNS4*	58	9	H	1Ø4-2
SNS4*	58	8	H	"	INH4F/F	31	1Ø	L	1Ø4-1
SNS5	59	12	H	"	SNS5*	58	5	H	1Ø4-1
SNS5*	58	6	L	"	INH5F/F	31	4	L	1Ø4-1
SNS6	56	14	H	"	SNS6*	55	5	H	1Ø4-2
SNS6*	55	6	L	"	INH6F/F	28	4	L	1Ø4-1
SNS7	56	12	H	"	SNS7*	55	9	H	1Ø4-2
SNS7*	55	8	L	"	INH7F/F	28	1Ø	L	1Ø4-1
SNS8	49	14	H	"	SNS8*	48	9	H	1Ø4-2
SNS8*	48	8	L	"	INH8F/F	27	1Ø	L	1Ø4-1
SNS9	49	12	H	"	SNS9*	48	5	H	1Ø4-2
SNS9*	48	6	L	"	INH8F/F	27	4	L	1Ø4-1
SNS1Ø	46	14	H	"	SNS1Ø*	45	5	H	1Ø4-2
SNS1Ø*	45	6	L	"	INH1ØF/F	24	4	L	1Ø4-1
SNS11	46	12	H	"	SNS11*	45	9	H	1Ø4-2
SNS11*	45	8	L	"	INH11F/F	24	1Ø	L	1Ø4-1
SNS12	4Ø	14	H	"	SNS12*	39	9	H	1Ø4-2
SNS12*	39	8	L	"	INH12F/F	23	1Ø	L	1Ø4-1
SNS13	4Ø	12	H	"	SNS13*	39	5	H	1Ø4-2
SNS13*	39	6	L	"	INH13F/F	23	4	L	1Ø4-1
SNS14	38	14	H	"	SNS14*	37	5	H	1Ø4-2
SNS14*	37	6	L	"	INH14F/F	21	4	L	1Ø4-1
SNS15	38	12	H	"	SNS15*	37	9	H	1Ø4-2
SNS15*	37	8	L	"	INH15F/F	21	1Ø	L	1Ø4-1
STA	65	6	H	92-2	FORCE D-L-SEL*	43	1	H	92-3
					OMIT STROBE*	43	15	H	"

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
STATE SUPPRESS	79	6	H	92-2	(STATES LOGIC)	41	1Ø	L	92-2
					F + PI	41	12	L	"
					PI SET/F SET	4Ø	9	L	"
					STATE SUPPRESS*	59	5	H	"
					FORCE SX-L-SEL*	63	1Ø	H	92-3
					FORCE SX-H-SEL*	63	12	H	"
					LOAD AR	99	5	H	"
					LOAD PC*	94	1	L	"
STATE SUPPRESS*	59	6	L	92-2	PI SET/F SET	4Ø	7	L	92-2
					E*	22	9	H	"
					D*	23	2	H	"
					PI*	23	4	H	"
					F*	23	1Ø	H	"
STOP*	Connector J4, Pin 45→ (A31)				Back Plane	A31			
					STOP RQ	98	12	L	92-2
STOP RQ	5	*	H	92-2	RUN SET	7Ø	2	H	92-4
STRB A	1	6	H	1Ø4-1	SNSØ*	68	1Ø	L	1Ø4-2
					SNS1*	68	4	L	"
					SNS2*	64	4	L	"
					SNS3*	64	1Ø	L	"
STRB B	1	6	H	1Ø4-1	SNS4*	58	1Ø	L	"
					SNS5*	58	4	L	"
					SNS6*	55	4	L	"
					SNS7*	55	1Ø	L	"
STRB C	1	6	H	1Ø4-1	SNS8*	48	1Ø	L	"
					SNS9*	48	4	L	"
					SNS1Ø*	45	4	L	"
					SNS11*	45	1Ø	L	"
STRB D	1	6		1Ø4-1	SNS12*	39	1Ø	L	"
					SNS13*	39	4	L	"
					SNS14*	37	4	L	"
					SNS15*	37	1Ø	L	"
STROBE	16	5		92-1		(B2Ø)			92-1
	Back Plane		B2Ø→		STRB A, B, C, B	1	5	H	1Ø4-1
						1	9	H	1Ø4-1
STROBE*	16	6		92-1					
STRT	26	8	H	92-4		(A52)			
					[ION]	89	13	H	92-4

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
SUM CRY	19	8	H	93-1	SUMØ	1Ø3	1Ø	H	93-2
					SUM15	85	3	H	93-3
SUMØ	84	8		93-2	ADDER=Ø	83	5		93-2
					ARØ	57	14		"
SUM1	84	11		93-2	ADDER=Ø	83	1		93-2
					AR1	57	15		"
					PC1	59	5		"
SUM2	84	3		93-2	ADDER=Ø	83	13		93-2
					AR2	57	2		"
					PC2	59	4		"
SUM3	84	6		93-2	ADDER=Ø	83	9		93-2
					AR3	57	3		"
					PC3	59	3		"
SUM4	82	8		93-2	ADDER=Ø	81	5		93-2
					AR4	52	14		"
					PC4	54	6		"
SUM5	82	11		93-2	ADDER=Ø	81	1		93-2
					AR5	52	15		"
					PC5	54	5		"
SUM6	82	3		93-2	ADDER=Ø	81	13		93-2
					AR6	52	2		
					PC6	54	4		
SUM7	82	6		93-2	ADDER-Ø	81	9		93-2
					AR7	52	3		
					PC7	54	3		
SUM8	8Ø	8		93-3	ADDER=Ø	79	5		93-3
					AR8	47	14		"
					PC8	49	6		"
SUM9	8Ø	11		93-3	ADDER=Ø	79	1		93-3
					AR9	47	15		"
					PC9	49	5		"
SUM1Ø	8Ø	3		93-3	ADDER=Ø	79	13		93-3
					AR1Ø	47	2		"
					PC1Ø	49	4		"
SUM11	8Ø	6		93-3	ADDER=Ø	79	9		93-3
					AUT INC + DEC	42	1	H	92-2
	(A73)			93-1	AR11	47	3		"
					PC11	49	3		"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
SUM12	78	8		93-3	ADDER=Ø	77	5		93-3
					AUT DEC	39	3	H	92-2
	(A75)			93-1	AR12	42	14		"
					PC12	44	6		"
SUM13	78	11		93-3	ADDER=Ø	77	1		93-3
					AR13	42	15		"
					PC13	44	5		"
SUM14	78	3		93-3	ADDER-Ø	77	12		93-3
					AR13	42	2		"
					PC13	44	4		"
SUM15	78	6		93-3	ADDER=0	77	9		93-3
					AR13	42	3		"
					PC13	44	3		"
SUPPRESS*	(A49)		L	92-2	STATE SUPPRESS	79	4	L	92-2
SX COM	1Ø	6	H	93-1	SX-MULTØ-3	76	15	H	93-2
					SX-MULT4-7	72	15	H	"
					SX-MULT8-11	68	15	H	93-3
					SX-MULT12-15	64	15	H	"
SX-H-READ	32	12		93-1	ACSXØ-3	75	4		93-2
					ACSX4	71	4		"
					ACSX8-11	67	4		93-3
					ACSX12-15	63	4		"
SX-L-READ	1Ø	12		93-1	ACSXØ-3	75	5		93-2
					ACSX4-7	71	5		"
					ACSX8-11	67	5		93-3
					ACSC12-15	63	5		"
SX-H-SEL	1Ø	8		93-1	SX-MULTØ-3	76	16		93-2
					SX-MULT4-7	72	16		"
					SX-MULT8-11	68	16		93-3
					SX-MULT12-15	64	16		"
SX-L-SEL	1Ø	1Ø		93-1	SX-MULTØ-3	76	17		93-2
					SX-MULT4-7	72	17		"
					SX-MULT8-11	68	17		93-3
					SX-MULT12-15	64	17		"

*Indicates "NOT"

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
SX-MULTØ	76	1Ø		93-2	ADDERØ	1Ø4	17		93-2
SX-MULT1	76	11		"	ADDER1	1Ø4	19		"
SX-MULT2	76	14		"	ADDER2	1Ø4	21		"
SX-MULT3	76	13		"	ADDER3	1Ø4	23		"
SX-MULT4	72	1Ø		"	ADDER4	99	17		"
SX-MULT5	72	11		"	ADDER5	99	19		"
SX-MULT6	72	14		"	ADDER6	99	21		"
SX-MULT7	72	13		"	ADDER7	99	23		"
SX-MULT8	68	1Ø		93-3	ADDER8	94	17		93-3
SX-MULT9	68	11		"	ADDER9	94	19		"
SX-MULT1Ø	68	14		"	ADDER1Ø	94	21		"
SX-MULT11	68	13		"	ADDER11	94	23		"
SX-MULT12	64	1Ø		"	ADDER12	89	17		"
SX-MULT13	64	11		"	ADDER13	89	19		"
SX-MULT14	64	14		"	ADDER14	89	21		"
SX-MULT15	64	13		"	ADDER15	89	23		"
TSM	34	9	H			(B25)			
					PTG2*	35	12	H	92-1
					PTG1*	32	9	H	"
					MEM LATCH	14	1	H	93-1
					MBB 9, 1Ø, 11	19	13	L	"
TSØ	34	6	H	92-1		(A71)			
						34	11	H	92-1
					PTGØ*	32	12	H	"
					PTG1*	32	1Ø	H	"
					RQENB*	79	13	H	92-2
					ALC*	41	5	L	"
					FORCE SX H SEL*	45	2	H	92-3
					MEM OUT*	44	1Ø	H	"
TS3	34	5	H	92-1		34	12	H	92-1
					PTG2*	33	13	H	"
					PTG3*	33	2	H	"
					FORCE PLUS ONE*	45	5	H	92-3
					(MULT)	45	7	L	"
WRITE AC*	1Ø7	8	L	92-3		(B54)			
					AC WRITE*	12	2	L	
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
WAIT*	(A9Ø)			92-4	WHOA	67	9	L	92-4
WHOA*	(B6)			92-4	WHOA	67	1,1Ø	L	92-4
WHOA	67	8	H	92-4	CLK-B	51	4	L	92-1
WRITE	17	9		92-1	WRITE SYNC	16	12	H	92-1
WRITE*	17	8		92-1					
WRITE MEM	41	6	H	1Ø4-1	XØØ	72	3		1Ø4-3
					X6Ø	76	3		"
					X3Ø	73	3		"
					X2Ø	77	3		"
					XØØ	79	2		"
					XØ1	74	2		"
					XØ4	78	2		"
					XØ5	75	2		"
					X7Ø	54	3		1Ø4-4
					Y6Ø	62	3		"
					Y3Ø	52	3		"
					Y2Ø	66	3		"
					YØØ	6Ø	2		"
					YØ1	5Ø	2		
					YØ4	57	2		"
					YØ5	47	2		"
WRITE SYNC	16	9		92-1	READ CY	2Ø	2	H	92-1
					READ 2	2Ø	12	H	"
WRITE SYNC*	16	8		92-1	INHIBIT	17	2	H	92-1
					WRITE	17	12	H	"
					INHIBIT	36	5	L	"
					CPU CLK	36	2	L	"
=Ø ENABLE	11Ø	11	L	92-3		(B52)			93-1
					SUMØ	83	4		93-2
					SUM11	79	1Ø		93-3
					SUM12	77	4		"
					SUM13	77	2		"
					SUM14	77	13		"
					SUM15	77	1Ø		"
*Indicates "NOT"									

SIGNAL LIST

CENTRAL PROCESSOR AND MEMORY

NOVA 800

ORIGIN					DESTINATION				
SIGNAL	CHIP	PIN	LEVEL	DWG	FUNCTION	CHIP	PIN	LEVEL	DWG
12TC	44	15	H	93-3	PC8-11	49	1Ø	H	93-3
8TC	49	15	H	"	PC4-7	54	1Ø	H	93-2
4TC	54	15	H	92-2	PCØ-3	59	1Ø	H	"
+ 5 OK	Connector		P2, Pin	5Ø →	Back Plane	A8			91-1