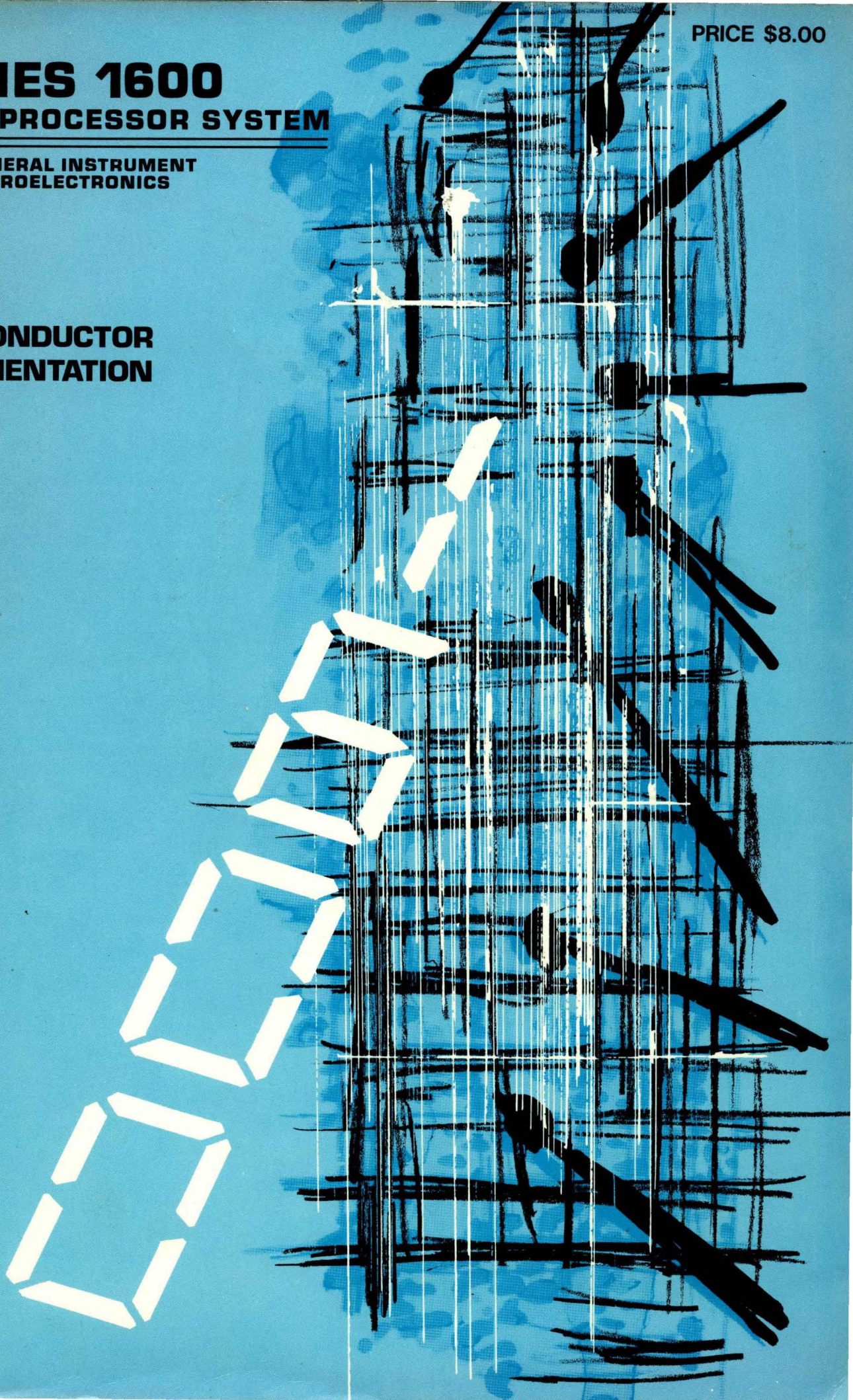
PRICE $8.00

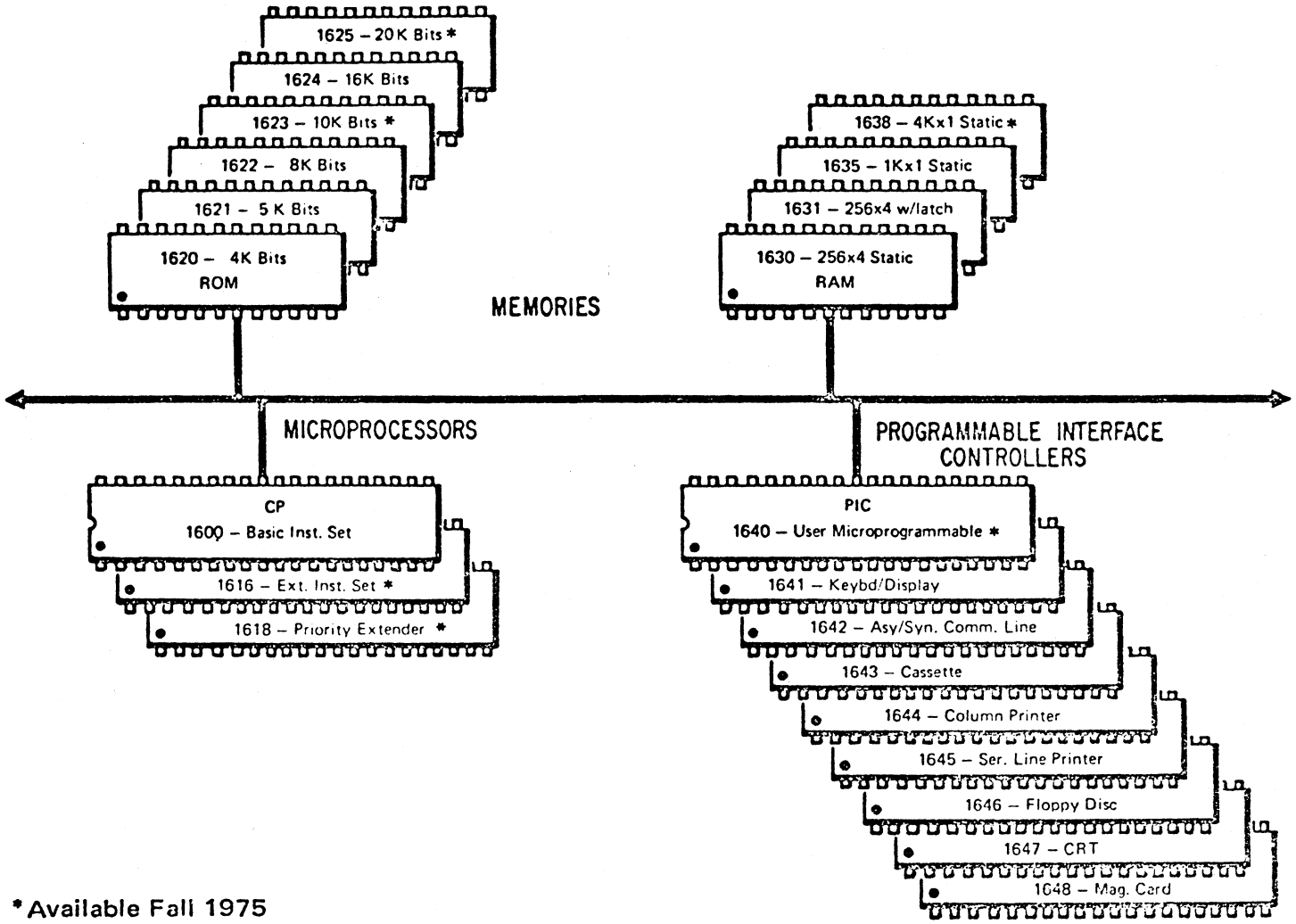
SERIES 1600
MICROPROCESSOR SYSTEM

GENERAL INSTRUMENT
MICROELECTRONICS

SEMICONDUCTOR DOCUMENTATION



THE SERIES 1600 SEMICONDUCTOR LINEUP



*Available Fall 1975

SERIES 1600 MICROPROCESSOR SYSTEM
SEMICONDUCTOR DOCUMENTATION

CP-1600 MICROPROCESSOR
USERS MANUAL

This manual contains a detailed specification of the CP-1600 microprocessor array including theory of operation, electrical parameters, instruction set, and some typical applications examples.

GENERAL INSTRUMENT CORPORATION ■ MICROELECTRONICS DIVISION ■ 600 W. John St., Hicksville, N.Y. 11802 ■ 516-733-3107

SUMMARY

Title: CP1600 Microprocessor Users Manual

Document No.: S16DOC-CP1600-04, May 1975

Revision Level: Supersedes S16DOC-CP1600-03, March 1975

Scope: This manual describes the CP 1600 microprocessor array. It includes information on the theory of operation of the processor, timing diagrams of all operational sequences, electrical parameters, and details of the instruction set. In addition, a number of systems configurations of the CP 1600 are shown by way of typical interface examples.

Reference Documents:

S16DOC-GIC1600-00	GIC1600 Microcomputer Users Manual
S16DOC-XALSIM-01	Series 1600 Cross Software Manual

TABLE OF CONTENTS

TABLE OF CONTENTS

(Continuation)

CHAPTER 1

CP1600 GENERAL DESCRIPTION

1.0 INTRODUCTION

The CP1600 MicroProcessor Unit is a compatible member of the Series 1600 MicroProcessor Products family. It is a complete, 16-bit, single chip, high speed MOS-LSI MicroProcessor. The Series 1600 is fabricated with General Instrument's N-Channel Ion-Implant GIANT II process, insuring high performance with proven reliability and production history. All members of the Series 1600 Family, including Programmable Interface Controllers, Read Only Memories, and Random Access Read/Write Memories, are fully compatible with the CP1600.

The CP1600 MicroProcessor Unit is designed for high speed data processing and real time applications. Using a 5 MHz, 2-phase clock, the CP1600 completes a microcycle in 400 nanoseconds. Typical applications include programmable calculator systems, peripheral controllers, process controllers, intelligent terminals and instruments, data acquisition and digital communication processors, numerical control systems, and many general purpose computer applications. The MicroProcessor can readily support a variety of peripheral equipment such as TTY, CRT display, tape reader/punch, A/D & D/A converter, keyboard, cassette tape, floppy disk, and RS-232C data communication lines.

The CP1600 MPU utilizes third generation minicomputer architecture with eight general purpose registers to achieve a versatile, sophisticated microcomputer system. The 16-word enables fast and efficient processing of alpha-numeric or byte oriented data. The 16-bit address capability permits accessing of 65,536 words in any combination of program memory, data memory, or peripheral devices. This single address space concept, combined with a powerful instruction set and microprogrammable Peripheral Interface devices, provides an efficient solution to all microcomputer and many minicomputer-based product requirements.

- 87 BASIC INSTRUCTIONS
- 16-BIT, TWO's COMPLEMENT ARITHMETIC AND LOGIC OPERATIONS
- STATUS REGISTER - OVERFLOW, CARRY, SIGN, ZERO, INTERRUPT ENABLE
- ARITHMETIC, LOGICAL, AND CIRCULAR SHIFTS
- 8 INTERNAL PROGRAM ACCESSIBLE GENERAL REGISTERS (16-BIT)
- MEMORY STACK POINTER/UNLIMITED STACK DEPTH
- IMMEDIATE DATA, DIRECT (65K), AND REGISTER INDIRECT (65K) ADDRESSING
- PROGRAM COUNTER RELATIVE ADDRESSING ON CONDITIONAL BRANCH
- CONDITIONAL BRANCH - STATUS REGISTER, AND 16 EXTERNALS
- TWO PROGRAMMABLE INTERRUPT LINES WITH PRIORITY RESOLUTION AND SELF IDENTIFYING ADDRESSES
- DMA CHANNELS FOR HIGH SPEED DEVICE TRANSFERS
- TTL COMPATIBLE/SIMPLE BUS STRUCTURE
- CYCLE TIME: 400 NSECS
- ADDS TWO 16-BIT INTERNAL REGISTERS IN 2.4 μSECS
- MEMORY TO REGISTER ADD TWO 16-BIT NUMBERS IN 3.2 μSECS

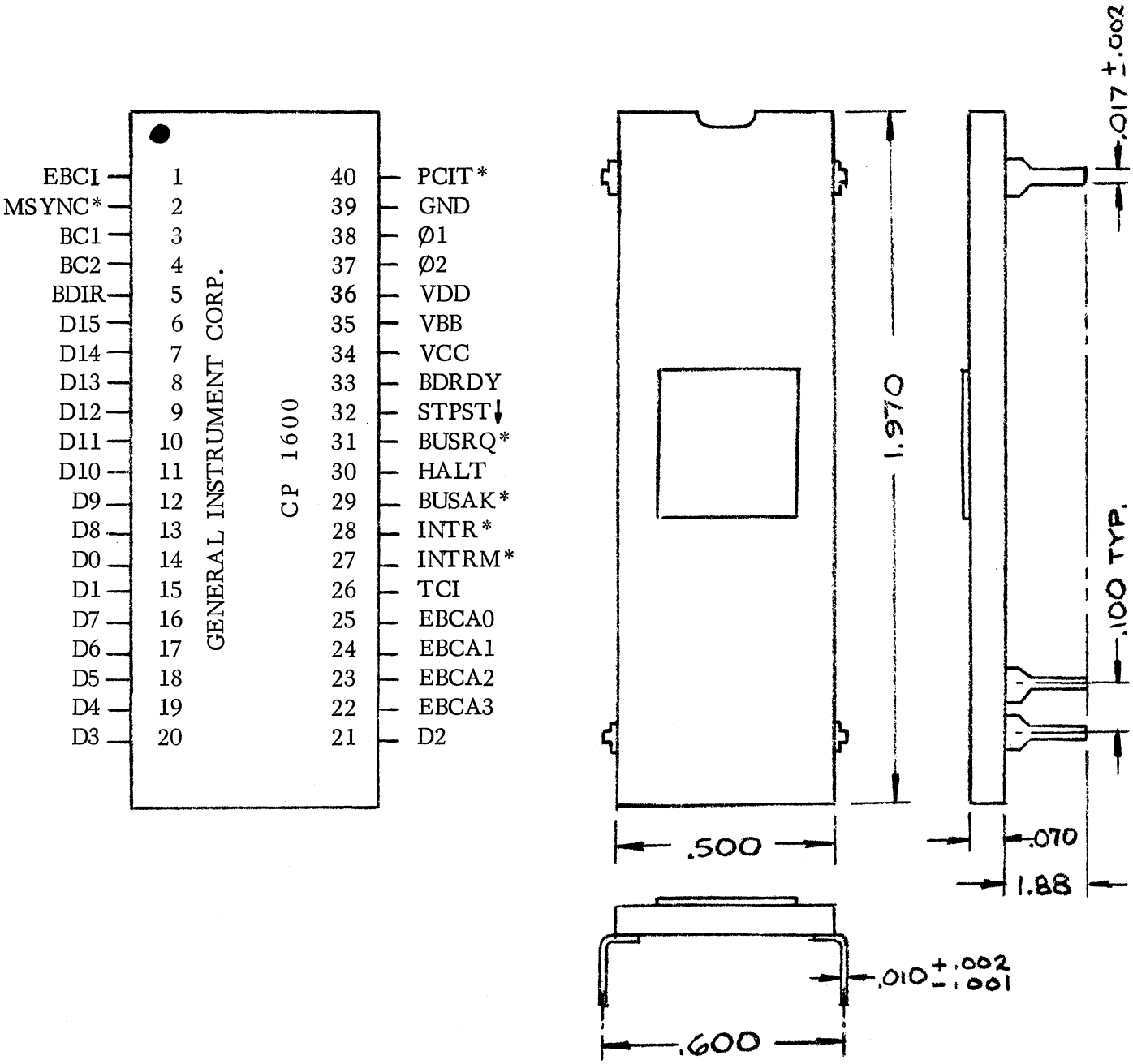
CHAPTER 2

CP 1600 PROCESSOR SPECIFICATION

2.0 PROCESSOR DESCRIPTION

The General Instrument Corporation CP 1600 microprocessor is a high speed, 16-bit machine featuring full 16-bit address/data transfer and a 400 nanosecond internal cycle time. Full instruction execution times range from 1.6 to 4.8 microseconds.

The processor contains 8 general purpose 16-bit program accessible registers, a high speed ALU, an instruction register and microcontrol unit, and TTL compatible input/output buffers. The CP 1600 is supplied in a 40 pin dual-in-line ceramic package with input/output signals shown in Fig. 2-1.



Logic Conventions:

* Negative logic (low assertion)

↑ Low to high edge trigger

↓ High to low edge trigger

All others Positive logic (high assertion)

FIG. 2 - 1

2.1 PROCESSOR SIGNALS

D0 - D15 (input/output/high impedance)
DATA 0 - 15 (Positive Logic)

These 16 signals comprise the 16-bit bidirectional bus used by the CPU to send both data and addresses to the external subsystems and receive data and instructions from the external subsystems, i.e., memory, printers, keyboards, displays, A/D and D/A converters, magnetic tape, cassettes, and other peripherals of various types. All operations on the bus involve 16-bit transfers of either data, address or instruction words.

BDIR (output)
BUS DIRECTION (Positive Logic)

This signal indicates the direction of the 16-bit bidirectional bus at the CPU. A logic "1" (High) indicates that D0-D15 are outputting to the bus. A logic "0" (Low) indicates that D0-D15 are either inputting from the bus or are in a high impedance state.

BC1, BC2 (outputs)
BUS CONTROL 1, 2 (Positive Logic)

These signals in conjunction with BDIR are used to control all external bus operations. In simple systems, they can be used directly as control signals while in more elaborate ones they can be decoded in a single 3 to 8 decoder to provide all eight external bus control functions listed below:

BC1	BC2	BDIR	BUS CONTROL	FUNCTION
0	0	1	BAR	BUS TO ADDRESS REGISTER This signal is used to load the contents of the bus into the external address register. D0-D15 are in an output mode.
0	1	1	DWS	DATA WRITE STROBE This signal is used as a write enable for memory or any peripheral device. D0-D15 are in an output mode.
1	0	1	DW	DATA WRITE This signal is functionally identical to DWS except that it occurs one machine cycle time prior to DWS. It is used for extended write operation purposes. D0-D15 are in an output mode.
1	1	1	INTAK	INTERRUPT ACKNOWLEDGE This signal denotes CPU acceptance of an interrupt and is used by the interrupting devices to resolve priority. The INTERRUPT SEQUENCE of the internal CPU state timing logic commences with this acknowledgement. Further details of the interrupt service are given

BC1	BC2	BDIR	BUS CONTROL	FUNCTION
				in the section on INTERNAL CPU ARCHITECTURE. D0-D15 are in an output mode.
0	1	0	IAB	INTERRUPT ADDRESS TO BUS This signal which occurs during the INTERRUPT SEQUENCE of the CPU, serves to gate the starting address of the interrupt service routine for the acknowledged interrupting device onto the bus. D0-D15 are in an input mode. The CPU will input this starting address into the PC to commence execution of the service routine. IAB also occurs during the power-up initialization sequence to input the starting address of the main program to the PC.
1	0	0	ADAR	ADDRESSED DATA TO ADDRESS REGISTER This signal causes the addressed contents of memory to be gated onto the bus and strobed into the address register. It is activated during all instructions which specify direct addressing. D0-D15 are in the high impedance state.
1	1	0	DTB	DATA TO BUS This signal is used to gate data from memory or any peripheral device onto the bus. Data in this sense can mean instructions, addresses, or data. D0-D15 are in an input mode.
0	0	0	NACT	NO ACTION This signal indicates the CPU is not using the bus. D0-D15 are in the high impedance state.

EBCA0, EBCA1, EBCA2, EBCA3 (outputs)
EXTERNAL BRANCH CONDITION ADDRESS 0, 1, 2, 3 (Positive Logic)

These lines are the buffered outputs from the 4 least significant bits of the instruction register (bits 0-3) and are used to externally select one of sixteen digital states to be sampled by the CPU during the execution of the BEXT (Branch on EXTernal) instruction.

EBCI (input)
EXTERNAL BRANCH CONDITION IN (Positive Logic)

This is the return signal of the 1 of 16 selection made by EBCA 0, 1, 2, 3. It is high active such that a logic "1" (High) would test as true and cause the branch to occur as part of the BEXT instruction. A logic "0" would not allow a branch to occur.

MSYNC* (input)
MASTER SYNC (Negative Logic)

This signal is used to synchronize the internal clock generator logic to the main Ø1, Ø2 clock phases. Synchronization is achieved by remaining active (low) for at least ten milliseconds after power is applied to the CPU and becoming inactive (high) on any rising edge of a Ø1 clock. That Ø1 clock corresponds to an internal TS3. This signal also disables the interrupt system.

VDD
Nominal 12 volts power supply input to the CPU.

VCC
Nominal 5 volts power supply input to the CPU.

VBB
Nominal -3 volts power supply input to the CPU.

BDRDY (input)
BUS DATA READY (Positive Logic)

This signal is used to make the CPU "wait" and resynchronize itself to peripheral subsystems that cannot respond to requests for reads and writes at full CPU speed. This is intended primarily for synchronization to slow speed memories. The duration of the "wait" period must be less than 40 microseconds to preserve the dynamic state of the CPU.

INTR*, INTRM* (inputs)
INTERRUPT REQUEST* (Negative Logic)
INTERRUPT REQUEST MASKABLE* (Negative Logic)

These two signals are low active signals that request the CPU to honor an interrupt at the completion of any interruptable instruction under the following conditions.

a) INTR* is always honored by the CPU and hence is the highest priority interrupt request line.

b) INTRM* is honored by the CPU only if the internal CPU interrupt flip-flop, INTFF, is set. This flip-flop is controlled via the EIS and DIS instructions and several others that have similar capability as described in the instruction set.

The CPU will enter the INTERRUPT SEQUENCE if either INTR* is active or if INTRM* is active and INTFF is set. During this sequence, the CPU will store the PC onto the top of the memory stack and resolve interrupt device priority by enabling INTAK. This is followed by inputting the starting address of the interrupt service routine from the appropriate interrupting

device into the PC by enabling IAB. The next instruction is fetched from the location pointed to by the new PC and execution of the service routine begins from that point.

TCI (output)
TERMINATE CURRENT INTERRUPT (Positive Logic)

This high active pulse signal is generated via the TCI instruction to terminate the highest priority interrupt presently in service.

Ø1, Ø2 (inputs)

These signals supply the high-level, high-speed, non-overlapping two phase clocks to the CPU.

PCIT* (input/output)
PROGRAM COUNTER INHIBIT*/TRAP* (Negative Logic)

This pin provides two functions.

a) As an input, this signal is a low active signal that prevents the incrementation of the program counter (R7) during the fetch phase of all instructions. Care should be used when utilizing this function during multiword instructions.

b) As an output, this signal will generate a low active pulse during the execution of the SIN (Software INterrupt) instruction. Properly used, this signal can be returned to the CPU by the interfacing hardware as an interrupt request on either of the INTR* or INTRM* input pins. The interrupt will be acknowledged at the end of the SIN instruction.

These functions will not interact with each other under normal operation. The Pin Timing Diagrams in Section 2.3 will precisely describe the timing of each function.

BUSRQ* (input)
BUS REQUEST* (Negative Logic)

This signal is low active and causes the CPU to relinquish all control of the bus to allow other devices to carry out direct bus transfer operations. The CPU grants use of the bus only after the completion of an interruptable instruction (refer to the instruction listings for those instructions which are interruptable and those which are not). The CPU signifies release of the bus by causing BUSAK* to go low and the CPU will remain in this condition until the external device releases BUSRQ*.

BUSAK* (output)
BUS ACKNOWLEDGE* (Negative Logic)

This is a low active signal used to inform external devices that the bus has been released by the CPU in response to the BUSRQ* signal.

STPST↓ (input)
STOP START (Negative Edge Triggered)

This is an edge-triggered signal used to control the running condition of the CPU. If the CPU is presently running, the negative transition of STPST↓ will cause the CPU to stop but only after the completion of an interruptable instruction. The CPU will generate a high active HALT signal acknowledging the stopped mode. The next negative transition of STPST↓ will cause the CPU to return to the run mode. The HALT output will then return to a logic "0" (Low) condition.

HALT (output)
HALT (Positive Logic)

This is a high active signal indicating that the CPU is in the stopped mode. This mode can occur by either the toggle action from the STPST↓ input or by the execution of a HALT instruction by the CPU.

2.2 INTERNAL CPU ARCHITECTURE

The CP 1600 is a general register oriented, central processing unit consisting of eight 16-bit registers. These registers may be used as accumulators or as addressing pointers to locations external to the CPU. R7 and R6 have special characteristics which distinguish them from the other registers. R7 serves as the program counter (PC) and points to the next instruction to be executed. R6 serves as a stack pointer (SP) which points to the next available location in the "last-in, first-out" stack maintained anywhere in main memory.

A. Basic Functional Blocks

A basic internal block diagram of the CP 1600 showing major subsystems and data flow is shown in Figure 2-2. It is noted that all data transfers within the machine are performed on 16-bit words processed in two 8-bit bytes. Internally, the architectural effect of this byte serial technique is to organize all data paths as 8 bits in width. In communicating with external devices, however, the CPU transfers 16 bits in parallel.

The major logic blocks through which all data flow are the Arithmetic Logic Unit (ALU), Shifter, Register Array, and the Internal 16-bit Bus.

The ALU is the primary data processing element of the CPU. It accepts two data words each 8 bits wide from the internal 16-bit bus and produces an 8-bit resultant.

The ALU passes the resultant to the Shifter which is also 8 bits wide. The Shifter can transfer data to the register array transparently or can skew data one bit position right or left. The Shifter in combination with the ALU can effect changes in the four status bits which monitor CARRY OUT from the most significant bit of the resultant, ARITHMETIC OVERFLOW, SIGN DETECT, and ZERO DETECT.

The Register Array is comprised of the eight 16-bit registers with one 8-bit write port and two 8-bit read ports. The write port can direct data to any one of the eight registers and then either to that register's lower or upper 8-bit byte. The read ports can simultaneously output to the internal bus any two byte combination, be they from the same register, different registers, left or right byte.

The Internal 16-bit Bidirectional Bus is the primary link for all information transfers between the CPU and the external world. Communications between the internal bus and the external bus are governed by the bus control signals BC1, BC2, and BDIR.

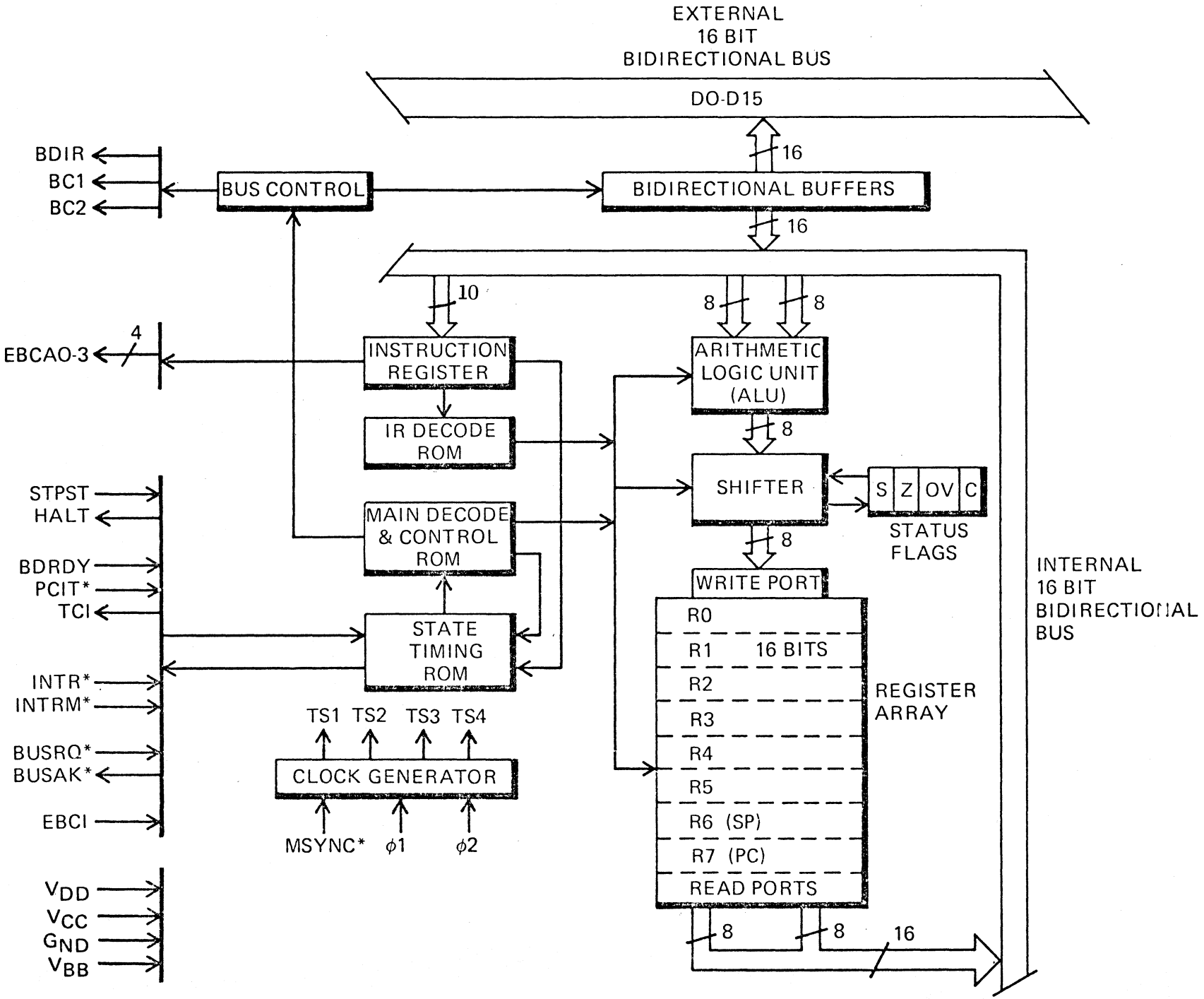


FIG. 2-2 INTERNAL BLOCK DIAGRAM

B. Processor Timing

The sequential operation of the CPU is governed by an internal State Timing ROM. This network generates next addresses to the Main Decode and Control ROM. It maps these addresses from the Instruction Register contents and makes next address decisions based on conditions in the machine and/or the present microinstruction. The Main Decode and Control ROM generates 21-bit microinstructions or micro states during each processor cycle. These micro states are grouped into various Fetch states, Address states, Execute states, Wait states, states which control the Interrupt and Bus Request Sequences, and others. Each micro state time is defined as four internal time slots - TS1, TS2, TS3, TS4 - as generated by the Clock Generator from the external non-overlapping clocks. See Figure 2-3.

If the CPU is driven by a 5 MHz, two-phase clock system as shown, then the duration of each micro state is 400 nanoseconds.

The State Timing ROM operates in parallel with the data processing logic thus keeping the Control ROM synchronous with the data flow operations. Thus, during any internal state time, the Control ROM and the Instruction Register Decode ROM are dictating the data processing functions in the ALU, Shifter, and the Register Array.

The CP 1600 achieves high internal processing speed through the use of microinstruction overlapping and data pipelining (Fig. 2-4). During each 400 nanosecond cycle, a 21-bit control word is executed while the next address decision is made and the new control word accessed. In addition, the Register/ALU data loop is pipelined so that two sets of byte pairs are being processed during each cycle . Before the result of an ALU operation is complete and returned to the Registers, the next operation on new data has already started.

In order to achieve this "data streaming" effect, the microinstruction word of the processor is divided into three major fields (Fig. 2-5). The first controls the top of the pipe which includes the READ PORTS and front end of the ALU logic. The second controls the bottom of the pipe which includes the lookahead carry network, the final part of the ALU logic and shifter, and the rewrite to the registers. The third field controls the selection of the next control state.

Finally, each state has its own unique pattern of bus control signals (BC1, BC2, BDIR). For example, in the first fetch state, the bus controls signals will be decoding BAR. In the next, they decode NACT, and, in the next, they decode DTB.

C. State Timing

Although it is not necessary for the user to understand the exact sequencing through the internal microcoded states, it is important to understand

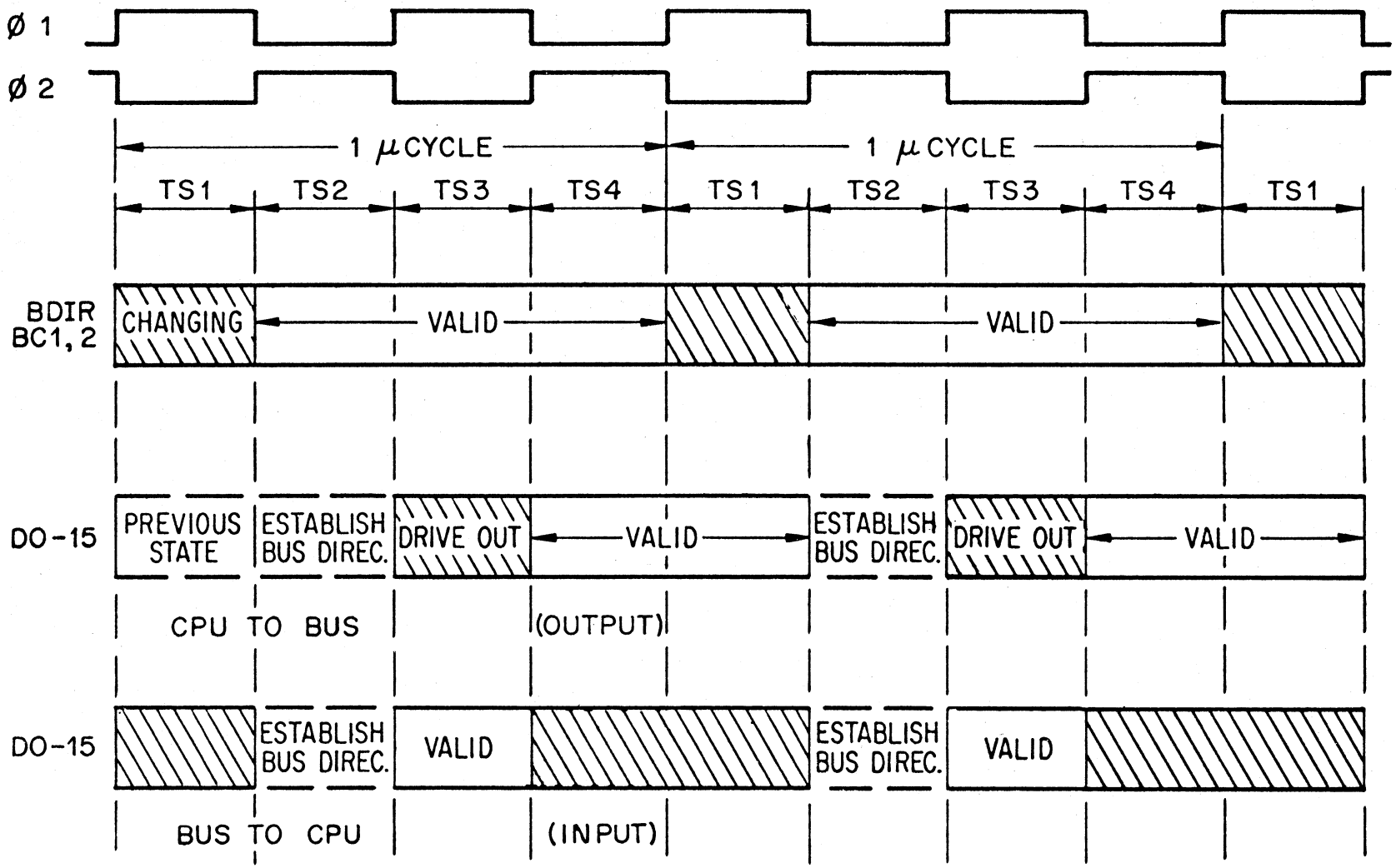


FIG. 2-3 BIDIRECTIONAL BUS TIMING

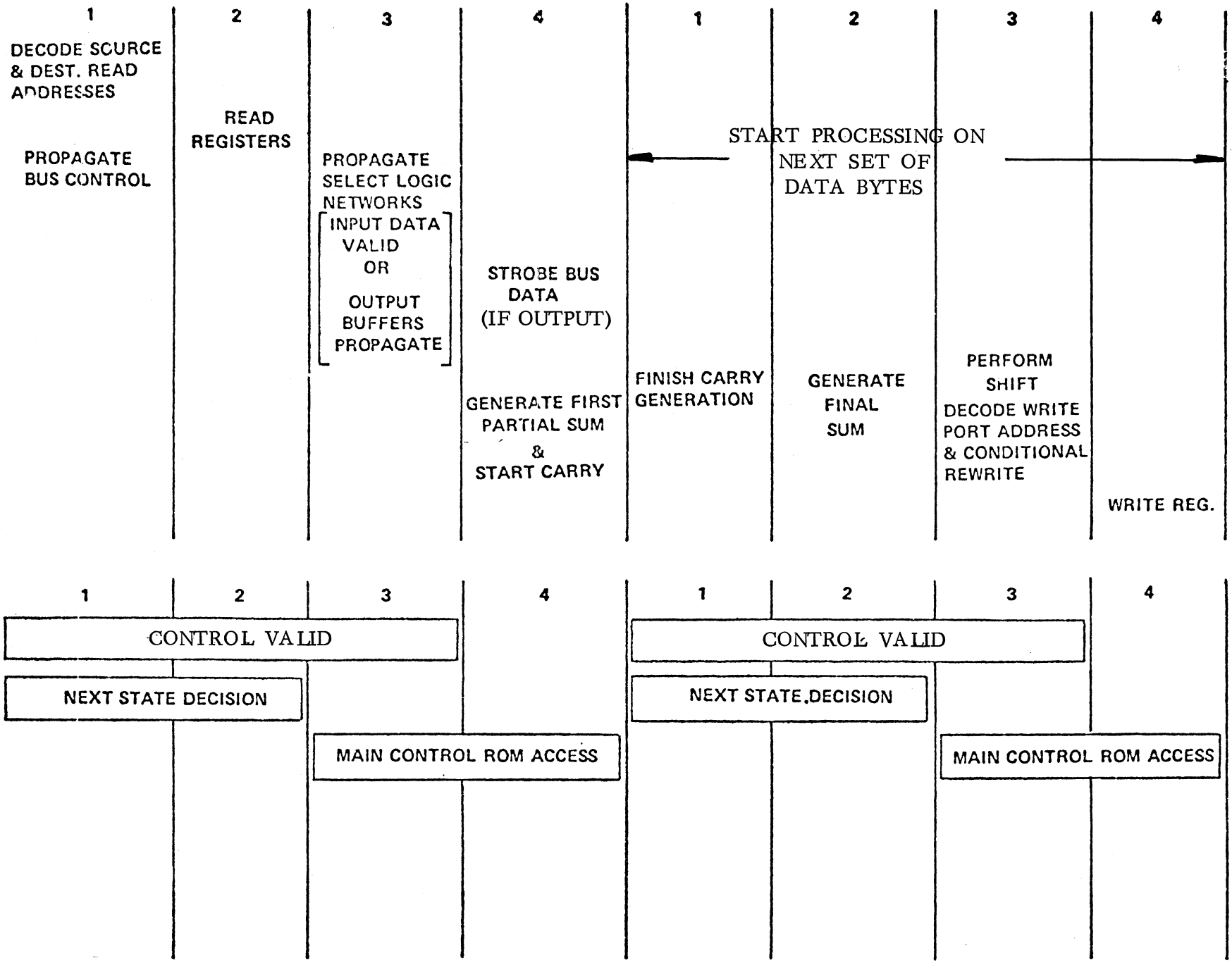


FIG. 2-4 PIPELINE TIMING

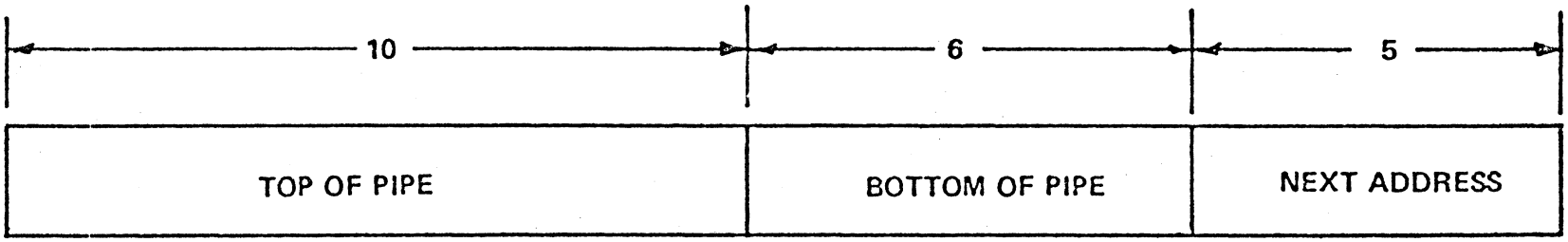


FIG. 2-5 MICROINSTRUCTION FORMAT

the general state flow and how it relates to the execution of instructions and their interaction with the various interfacing signals like BDRDY, BUSRQ*, INTR* and others. A basic state flow diagram is shown in Fig. 2-6.

There are six major state sequences through which the CPU passes: Initialization, Fetch, Address, Execute, Bus Request, and Interrupt Request sequences. Each of the sequences consist of one or more micro states (indicated by rectangular blocks). However, the decoded bus control signal in parenthesis indicates only the primary bus function to be performed during that functional operation. (Recall there is a unique set of bus control signals for each microcode state). Illustrations of the exact timing during each of these sequences is given in the "Pin Timing Diagrams" section.

C.1 Initialization Sequence

When power is first applied to the CPU, the MSYNC* signal must be at an active (low) level. This will force the CPU to an inactive state in which D0-D15 are in a high impedance state with the bus control signals issuing NACT. All other signals are in their inactive state and the interrupt system is disabled. All eight internal registers and the status bits contain arbitrary, undefined data. When MSYNC* goes inactive (high), the bus control signals issue IAB, and the CPU inputs from the bus into the PC the starting address of the main program. Note that the initialization address can be defined by the user at any desired bus address or can be the default address resulting from the logical state of the non-driven bus.

C.2 Fetch Sequence

The Instruction Fetch sequence will be entered after the initialization of the CPU, or after the execution of the previous instruction and the resolution of pertinent bus and interrupt requests. During the first microstate, the bus control signals issue BAR. During this state, the CPU will output the current contents of the PC onto the bus. BAR will strobe this data from the bus into the address register to fetch the instruction pointed to by the PC. Internally, the CPU increments PC and returns it to R7.

Beginning in the next micro state, the CPU will sample the BDRDY input line. If BDRDY is at a logic "1" level, the CPU will proceed to the next state where the fetched instruction will be inputted into the CPU via the signal DTB. If BDRDY is at a logic "0" level, the CPU will enter a WAIT state and remain in this state until BDRDY is brought to a logic "1" level. This WAIT state cannot be of indefinite duration; it must last no longer then 40 microseconds in order to preserve the dynamic characteristics of the CPU during this fetch sequence. The bus control signals will change to NACT throughout the WAIT state. External systems will need to utilize this WAIT state when their access times cannot meet the CPU speed requirements. As long as the CPU is in the WAIT state, it will sample the BDRDY line every microcycle (every 400 nanoseconds if the clock input is 5 MHz).

Once BDRDY is high, the CPU will issue the DTB bus signal. The external system presently addressed will gate data (in this instance the fetched instruction word) onto the bus. The instruction will be inputted to the CPU and loaded into its instruction register.

BASIC STATE FLOW DIAGRAM

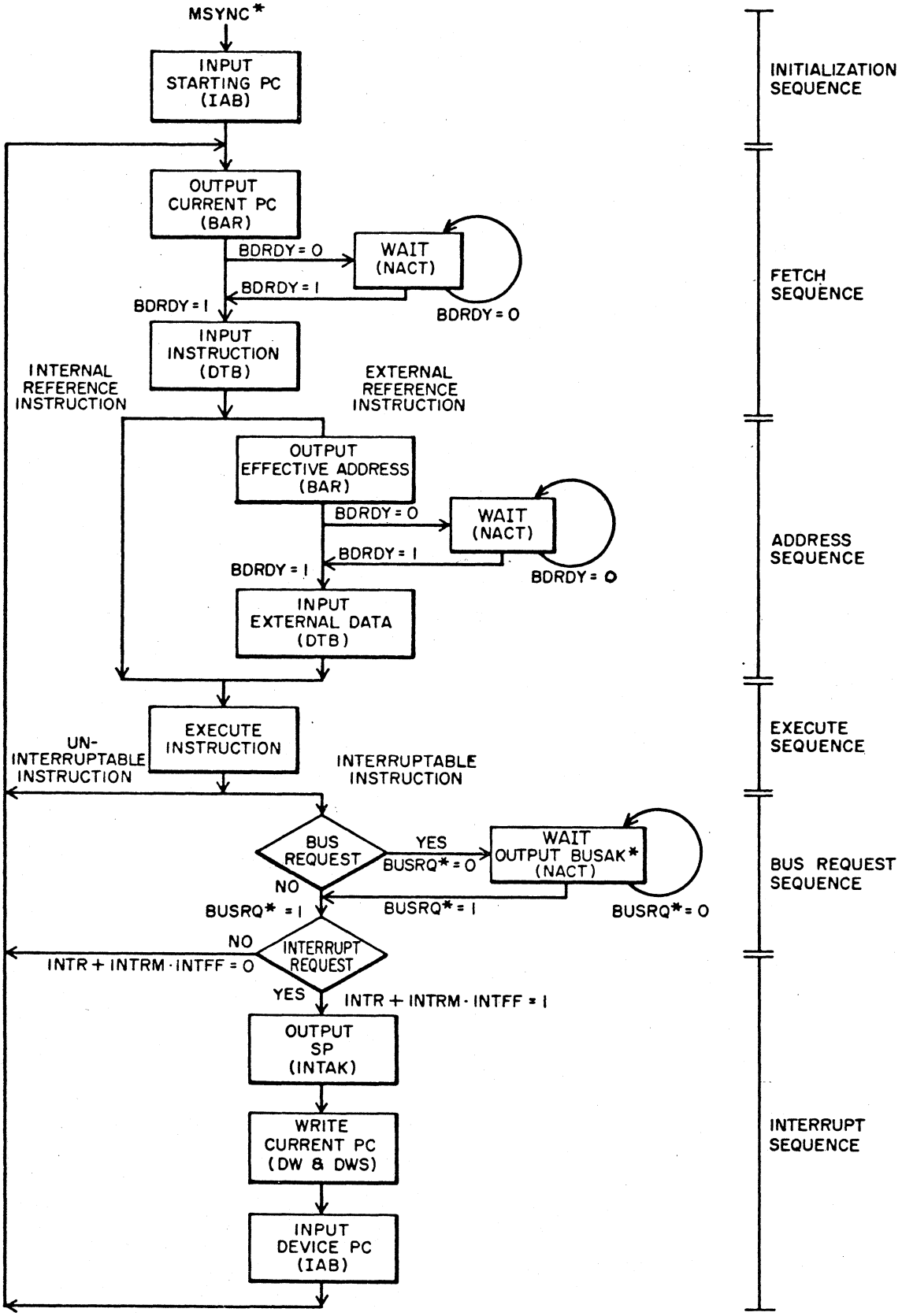


FIG. 2-6 INTERNAL FLOW DIAGRAM

C.3 Address Sequence

The Address Sequence will only be entered if the instruction just fetched is of an External Reference type. The distinctions between external and internal reference types will be made clearer in the chapter "Instruction Set". Basically, External Reference instructions need to retrieve external data to complete execution of the instruction. For example, those instructions which specify the contents of a register to be the address of an operand are all of external reference type. Internal Reference instructions process operands already held within the registers of the CPU.

The Address Sequence is very similar to the Fetch Sequence. The CPU will begin by issuing BAR and outputting some computed effective address on the bus. The particular instruction will determine how the effective address is computed. The effective address will be gated into the address register. The CPU will then sample BDRDY exactly as before. Again, if a WAIT state is entered, its duration must be less than 40 microseconds. Once BDRDY has been resolved, the CPU will issue a DTB which will gate the retrieved data onto the bus. The CPU inputs this data which will be used as an operand in executing the instruction.

Note: This sequence becomes slightly more complex if the instruction specifies direct addressing. The CPU will issue BAR, sample BDRDY, then issue ADAR to readdress memory, sample BDRDY again, and finally issue DTB.

C.4 Execute Sequence

After the data operands have been obtained, the CPU will execute the instruction. Execution can take as few as two microcode states or as many as six for the most complex instructions. Different types of bus control signals are activated according to instruction type. Examples are shown in the following table.

Instruction Class	# of Micro States	Bus Control Signals
External Reference		
Input Group	2	DTB, NACT
Move Out	3	DW, DWS, then NACT
Double Byte Data	3	BAR, NACT, and then DTB
Internal Reference		
Register - Register	2	NACT
Shift	2 or 4	2 or 4 NACT

C.5 Bus Request Sequence

If there are no Bus Requests or Interrupt Requests following the execution of the present instruction, the CPU will return to the Fetch Sequence and fetch the next instruction from memory.

If some external device has activated the BUSRQ* input line (by pulling it low) prior to the completion of the execution of the current interruptable instruction, the CPU will honor the Bus Request by causing the output signal BUSAK* to go low. The CPU will enter a WAIT state and remain in this state until BUSRQ* is brought to a logic "1" level. This WAIT state can be of indefinite duration. BUSAK* will be low for as long as the CPU is in this state and the CPU will not use the bus; the bus control signals will be issuing NACT. All other signals will be in their inactive state.

The BUSRQ* input signal can be activated asynchronously with respect to the CPU clocks. The CP 1600 digitally synchronizes BUSRQ* during every TS1 time.

C.6 Interrupt Sequence

The CPU will enter its Interrupt Sequence after all bus requests have been resolved and either of the following conditions have been met:

a) The INTR* input signal has been activiated, i.e., it has been pulled to a logic "0" level in sufficient time for the CPU to recognize the input as a valid interrupt request; or,

b) The INTRM* input signal has been similarly activated AND the internal interrupt flip-flop, INTFF, is set (enabled).

Once an interrupt request has been accepted by the CPU, the CPU performs the following functions automatically:

1. The first micro state will issue the bus control signal INTAK. The presence of this signal denotes acceptance by the CPU of the interrupt request and can be used by the peripheral devices to resolve interrupt priority. During this state, the CPU will output the current contents of Register 6, the Stack Pointer, to the Address Register in preparation for saving the program counter on the Stack.

2. The CPU will then output the PC as data to be stored and issue bus signals DW and DWS to write the Program Counter into the location pointed to by the Stack Pointer. That is, the CPU will store the PC into the top of the STACK. The Stack Pointer will be incremented after this store operation.

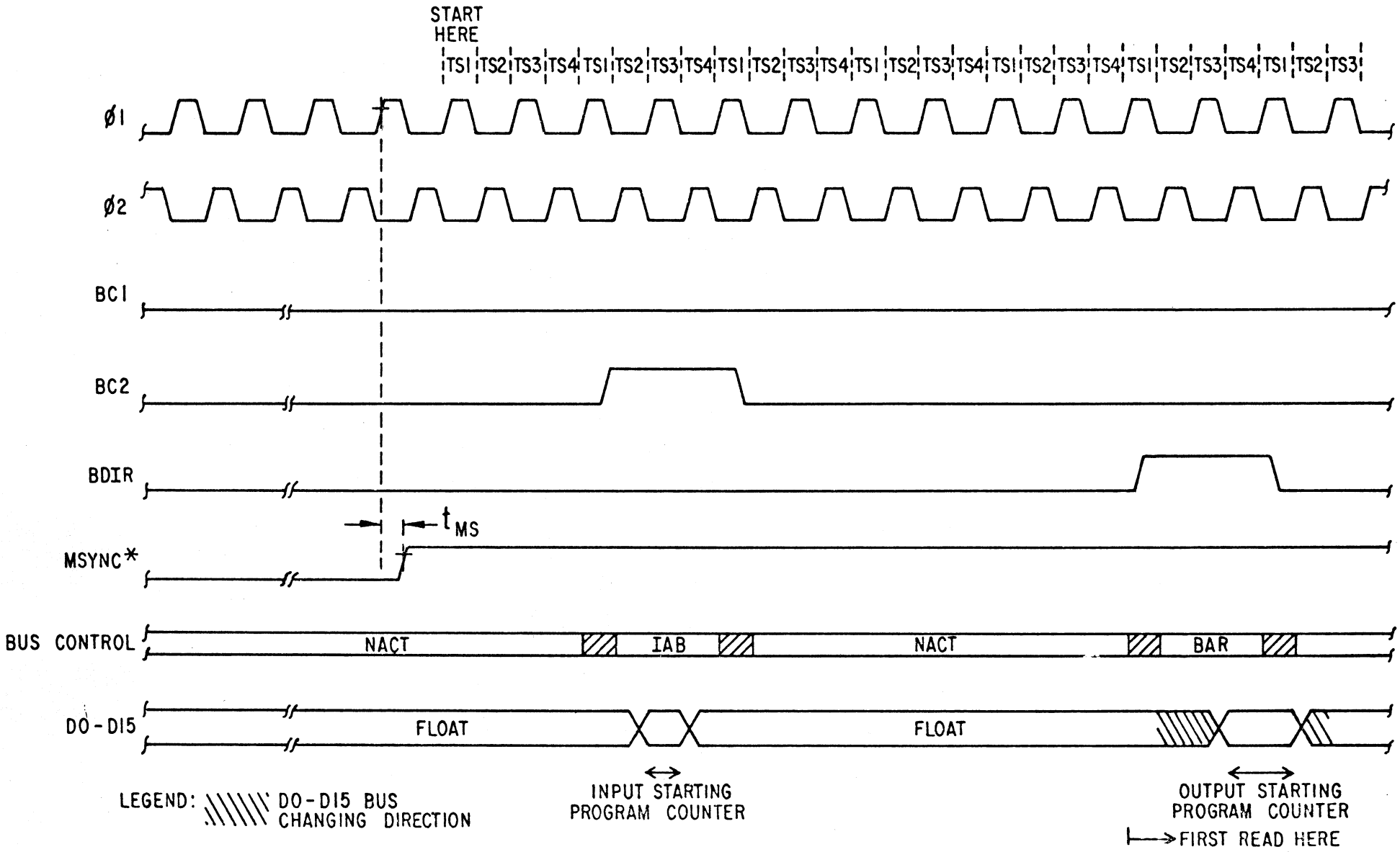
3. Finally, the CPU will issue bus signal IAB which will bring the starting address of the interrupt service routine into the program counter. The CPU will then return to the Fetch Sequence and execution continues in normal fashion from the new PC supplied by the interrupting device.

Again, in a manner similar to that of the BUSRQ* line, the CPU samples the INTR* and INTRM* inputs every TS1 time. In addition, the CPU will honor Bus Requests before it will honor Interrupt Requests.

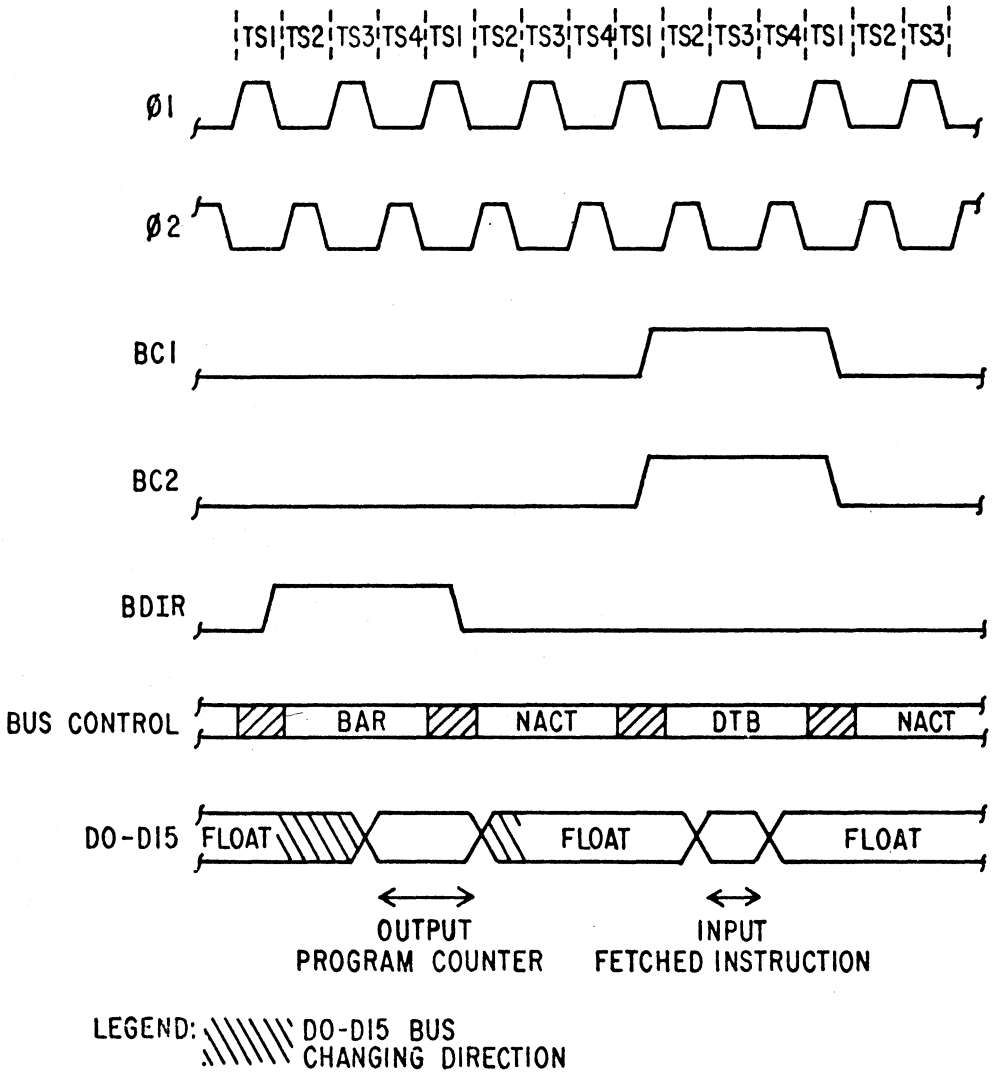
2.3 TIMING DIAGRAMS

The functional operation and relative timing of all CP 1600 sequences are shown in the following diagrams. These include:

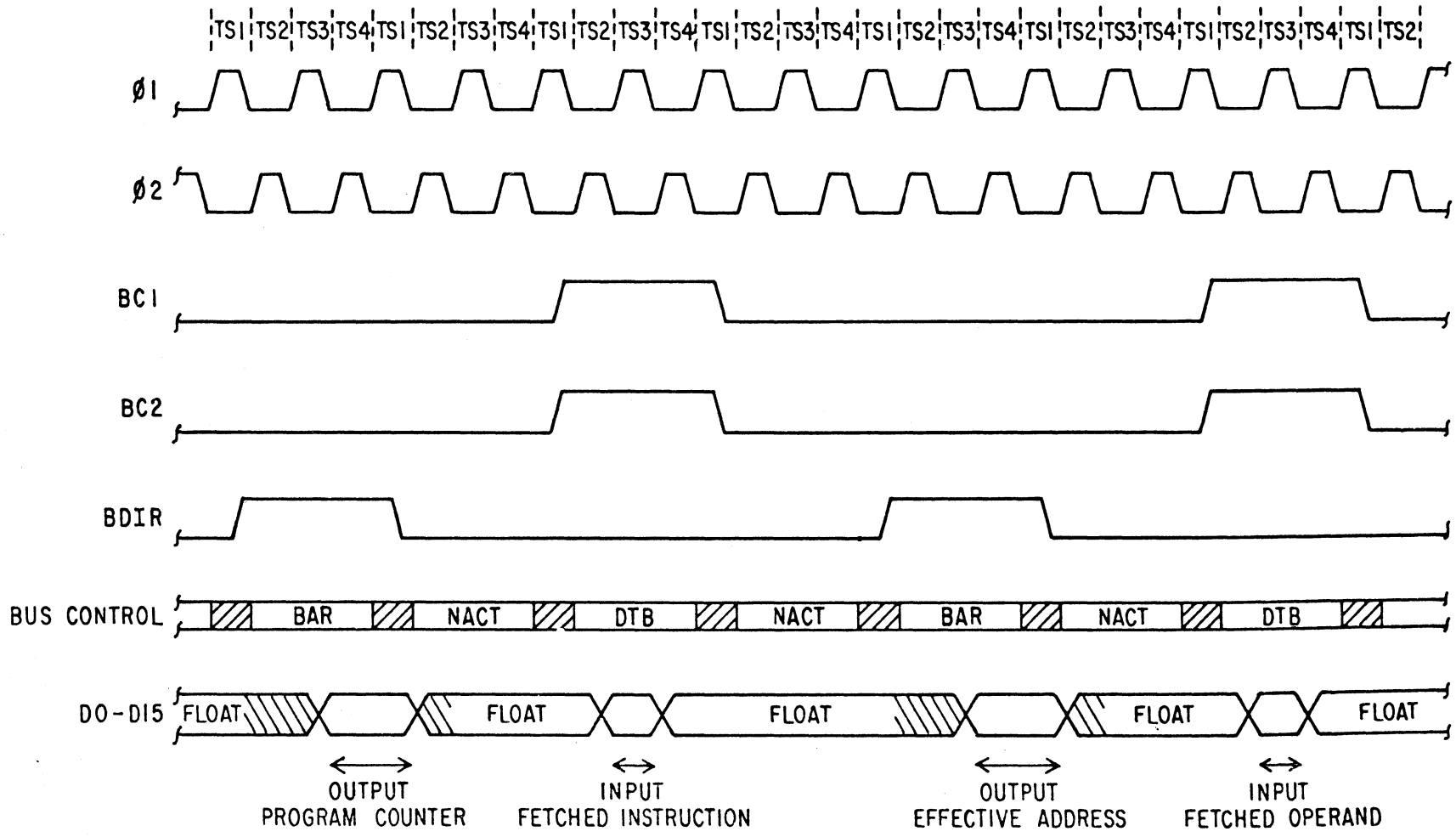
Initialization Sequence
Instruction Fetch
Instruction Fetch & Data Access
Write Sequence
External Condition Test Sequence
TCI Instruction Timing
PCIT * Pin Timing
Interrupt Sequence
BUSRQ/BUSAK Sequence
BDRDY Operation



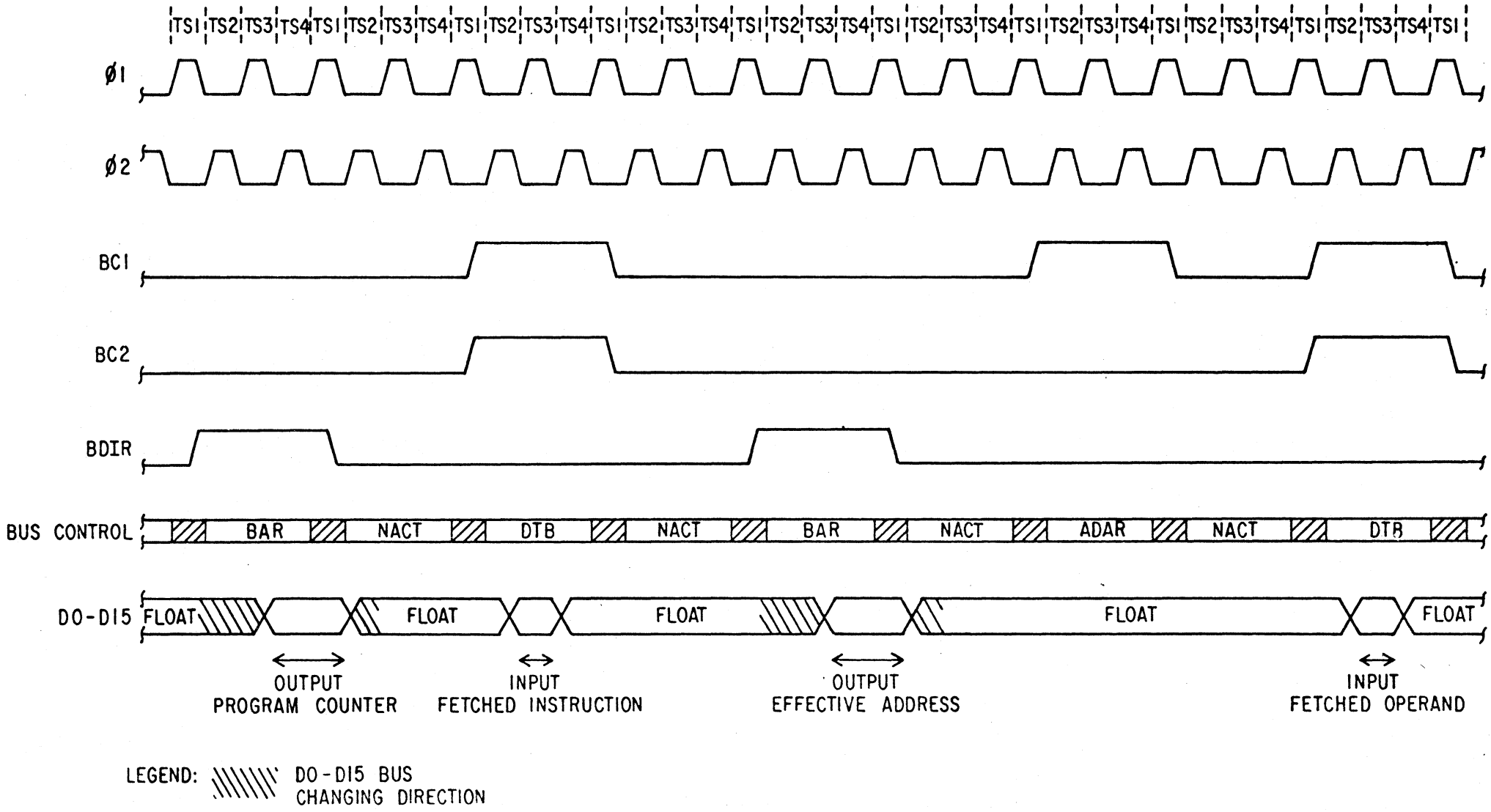
INITIALIZATION SEQUENCE



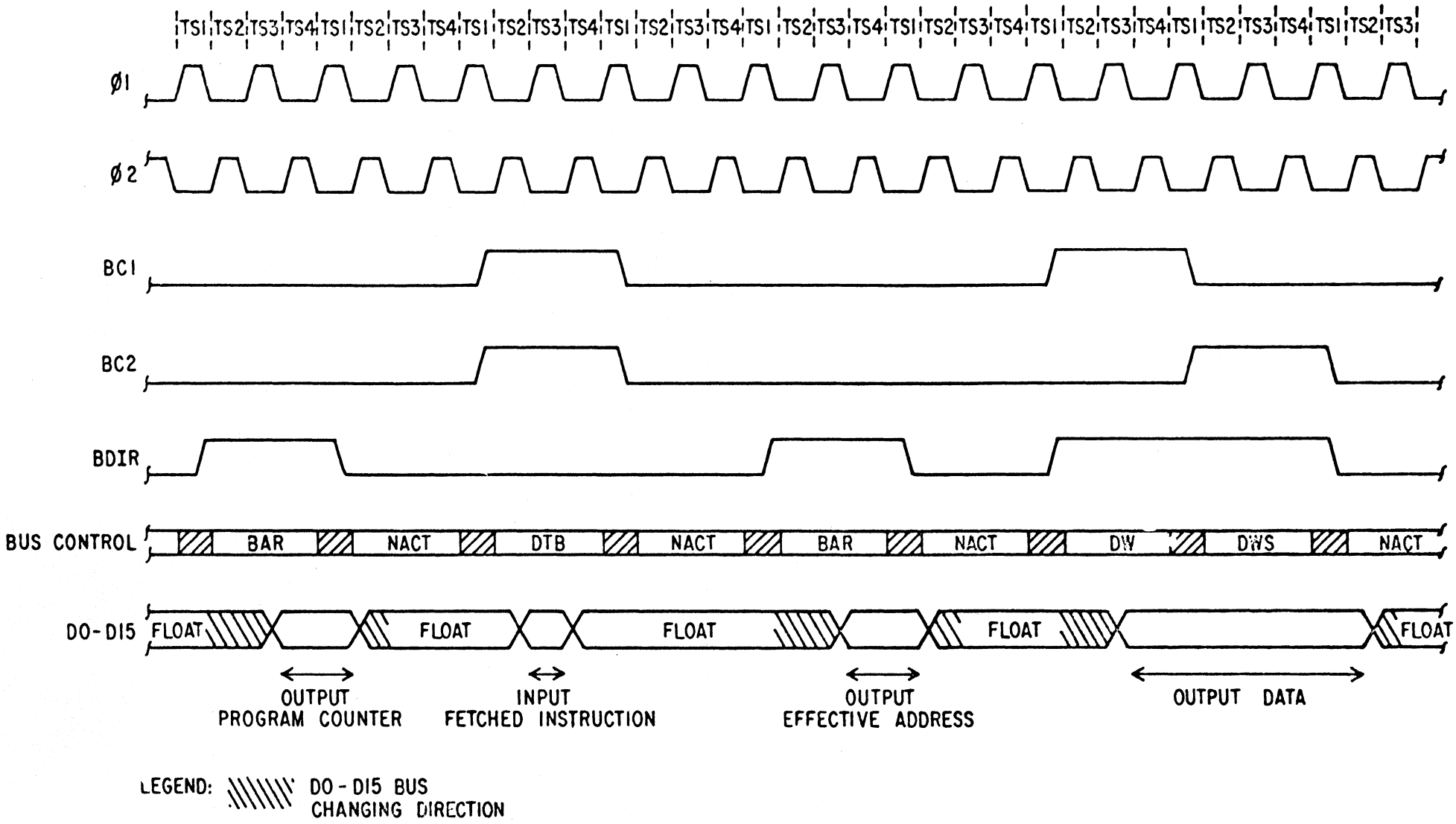
INSTRUCTION FETCH



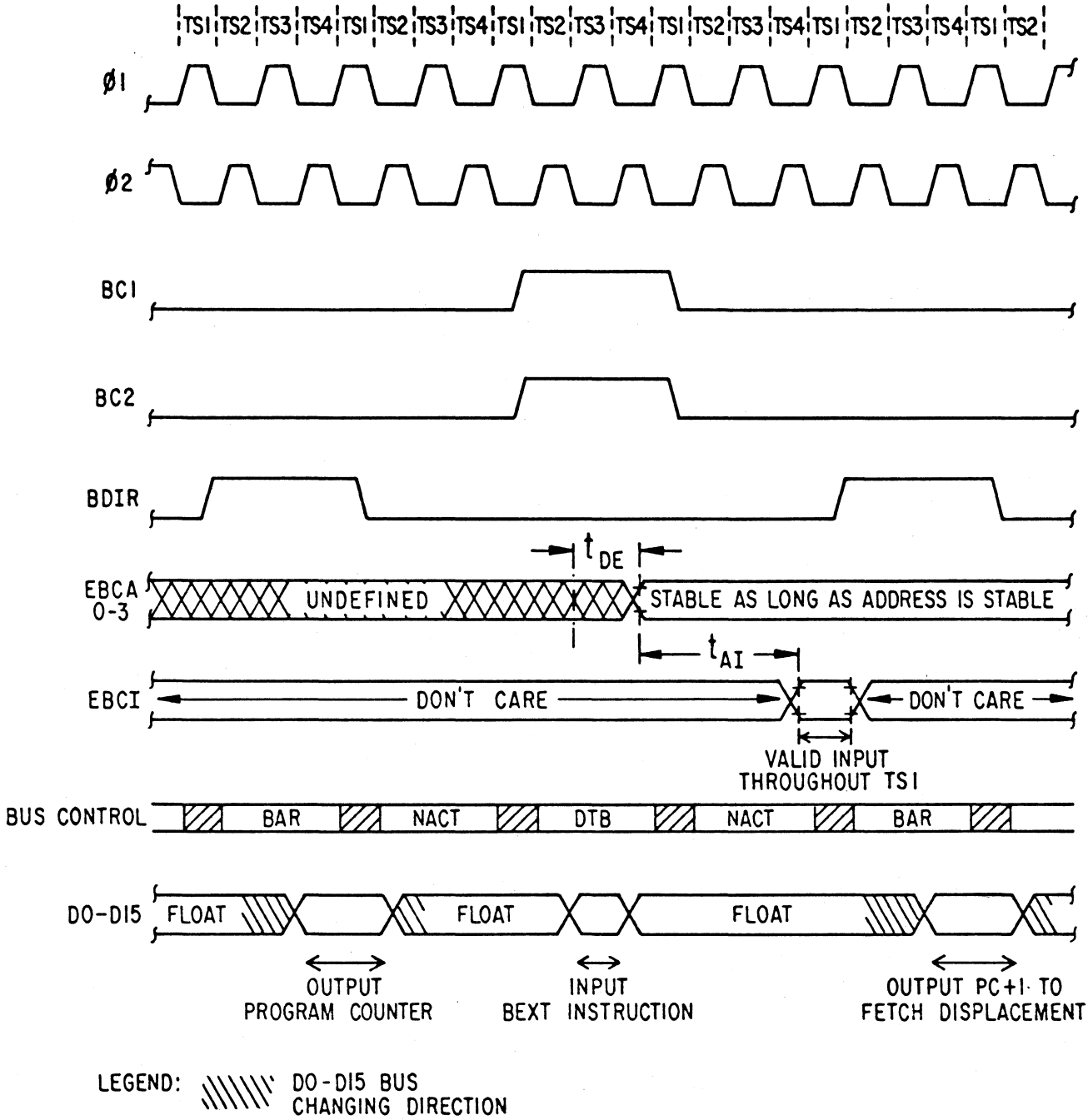
INSTRUCTION FETCH & DATA ACCESS
(INDIRECT REG. MODE)



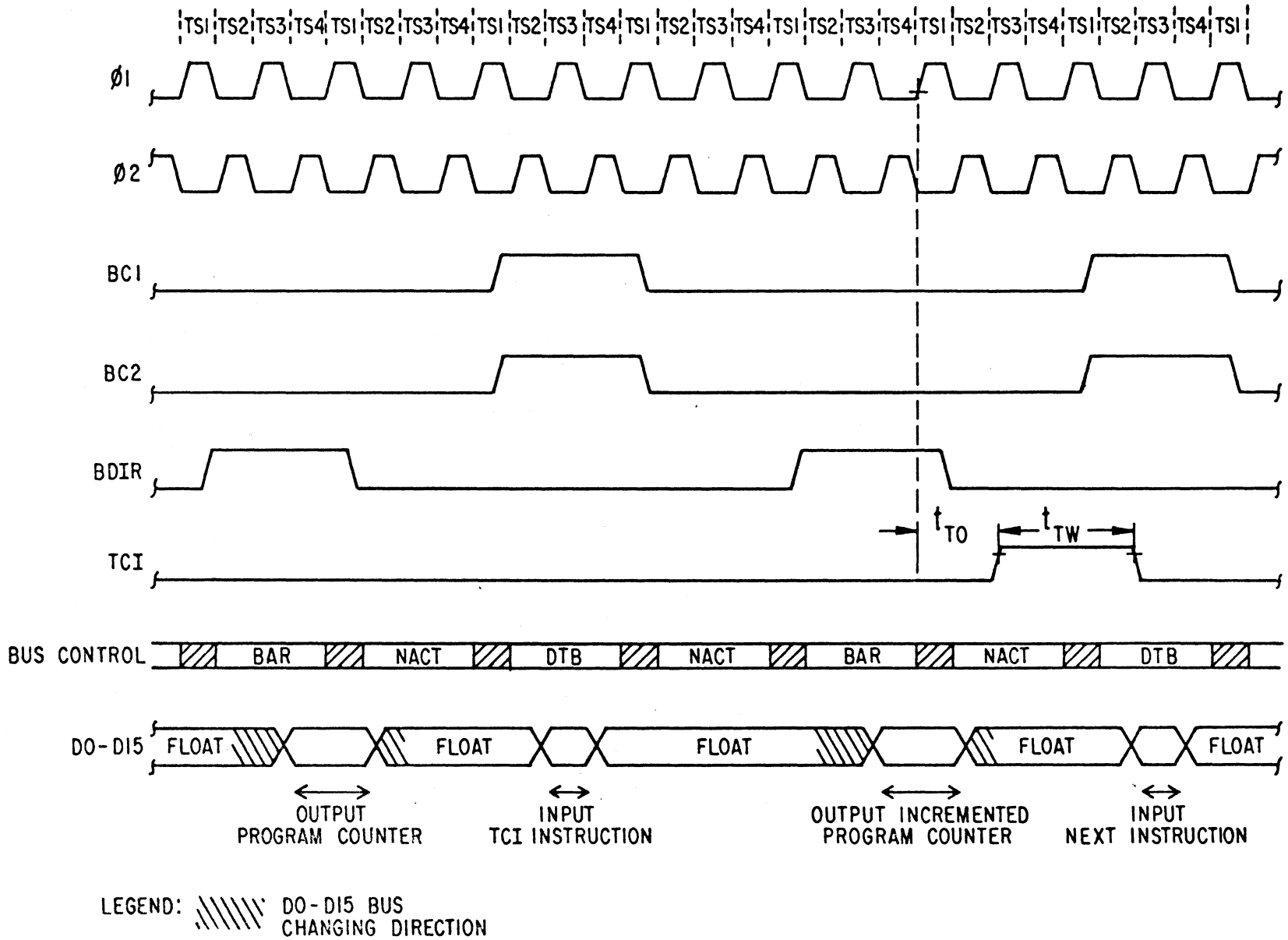
INSTRUCTION FETCH & DATA ACCESS
(DIRECT ADDR. MODE)



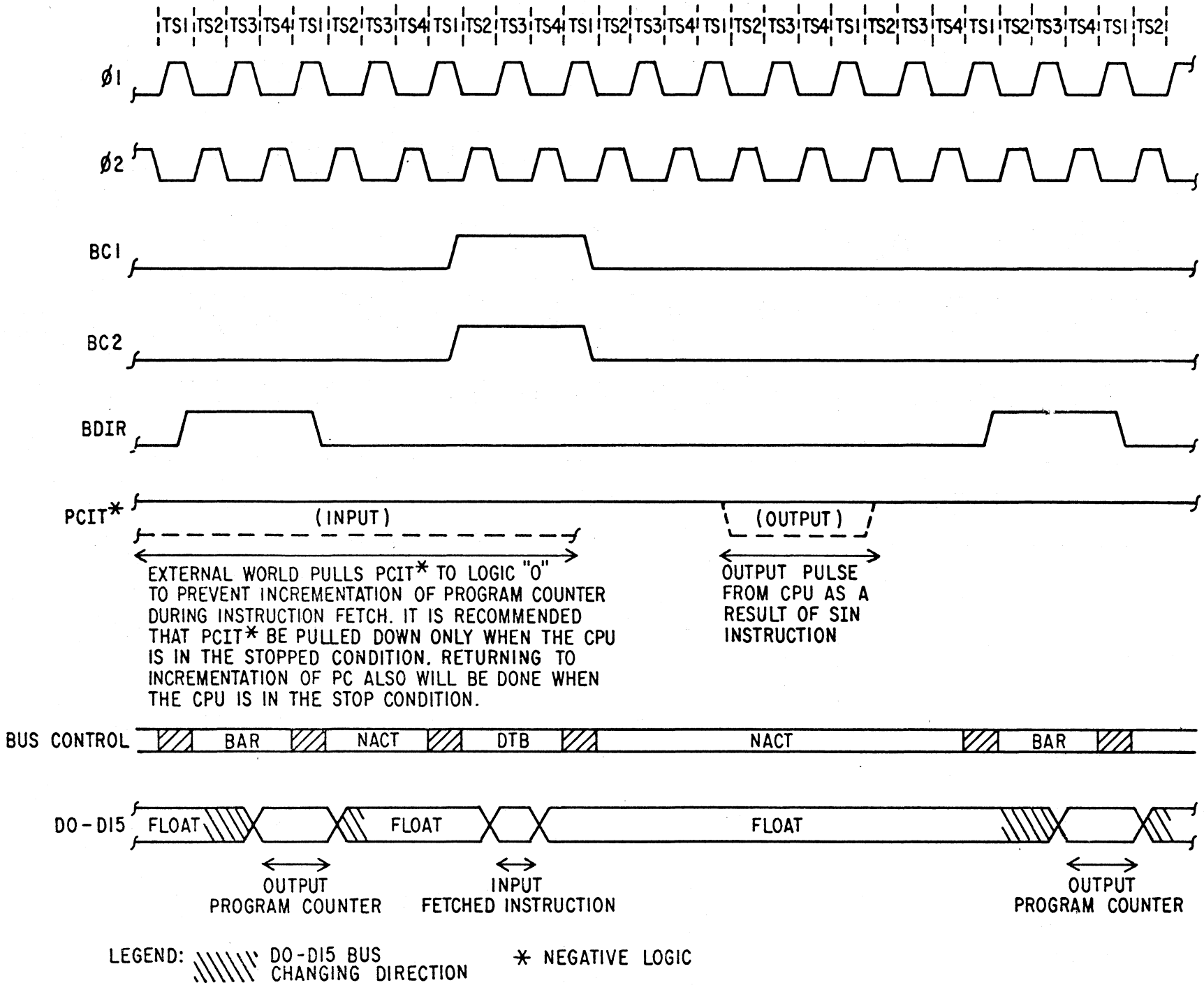
MVO TIMING (WRITE OPERATION)



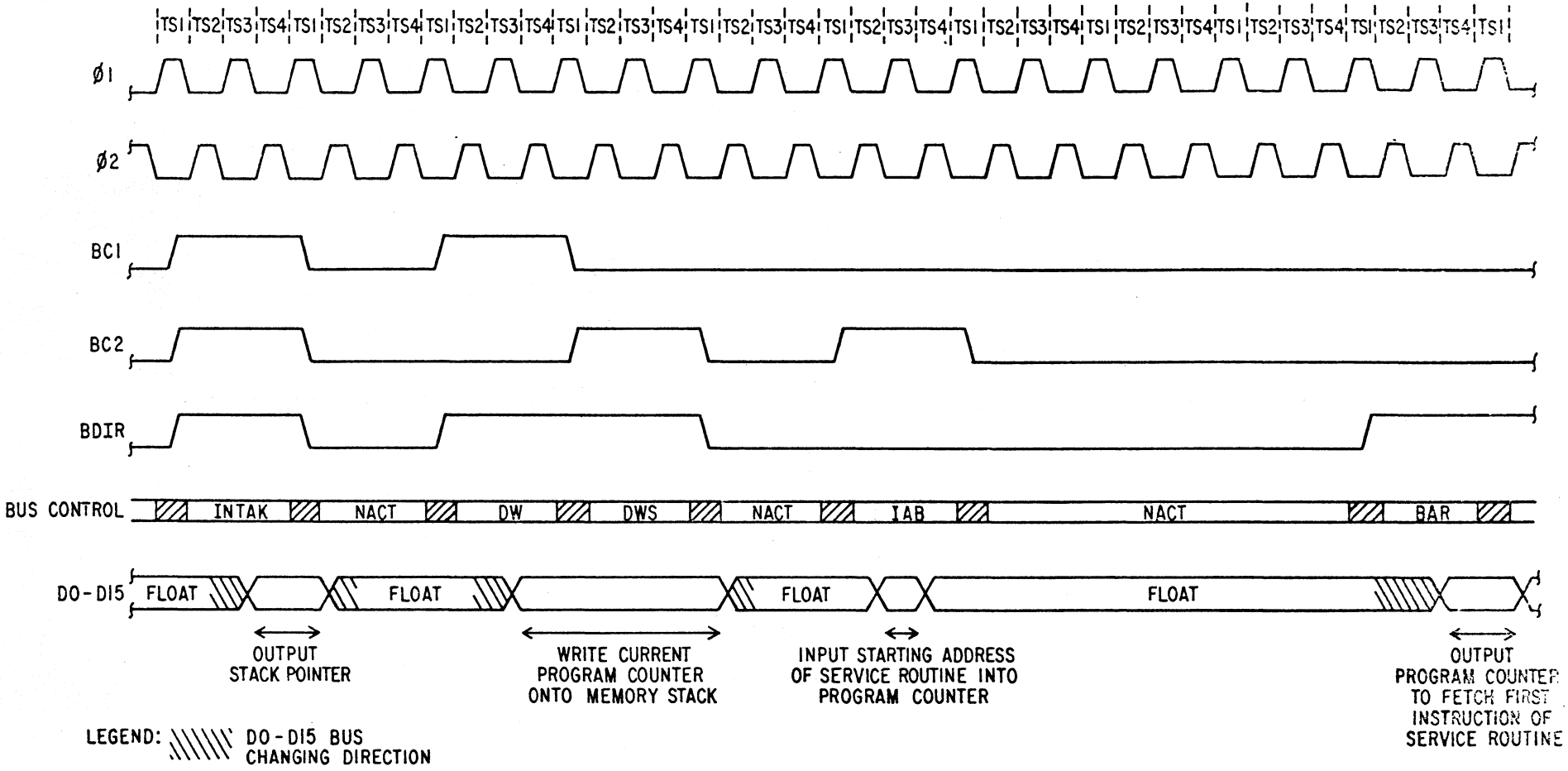
BRANCH ON EXTERNAL INSTRUCTION



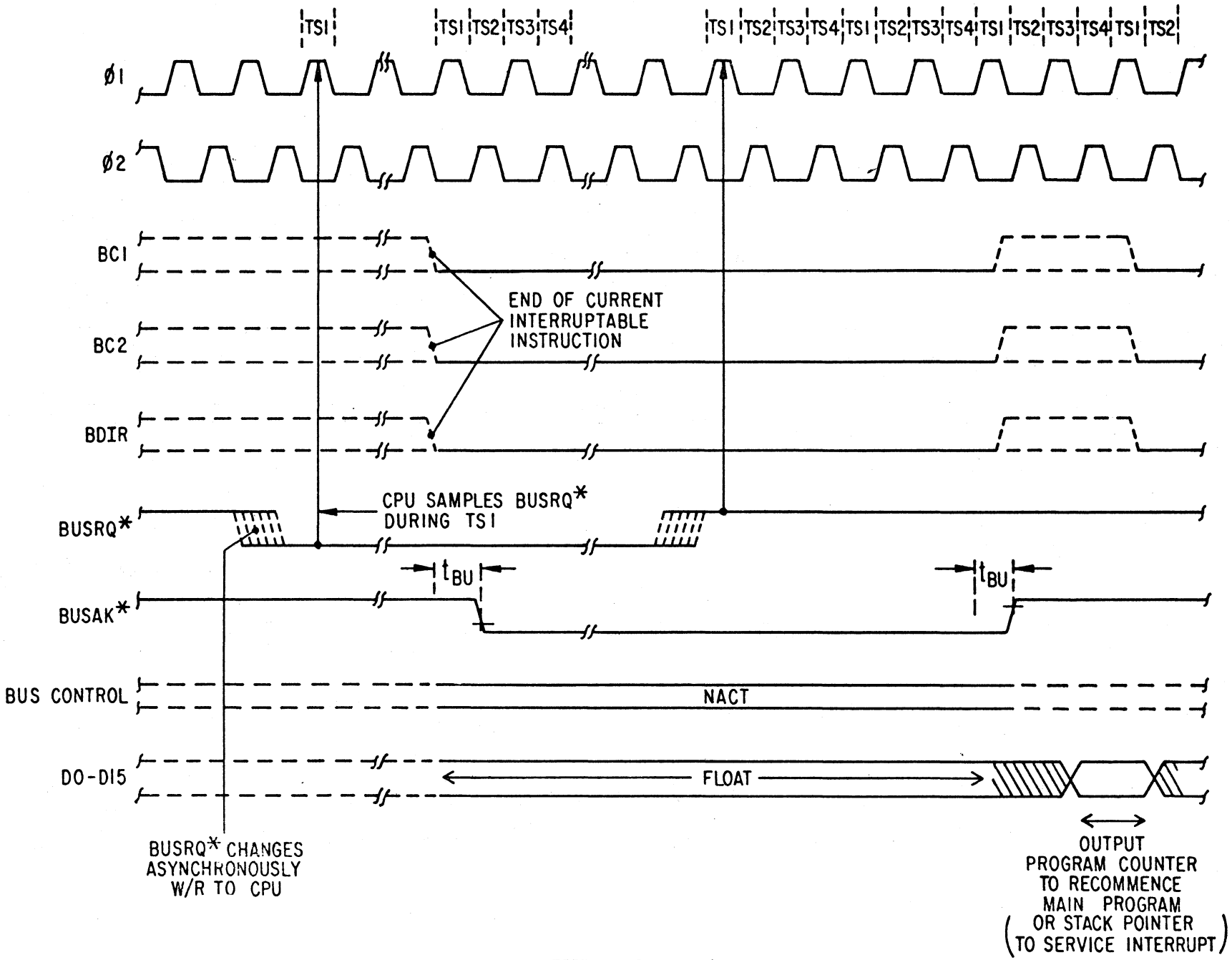
TCI INSTRUCTION TIMING


TS1 TS2 TS3 TS4 TS1 TS2 TS3 TS4 TS1 TS2 TS3 TS4 TS1 TS2 TS3 TS4 TS1 TS2 TS3 TS4 TS1 TS2 TS3 TS4 TS1 TS2 TS3 TS4 TS1 TS2
Ø1
Ø2
BC1
BC2
BDIR
PCIT*
(INPUT)
(OUTPUT)
EXTERNAL WORLD PULLS PCIT* TO LOGIC "0" TO PREVENT INCREMENTATION OF PROGRAM COUNTER DURING INSTRUCTION FETCH. IT IS RECOMMENDED THAT PCIT* BE PULLED DOWN ONLY WHEN THE CPU IS IN THE STOPPED CONDITION. RETURNING TO INCREMENTATION OF PC ALSO WILL BE DONE WHEN THE CPU IS IN THE STOP CONDITION.
OUTPUT PULSE FROM CPU AS A RESULT OF SIN INSTRUCTION
BUS CONTROL
BAR
NACT
DTB
NACT
BAR
DO-DI5
FLOAT
FLOAT
FLOAT
OUTPUT PROGRAM COUNTER
INPUT FETCHED INSTRUCTION
OUTPUT PROGRAM COUNTER
LEGEND:
DO-DI5 BUS CHANGING DIRECTION
* NEGATIVE LOGIC

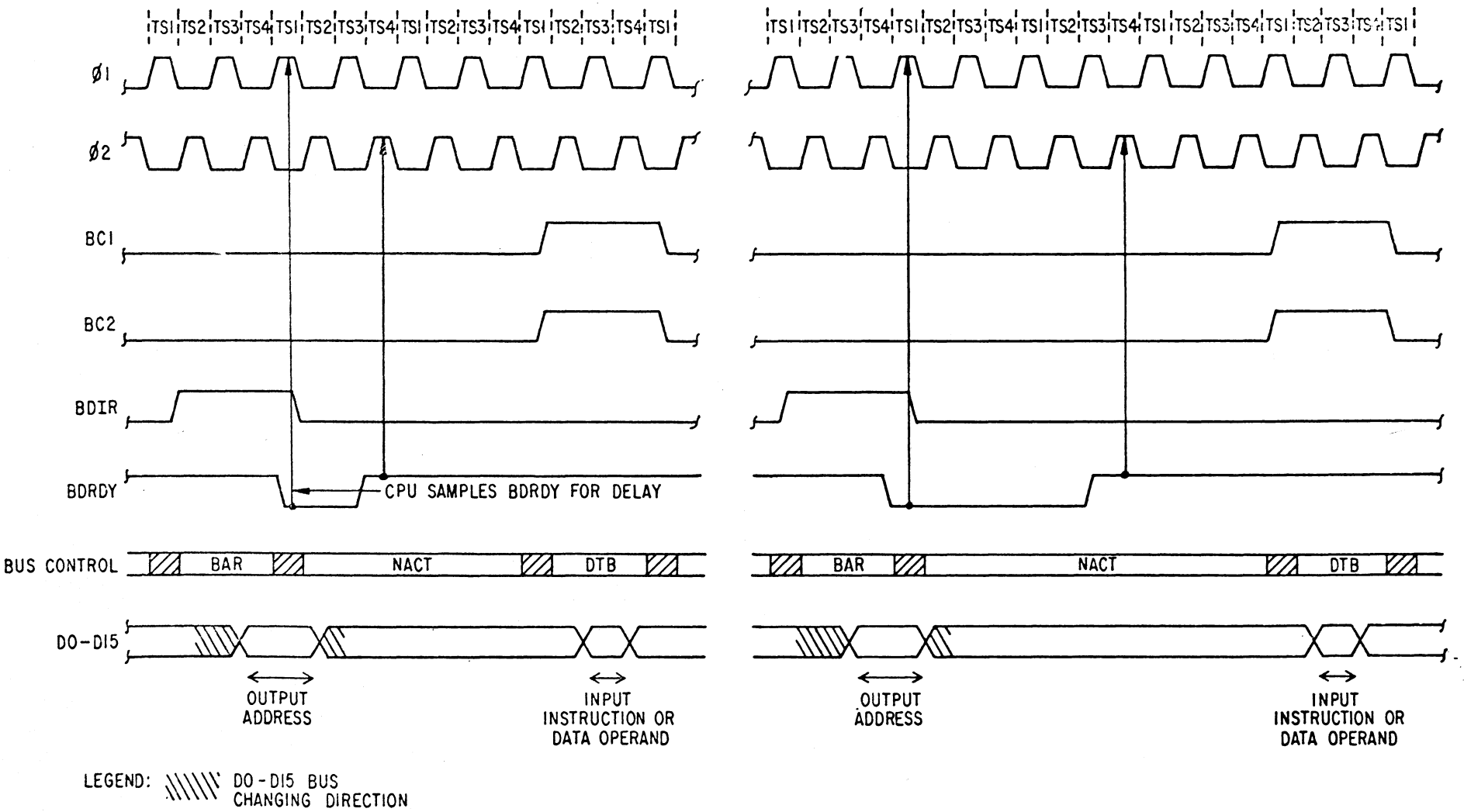
PCIT* TIMING



INTERRUPT SEQUENCE



BUS REQUEST/ACKNOWLEDGE SEQUENCE



BUS DATA READY (BDRDY) TIMING

2.4 ELECTRICAL SPECIFICATIONS (Tentative)

Maximum Ratings

VDD, VCC, GND, and all other input or output voltages with respect to VBB	-0.3V to +18.0V
Storage Temperature	$-55^{o}C$ to $+150^{o}C$
Temperature Under Bias	$0^{o}C$ to $70^{o}C$

Exceeding these ratings could cause permanent damage to the device.

Functional operation at these conditions is not implied - operating conditions are specified below.

Recommended DC Operating Conditions & Characteristics

All signals referenced to GND, unless otherwise noted.

$T_A = 0^{o}C$ to $70^{o}C$, unless otherwise noted.

	Symbol	Min.	Typ.	Max.	Units	Conditions
Supply Voltage	VDD	11.4	12.0	12.6	V	I_{DD}=70mA (typ)
	VCC	4.75	5.0	5.25	V	I_{CC}=12mA (typ)
	VBB	-2.7	-3.0	-3.3	V	I_{BB}=1mA (typ)
Clock Input High Voltage	VIHC	10.4		VDD	V	
Clock Input Low Voltage	VILC	0		.5	V	
Input Logic High Voltage	VIH					
D0-D15		2.4		VCC	V	
BDRDY		3.0		VCC	V	
All other		2.4		VCC	V	
Input Logic Low Voltage	VIL	0		.65	V	
Output High Voltage	VOH	2.4			V	IOH=100μA
Output Low Voltage	VOL					
D0-D15				.5	V	IOL=1.6mA
BDIR, BC1, BC2				.45	V	IOL=2.0mA
All other				.45	V	IOL=1.6mA

A. C. Characteristics

$T_A = 0^oC$ to 70^oC, $V_{DD} = 12V \pm 5\%$, $V_{CC} = 5V \pm 5\%$, $V_{BB} = -3 \pm 10\%$, unless otherwise noted.

All notations refer to the BUS TIMING DIAGRAM unless otherwise noted.

	Symbol	Min.	Typ.*	Max.	Units	Conditions
Pulse Width of Ø1	$t_{Ø1}$	70			ns	
Pulse Width of Ø2	$t_{Ø2}$	70			ns	
Clock Delay between Ø1 & Ø2	t_{12}	0			ns	
Clock Delay between Ø2 & Ø1	t_{21}	0			ns	
Clock Period	t_{cy}	.2		5.0	µs	
Clock Rise & Fall Times	tr, tf			15	ns	
D0-D15 Bus Delay from Ø1 (float to output)	t_{BO}			70	ns	1 TTL & 25 pf load
D0-D15 Bus Input from Ø2 (output to float)	t_{BF}			50	ns	1 TTL & 25 pf load
BC1, BC2, BDIR Output Delay from Ø1	t_{DC}			70	ns	1 TTL & 25 pf load
D0-D15 Bus Input Set-up Time before Ø1	t_{B1}	0			ns	
D0-D15 Bus Input Hold Time after Ø1	t_{B2}	10			ns	
MSYNC* Input Delay from Ø1	t_{MS}			30	ns	
BUSAK* Output Delay from Ø1	t_{BU}		150		ns	1 TTL & 25 pf load
TCI Output Delay from Ø1	t_{TO}		200		ns	1 TTL & 25 pf load
TCI Pulse Width	t_{TW}		300		ns	1 TTL & 25 pf load
EBCA Output Delay from input of BEXT instruction	t_{DE}			150	ns	1 TTL & 25 pf load
EBCA to EBCI Input Delay	t_{AI}			400	ns	

*at 25°C and nominal supply voltages

CAPACITANCE

$T_A = 25^oC$; $V_{DD} = 12.0V$, $V_{CC} = 5.0V$, $V_{BB} = -3.0V$

$t\emptyset 1 = t\emptyset 2 = 100$ ns

	Symbol	Min.	Typ.	Max.	Units
Ø1 Clock Capacitance	$C_{\emptyset 1}$		20	30	pf
Ø2 Clock Capacitance	$C_{\emptyset 2}$		20	30	pf
Input Capacitance	C_{in}				
D0-D15			6	12	pf
All other			5	10	pf
Output Capacitance	C_{out}				
D0-D15 bus high impedance state			8	15	pf

Notes on Some Timing Signals

1. BDRDY

The CPU samples the BDRDY input line every TS1 immediately following every micro state in which a BAR or ADAR bus control signal is issued. The following details apply:

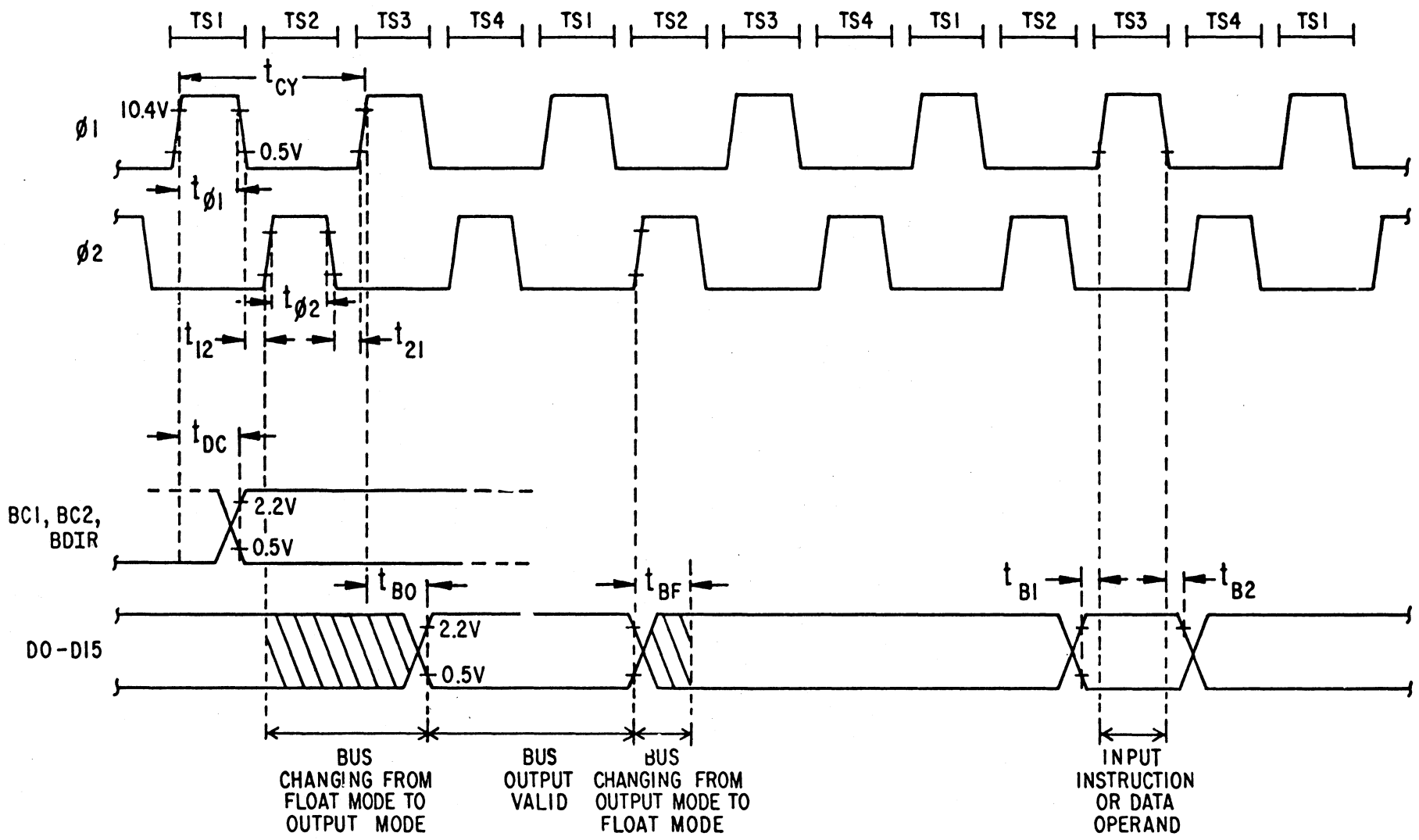
a) BDRDY must go to logic low requesting a wait state no later than 50 nsec into TS1. It will never go to logic low in any time slot other than TS1.

b) BDRDY must stay low for a minimum of 50 nsec.

c) BDRDY may go to a logic "1" level asynchronously to the CPU clocks. The CPU will synchronize this signal during TS4.

2. STPST↓

The STPST↓ signal is sensitive to high-to-low transitions. The width of the pulse at the low level is 200 nsec minimum, with a maximum frequency of 800 kHz.



BUS TIMING DIAGRAM

CHAPTER 3

CP-1600 INSTRUCTION SET

3.0 GENERAL INSTRUCTION FORMAT

The basic instruction word format for the CP 1600 microprocessor consists of 10 bits located in the low order bit positions of a 16-bit processor word. The high order 6 bits of every 16-bit word supplied to the CP 1600 as an instruction word are ignored by the internal microcontrol logic. This results in a compact, highly efficient 10-bit instruction word that can be implemented directly in a single 10-bit wide ROM instead of dual 8-bit wide ROMs for those systems requiring the ultimate in ROM bit efficiency. The 8 internal working registers, all data operands, displacement constants, and arithmetic and logical operations are implemented to full 16-bit significance providing the performance, flexibility, and efficiency of 16-bit operation.

The 10-bit instruction word is divided into 3 functional fields to which most instructions conform. This is shown in Fig. 3.0.1.

15 - - - - - 10	9 8 7 6	5 4 3	2 1 0
Future Use *	Operation	F1	F2

*This field is not implemented by the microcontrol logic in the CP1600.

FIG. 3.0.1

The most significant 4 bits of the CP 1600 instruction word is the Operational Field which provides 16 basic operations. The 16 operations are evenly divided into 2 groups of 8 instructions each; the External Reference group and the Internal Reference group.

3.1 INSTRUCTION DESCRIPTION

All Series 1600 symbolic instructions are expressed as statements containing an operator (OP Code) and in most cases one or two operands (Source/Address Mode and Destination). In general, the first three characters of an operator mnemonic represent the operation to be performed while the last character represents the addressing mode, such as ADDR for ADD Registers. The following description of instructions depicts both the symbolic format and also the binary translation. The timing, status register change, and instruction description is also shown.

The timing of each instruction is shown in terms of CPU cycle times. In addition to the basic number of cycle times for each instruction there are also potential wait times for every memory access associated with that instruction. Each wait time is dependent on the access time of the particular segment of memory being accessed. If the CPU cycle time is 400 nanoseconds and the memory speed is 1-700 nanoseconds, then W=0; if the memory speed is 701-1100 nanoseconds, then W=1; and if the memory speed is 1101-1500 nanoseconds then W=3; etc. For example, each instruction fetch has an associated wait time as does each word of a multi word instruction. If data is addressed during execution of an instruction, this additional memory access also has an associated potential wait time. Thus for an external reference instruction that is itself accessed from fast memory but its data is accessed from slow memory the following might hold:

$$\begin{aligned} \text{Number of Cycles} = 8 + 2W &= 8 + W + W \\ &= 8 + 0 + 2 \\ &= 10 \text{ cycles total} \end{aligned}$$

3.1.1 SYMBOLIC NOTATION

The following symbolic notation is used in all Series 1600 instruction documentation:

Address Modes:	R	- register
	'blank'	- direct address
	@	- indirect address
	I	- immediate data
Functions:	+	- addition
	-	- subtraction
	V	- inclusive OR
	$\not\vee$	- exclusive OR
	$\wedge$	- AND
	( )	- contents of
	$\leftarrow$	- is replaced by
	< >	- optional operand
Status:	S	- Sign bit
	Z	- Zero bit
	C	- Carry bit
	OV	- OVerflow bit
Operands:	SSS	- Source Register
	DDD	- Destination Register
	MMM	- Address Mode (register)
	R	- Last bit of Register address
	RR	- Register (0-3)
	RRR	- Register (0-7)
	N	- Number of Shifts (1 or 2)
	S	- Sign of Address Displacement
	EEEE	- External Condition Code (0-15)

3.1.2 OPERATION OF STATUS BITS

During arithmetic and logical operations in the CPU, the ALU Status Bits are used to monitor and record four characteristics of the resultant data. These bits are Carry (C) out of the ALU, arithmetic OVerflow (OV) from the ALU, Zero (Z) result from the output of the Shifter, and Sign (S) detect from the output of the Shifter. These bits operate in the following manner:

C - set if the result of an ADD, SUB, or CMP instruction produces a carry; cleared otherwise (SUB and CMP are performed by addition of two's complement of the subtrahend operand).

OV - set if the result of an ADD, SUB, or CMP instruction produces arithmetic overflow (two's complement notation); cleared otherwise (SUB and CMP are performed by addition of two's complement of the subtrahend operand).

Z - set if the result of an operation produces an all zero result; cleared otherwise.

S - set if the result of an operation contains a "one" in the high order bit position; cleared otherwise.

3.2 EXTERNAL REFERENCE INSTRUCTIONS

The External Reference instructions, which have a one in the high order bit position of the Operation Field, route data into or out of the CP 1600 to externally located bus addresses. Since a single address bus structure is utilized in the CP 1600 architecture, both memory and peripheral devices reside in the same Address Space. Only the system address allocation, as defined by the user, differentiates memory from I/O devices; therefore, no special input/output instructions are required and any External Reference instruction can access memory or peripheral devices. The CP 1600 utilizes a full 16-bit addressing structure which provides access to 65,536 unique bus locations.

3.2.1 DATA ACCESS INSTRUCTIONS

The format for 7 of the 8 External Reference instructions is shown in Fig. 3.2.1.

15 - - - - - 10	9	8 7 6	5 4 3	2 1 0
Future Use	1	Operation	Mode	Dest. Reg.

MoVe Out	(MVO)	001	000 -- Direct address in next memory location
MoVe In	(MVI)	010	001 -- Indirect address thru R1
ADD	(ADD)	011	010 -- Indirect address thru R2
SUBtract	(SUB)	100	011 -- Indirect address thru R3
CoMPare	(CMP)	101	100 -- Indirect address thru R4, R4+1→R4
logical AND	(AND)	110	101 -- Indirect address thru R5, R5+1→R5
eXclusive OR	(XOR)	111	110 -- Stack thru R6; Post Increment if Output; Pre-Decrement if Input
			111 -- Immediate data in next memory location

FIG. 3.2.1

These instructions use the lower 3 bits of the word as the register designator. This register is the source of data for output operations and the destination for data on input operations. The Mode Field of the instruction specifies the addressing mode to be used to access the Address Space. Mode 0 indicates that the contents of the next memory location is to be used directly as an external address. Modes 1, 2 and 3 indicate that registers 1, 2 and 3 respectively contain the external address. Modes 4 and 5 indicate indirect addressing through registers 4 and 5 with the contents of the register automatically incremented after use as the address; i.e., post increment. Mode 6 indicates stack operations, i.e., indirect through register 6 with the contents of the register post incremented for output operations (push) and pre-decremented for input operations (pull). The Stack is maintained by register 6 and expands upward in external memory from low address to high address. Mode 7 indicates that the next memory location holds data to be used directly; i.e., immediate data.

In order to utilize an efficient 10-bit instruction ROM and also reference 16-bit data from the same memory area, a Double Byte Data feature is available on the CP-1600. When a Set Double Byte Date (SDBD)

instruction preceeds an external reference instruction, two words of memory are accessed. The least significant data byte is obtained from the low order eight bits of the first word and the most significant data byte is obtained from the low order eight bits of the second word. These bytes are then combined by the processor to form a 16-bit data word. Double byte data may be accessed via indirect addressing modes 4, 5, and 7. If modes 1, 2 and 3 are used only one word is accessed with both bytes of data obtained from the lower eight bits of the same word. Modes 0 and 6 are not supported in the double byte data operation.

3.2.2 CONDITIONAL BRANCH INSTRUCTIONS

The eighth instruction in the External Reference group is the Conditional Branch type which has the format shown in Fig. 3.2.2.

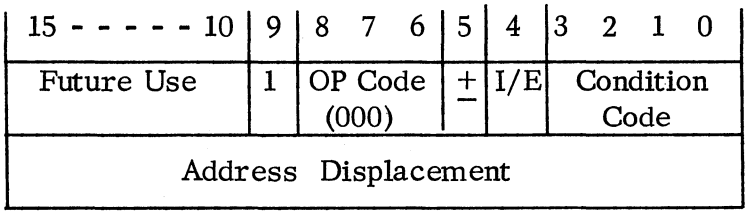


FIG. 3.2.2

This instruction tests any one of 32 conditions and causes a branch in the program if the test has a true result. The 32 conditions are evenly divided into 2 groups of 16 each. The Internal Condition Codes comprise various tests on the state of the status register bits of the ALU while the External Condition Codes comprise 16 tests on external digital status signals of the user's choice. (Bit 4 = 1 for external.)

In case of a false result, the program continues to the next sequential instruction. When the appropriate conditions are true, the instruction transfers program control to the specified destination. The actual destination address is determined by adding or subtracting the address displacement contained in the second word of the instruction to the program counter. If the destination direction bit of the instruction (±) is zero, then the displacement is added to PC+2. If it is one, then the displacement is one's complemented and then added to PC+2.

3.2.3 EXTERNAL REFERENCE ADDRESSING MODES

All of the instructions in the external reference group require that data be accessed from the external Address Space of the processor. The detailed sequence for each of the various addressing modes is shown in Fig. 3.2.3 to Fig. 3.27.

The effective address (EA) is used to access the first operand. The second operand is always contained in an internal CPU register with the result of the operation replaced in this same register. Note that the program counter (R7) is incremented before the Operation Code is performed; thus, if the destination register (DDD) specified by the instruction is R7, the incremented program counter is modified creating the effect of a branch in the program sequence.

DIRECT ADDRESS (Mode=000)

PC	Future Use		1	OP Code			0	0	0	D	D D
PC+1	a a a a a a		A	A	A	A	A	A	A	A	A A

PC ← PC + 2

EA ← aaaaaaAAAAAAAAAA

Note: aaaaaa is dependent upon the memory word size.

FIG. 3.2.3

INDIRECT ADDRESS (Mode=001 to 110)

PC	Future Use		1	OP Code			M	M	M	D	D D

MMM = 001, 010, 011	MMM = 100, 101
PC ← PC + 1	PC ← PC + 1
EA ← (MMM)	EA ← (MMM)
	(MMM) ← (MMM) + 1

MMM = 110 for MVO only	MMM = 110 for non-MVO
PC ← PC + 1	PC ← PC + 1
EA ← (MMM)	(MMM) ← (MMM) -- 1
(MMM) ← (MMM) + 1	EA ← (MMM)

FIG. 3.2.4

IMMEDIATE DATA (Mode=111)

PC	Future Use		1	OP Code			1	1	1	D	D D
PC+1	i i i i i i		I	I	I	I	I	I	I	I	I I

PC←PC + 2 EA←PC + 1

Note: i i i i i i is dependent upon the memory word size.

FIG. 3.2.5

INDIRECT ADDRESSED DOUBLE BYTE DATA

PC-1	Future Use		0	0	0	0	0	0	0	0	0 1
PC	Future Use		1	OP Code			M	M	M	D	D D

SDBD instruction

PC←PC + 1

If MMM = 1, 2, 3

EA_L←(MMM)

EA_H←(MMM)

If MMM = 4, 5

EA_L←(MMM)

(MMM)←(MMM) + 1

EA_H←(MMM)

(MMM)← (MMM) + 1

Note: The SDBD instruction must immediately preceed the external reference instruction. Mode 000 and 110 are not supported for double byte data operations by this version of the CP-1600.

FIG. 3.2.6

IMMEDIATE DOUBLE BYTE DATA

PC-1	Future Use		0	0	0	0	0	0	0	0	0 1
PC	Future Use		1	OP Code			I	I	I	D	D D
PC+1	Not Used						L	L	L	L	L L L L
PC+2	Not Used						U	U	U	U	U U U U

SDBD instruction

PC←PC + 3

EA_L←(PC+1)

EA_H←(PC+2)

Note: The assembler will automatically generate the SDBD instruction preceeding the external reference instruction when the constant specified by the assembly language statement exceeds the memory word size specified. If the user specifies the SDBD, then the data word will always be generated to fill two bytes.

FIG. 3.2.7

3.3 INTERNAL REFERENCE INSTRUCTIONS

The Internal Reference instructions, which has a zero in the high order bit position of the Operation Field, process data already held within the 8 working registers of the CP-1600. These instructions are grouped into register to register, register shift, single register, and internal control instructions. The basic formats are depicted in Fig. 3.3.1 to Fig. 3.3.5.

3.3.1 REGISTER TO REGISTER

15 - - - - - 10	9	8 7 6	5 4 3	2 1 0
Future Use	0	OP Code	S S S	D D D

OPERATION

(010)	MOVe Register	(MOVR)
(011)	ADD Register	(ADDR)
(100)	SUBtract Register	(SUBR)
(101)	CoMPare Register	(CMPR)
(110)	logical AND Register	(ANDR)
(111)	eXclusive OR Register	(XORR)

FIG. 3.3.1

3.3.2 REGISTER SHIFT

15 - - - - - 10	9	8 7 6	5 4 3 2	1 0
Future Use	0	0 0 1	B L A N	D D

MODE

B=1, Right Shift	B=0, Left Shift
L=1, With Link Bits	L=0, No Link Bits
A=1, Arithmetic	A=0, Logical or Rotate
N=1, 2 Position Shift	N=0, 1 Position Shift

Note: Shifts can only be performed on Registers R0 - R3.

FIG. 3.3.2

3.3.3 SINGLE REGISTER

15 - - - - - 10	9	8 7 6	5 4 3	2 1 0
Future Use	0	0 0 0	OP Code	D D D

OPERATION

001	INCrement Register	(INCR)
010	DECrement Register	(DECR)
011	COMplement Register	(COMR)
100	NEGate (2's complement) Register	(NEGR)
101	ADd Carry bit Register	(ADCR)
110	Get Status WorD register *	(GSWD)
111	Restore Status WorD register	(RSWD)

FIG. 3.3.3

3.3.4 INTERNAL CONTROL

15 - - - - - 10	9	8 7 6	5 4 3	2 1 0
Future Use	0	0 0 0	0 0 0	OP Code

OPERATION

000	HaLT	(HLT)
001	Set Double Byte Data	(SDBD)
010	Enable Interrupt System	(EIS)
011	Disable Interrupt System	(DIS)
101	Terminate Current Interrupt	(TCI)
110	CLeaR Carry Bit	(CLRC)
111	SET Carry Bit	(SETC)

FIG. 3.3.4

* The GSWD instruction can use only R0, R1, R2, or R3 as register specifications for true Get Status Word operation. Use of R4 or R5 results in a NOP instruction. Use of R6 or R7 results in a NOP internal to the CP-1600 and causes a pulse to be outputted on the PCIT* pin. This can be used to cause an interrupt to form a Software Interrupt (SIN) or Trap instruction.

3.3.5 JUMP/JUMP AND SAVE RETURN

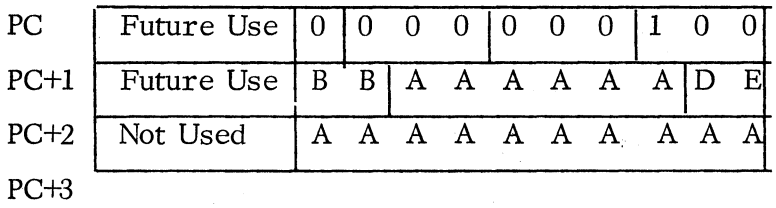


FIG. 3.3.5

The instructions in the jump group occupy three consecutive memory locations and are used to perform direct jumps and subroutine calls. In addition the interrupt system can be either enabled or disabled by bits 0-1 of the second word of the instruction.

On direct jumps the return address is not saved. The destination address is formed by using bits 2-7 of the second word as bits 10-15 of the jump address and using bits 0-9 of the third word as bits 0-9 of the jump address.

On subroutine jumps, the return address (PC+3) is saved in register 4, 5, or 6. This register designation is formed by using bits 8-9 of the second word (field BB) and adding 4. Note if BB=3, the designated register becomes 7 (the PC) which is the jump instruction. By using one of the Jump and Save instructions, the linkage to a subroutine is accomplished with the return address saved in register 1BB.

INSTRUCTION SET

REGISTER - REGISTER

MNEMONIC	OPERAND	CYCLES	INSTRUCTION			DESCRIPTION	STATUS CHANGE
MOVR	SSS, DDD	6 *	0010	SSS	DDD	MOVe contents of Register SSS to register DDD. *If DDD is 6 or 7 add 1 to Cycles.	S, Z
TSTR	SSS	6 *	0010	SSS	SSS	TeST contents of Register SSS. *If SSS is 6 or 7 add 1 to Cycles.	S, Z
JR	SSS	7	0010	SSS	111	Jump to address in Register SSS. (Move address to Register 7).	S, Z
ADDR	SSS, DDD	6	0011	SSS	DDD	ADD contents of Register SSS to contents of register DDD. Results to DDD	S, Z, C, OV
SUBR	SSS, DDD	6	0100	SSS	DDD	SUBtract of Register SSS from contents of register DDD. Results to DDD	S, Z, C, OV
CMPR	SSS, DDD	6	0101	SSS	DDD	CoMPare Register SSS with register DDD by subtraction. Results not stored.	S, Z, C, OV
ANDR	SSS, DDD	6	0110	SSS	DDD	logical AND contents of Register SSS with contents of register DDD.Results to DDD	S, Z
XORR	SSS, DDD	6	0111	SSS	DDD	eXclusive OR contents of Register SSS with contents of register DDD.Results to DDD	S, Z
CLRR	DDD	6	0111	DDD	DDD	CLeaR Register to zero.	S, Z
INCR	DDD	6	0000	001	DDD	INCrement contents of Register DDD. Results to DDD	S, Z
DECR	DDD	6	0000	010	DDD	DECrement contents of Register DDD. Results to DDD	S, Z
COMR	DDD	6	0000	011	DDD	one's COMplement contents of Register DDD. Results to DDD	S, Z
NEGR	DDD	6	0000	100	DDD	Two's complement contents of Register DDD. Results to DDD	S, Z, C, OV
ADCR	DDD	6	0000	101	DDD	ADd Carry bit to contents of Register DDD. Results to DDD	S, Z, C, OV

REGISTER SHIFT

Executable only with Register 0, 1, 2, 3.
Shift Right instructions set the S flip-flop with Bit 7 of the result after the instruction.
Add 2 cycles if shift is 2 bits or two bytes.
Shifts are not interruptable.

SWAP	RR<,n>	6	0001	000	NRR	N = 0, SWAP bytes of register RR. S equals Bit 7 of results of SWAP.	S, Z
		8				N = 1, SWAP bytes of register RR, then swap them back to original form.	S, Z
SLL	RR<,n>	6	0001	001	NRR	N = 0. Shift Logical Left one bit, zero to low bit.	S, Z
		8				N = 1, Shift Logical Left two bits, zero to low 2 bits.	S, Z
RLC	RR<,n>	6	0001	010	NRR	N = 0, Rotate Left one bit using Carry bit as bit 16.	S, Z, C
		8				N = 1, Rotate Left two bits using C as bit 17 and OV as bit 16.	S, Z, C, OV
SLLC	RR<,n>	6	0001	011	NRR	N = 0, Shift Logical Left one bit using C as bit 16, zero to low bit.	S, Z, C
		8				N = 1, Shift Logical Left two bits using C as bit 17, OV as bit 16, zero to low 2 bits.	S, Z, C, OV
SLR	RR<,n>	6	0001	100	NRR	N = 0, Shift Logical Right one bit, zero to high bit.	S, Z
		8				N = 1, Shift Logical Right two bits, zero to high two bits.	S, Z
SAR	RR<,n>	6	0001	101	NRR	N = 0, Shift Arithmetic Right one bit, sign bit copied to high bit.	S, Z
		8				N = 1, Shift Arithmetic Right two bits, sign bit copied to high bits.	S, Z
RRC	RR<,n>	6	0001	110	NRR	N = 0, Rotate Right one bit using Carry as bit 16.	S, Z, C
		8				N = 1, Rotate Right two bits using C as bit 16, OV as bit 17.	S, Z, C, OV
SARC	RR<,n>	6	0001	111	NRR	N = 0, Shift Arithmetic Right one bit, thru Carry, sign bit copied to high bit.	S, Z, C
		8				N = 1, Shift Arithmetic Right two bits, thru Carry and OV, sign bit copied to high 2 bits	S, Z, C, OV

BRANCHES The Branch instructions are Program Counter Relative, i.e., the Effective Address = PC+Displacement. PPPPPPPPPP is the Displacement and S is 0 for +, 1 for −. If Memory is greater than 10 bits then the appropriate number of lead bits pppppp will be a part of the Displacement. For a forward branch an addition is performed; for a backward branch a ones complement subtraction is performed. Computation performed on PC+2.

MNEMONIC	OPERAND	CYCLES	INSTRUCTION	DESCRIPTION	STATUS CHANGE
B	DA	7/9	1000 S0 0000 pppppp PPPP PP PPPP	Branch unconditional, Program Counter Relative (+1025 to -1024)	
NOPP		7	1000 S0 1000 pppppp PPPP PP PPPP	No OPeration, two words.	
BC	DA	7/9	1000 S0 0001	Branch on Carry. C = 1	
BLGT	DA		pppppp PPPP PP PPPP	Branch if Logical Greater Than. C = 1	
BNC	DA	7/9	1000 S0 1001	Branch on No Carry. C = 0	
BLLT	DA		pppppp PPPP PP PPPP	Branch if Logical Less Than. C = 0	
BOV	DA	7/9	1000 S0 0010 pppppp PPPP PP PPPP	Branch on OVerflow. OV = 1	
BNOV	DA	7/9	1000 S0 1010 pppppp PPPP PP PPPP	Branch on No OVerflow. OV = 0	
BPL	DA	7/9	1000 S0 0011 pppppp PPPP PP PPPP	Branch on PLus. S = 0	
BMI	DA	7/9	1000 S0 1011 pppppp PPPP PP PPPP	Branch on MInus. S = 1	
BZE	DA	7/9	1000 S0 0100	Branch on ZEro. Z = 1	
BEQ	DA		pppppp PPPP PP PPPP	Branch if EQual. Z = 1	
BNZE	DA	7/9	1000 S0 1100	Branch on No ZEro. Z = 0	
BNEQ	DA		pppppp PPPP PP PPPP	Branch if Not EQual. Z = 0	
BLT	DA	7/9	1000 S0 0101 pppppp PPPP PP PPPP	Branch if Less Than. S $\veebar$ OV = 1	
BGE	DA	7/9	1000 S0 1101 pppppp PPPP PP PPPP	Branch if Greater than or Equal. S $\veebar$ OV = 0	
BLE	DA	7/9	1000 S0 0110 pppppp PPPP PP PPPP	Branch if Less than or Equal. Z V (S $\veebar$ OV) = 1	
BGT	DA	7/9	1000 S0 1110 pppppp PPPP PP PPPP	Branch if Greater Than. Z V (S $\veebar$ OV) = 0	
BUSC	DA	7/9	1000 S0 0111 pppppp PPPP PP PPPP	Branch if Unequal Sign and Carry C $\veebar$ S = 1	
BESC	DA	7/9	1000 S0 1111 pppppp PPPP PP PPPP	Branch if Equal Sign and Carry C $\veebar$ S = 0	
BEXT	DA,E	7/9	1000 S1 EEEE pppppp PPPP PP PPPP	Branch if EXternal condition is True. Field E is externally decoded to select 1 of 16 conditions. Response is tested for true condition.	

DIRECT ADDRESSED DATA - MEMORY

Field aaa aaa is dependent on the width of memory. 16 bits is maximum for aaa aaa AAAAAAAAAA.

MNEMONIC	OPERAND	CYCLES	INSTRUCTION	DESCRIPTION	STATUS CHANGE
MVO	SSS, A	11	1001 000 SSS aaa aaa AAAA AAA AAA	MoV Out data from register SSS to address A - A.	
MVI	A, DDD	10	1010 000 DDD aaa aaa AAAA AAA AAA	MoVe In data from address A - A to register DDD.	
ADD	A, DDD	10	1011 000 DDD aaa aaa AAAA AAA AAA	ADD data from address A - A to register DDD. Results to DDD.	S, Z, C, OV
SUB	A, DDD	10	1100 000 DDD aaa aaa AAAA AAA AAA	SUBtract data from address A - A from register DDD. Results to DDD.	S, Z, C, OV
CMP	A, SSS	10	1101 000 SSS aaa aaa AAAA AAA AAA	CoMPare data from address A - A with register SSS by subtraction. Results not stored.	S, Z, C, OV
AND	A, DDD	10	1110 000 DDD aaa aaa AAAA AAA AAA	logical AND data from address A - A with register DDD. Results to DDD.	S, Z
XOR	A, DDD	10	1111 000 DDD aaa aaa AAAA AAA AAA	eXclusive OR data from address A - A with register DDD. Results to DDD.	S, Z

INDIRECT ADDRESSED DATA - REGISTER

MMM Source data is located at the address contained in Register.
MMM = 4, 5 post increment R4 or R5.
MMM = 6 - MVO instruction - post increment R6. PUSH data from Register SSS to the Stack.
Other instructions - pre-decrement R6. PULL data from the Stack to be used as the first operand.

MVO@	SSS, MMM	9	1001	MMM	SSS	MoVe Out data from register SSS to the address in register MMM Note: SSS = MMM = 4, 5, 6 or 7 not supported.	
PSHR	SSS	9	1001	110	SSS	PuSH data from Register SSS to the stack.	
MVI@	MMM, DDD	8 *	1010	MMM	DDD	MoVe In data to register DDD from address in register MMM.	
PULR	DDD	11	1010	110	DDD	PULl data from the stack to Register DDD.	
ADD@	MMM, DDD	8 *	1011	MMM	DDD	ADD data located at address in register MMM to the contents of register DDD. Results to DDD.	S, Z, C, OV
SUB@	MMM, DDD	8 *	1100	MMM	DDD	SUBtract data located at address in Register MMM from contents of register DDD. Results to DDD.	S. Z. C, OV
CMP@	MMM, DDD	8 *	1101	MMM	SSS	CoMPare data located at address in Register MMM with contents of register SSS, by subtraction. Results not stored.	S, Z, C, OV
AND@	MMM, DDD	8 *	1110	MMM	DDD	logical AND contents of register DDD with data located at address in register MMM. Results to DDD.	S. Z
XOR@	MMM, DDD	8 *	1111	MMM	DDD	eXclusive OR contents of register DDD with data located at address in register MMM. Results to DDD.	S, Z

*Add 3 to number of cycles if MMM=6.

INSTRUCTION SET (continued)

CONTROL

MNEMONIC	OPERAND	CYCLES	INSTRUCTION			DESCRIPTION	STATUS CHANGE
GSWD	DD	6	0000	110	0DD	Get Status WorD in register DD. Bits 0-3, 8-11 set to 0. Bits 4, 12 = C; 5, 13 = OV; 6, 14 = Z; 7, 15 = S.	
NOP	<n>	6	0000	110	10N	No operation.	
SIN	<n>	6	0000	110	11N	Software Interrupt; pulse to PCIT * pin	
RSWD	SSS	6	0000	111	SSS	Restore Status Word from register SSS; Bit 4 to C, Bit 5 to OV, Bit 6 to Z, Bit 7 to S.	S, Z, C, OV
HLT		4	0000	000	000	HaLT after next instruction is executed. Resume on control start.	
EIS		4	0000	000	010	Enable Interrupt System. Not Interruptable.	
DIS		4	0000	000	011	Disable Interrupt System. Not Interruptable.	
TCI		4	0000	000	101	Terminate Current Interrupt. Not Interruptable.	
CLRC		4	0000	000	110	CLeaR Carry to zero. Not Interruptable.	C
SETC		4	0000	000	111	SET Carry to one. Not Interruptable.	C

JUMP

J	DA	12	0000 11AA AAAA	000 AAA AAA	100 A00 AAA	Jump to address. Program counter is set to 16 bits of A's.	
JE	DA	12	0000 11AA AAAA	000 AAA AAA	100 A01 AAA	Jump to address. Enable interrupt system. Program counter is set to 16 bits of A's.	
JD	DA	12	0000 11AA AAAA	000 AAA AAA	100 A10 AAA	Jump to address. Disable interrupt system. Program counter is set to 16 bits of A's.	
JSR	BB, DA	12	0000 BBAA AAAA	000 AAA AAA	100 A00 AAA	Jump and Save Return address (PC+3) in register designated by BB. Program counter is set to 16 bits of A's. BB≠11	
JSRE	BB, DA	12	0000 BBAA AAAA	000 AAA AAA	100 A01 AAA	Jump and Save Return and Enable interrupt system. Return (PC+3) is saved in register BB. Program counter is set to 16 bits of A's. BB≠11	
JSRD	BB, DA	12	0000 BBAA AAAA	000 AAA AAA	100 A10 AAA	Jump and Save Return and Disable interrupt system. Return (PC+3) is saved in register BB. Program counter is set to 16 bits of A's. BB≠11	

IMMEDIATE DATA - REGISTER The number of iiiiii bits depends on the memory width, 16 bits is maximum.

MNEMONIC	OPERAND	CYCLES	INSTRUCTION				DESCRIPTION	STATUS CHANGE
MVOI	SSS,I	9		1001	111	SSS	MoVe Out Immediate data from register SSS to PC+1	
			iiiiii	IIII	III	III	(field).	
MVII	I,DDD	8		1010	111	DDD	MoVe In Immediate data to register DDD from PC+1	
			iiiiii	IIII	III	III	(field).	
ADDI	I,DDD	8		1011	111	DDD	ADD Immediate data to contents of register DDD.	S, Z, C, OV
			iiiiii	IIII	III	III	Results to DDD.	
SUBI	I,DDD	8		1100	111	DDD	SUBtract Immediate data from contents of register	S, Z. C. OV
			iiiiii	IIII	III	III	DDD. Results to DDD.	
CMPI	I,SSS	8		1101	111	SSS	CoMPare Immediate data from contents of register	S, Z, C, OV
			iiiiii	IIII	III	III	SSS by subtraction. Results not stored.	
ANDI	I,DDD	8		1110	111	DDD	logical AND Immediate data with contents of	S, Z
			iiiiii	IIII	III	III	register DDD. Results to DDD.	
XORI	I,DDD	8		1111	111	DDD	eXclusive OR Immediate data with contents of	S, Z
			iiiiii	IIII	III	III	register DDD. Results to DDD.	

SDBD		4		0000	000	001	Set Double Byte Data for the next instruction which must be an external reference instruction. The effective address of the external reference instruction will address the low order data byte; the address of the high order data byte will be EA+1 if register 4, 5 or 7 is used. If register 1-3 is used the EA will access the same byte twice resulting in both bytes of data being the same. Use of modes 0 and 6 are not supported by this instruction.	
			This instruction is normally supplied by the assembler as required to properly generate machine code.					

INDIRECT ADDRESSED DOUBLE BYTE DATA - REGISTER

SDBD		4		0000	000	001	MoVe In double byte data from the address in register MMM to	
MVI@	MMM, DDD	10		1010	MMM	DDD	register DDD.	
SDBD		4		000	000	001	ADD double byte data from the address in register MMM to the	S, Z, C, OV
ADD@	MMM, DDD	10		1011	MMM	DDD	content of register DDD. Results to DDD.	
SDBD		4		0000	000	001	SUBtract double byte data located at address MMM from the	S, Z, C, OV
SUB@	MMM, DDD	10		1100	MMM	DDD	content of register DDD. Results to DDD.	
SDBD		4		0000	000	001	CoMPare double byte data located at address in register MMM	S, Z, C, OV
CMP@	MMM, DDD	10		1101	MMM	SSS	with the content of register SSS by subtraction. Results is not stored.	
SDBD		4		0000	000	001	logical AND double byte data located at address in register MMM	S, Z
AND@	MMM, DDD	10		1110	MMM	DDD	with the content of register DDD. Results to DDD.	
SDBD		4		0000	000	001	eXclusive OR double byte data located at address in register MMM	S, Z
XOR@	MMM, DDD	10		111	MMM	DDD	with the content of register DDD. Results to DDD.	

IMMEDIATE DOUBLE BYTE DATA - REGISTER

Note: The SDBD command is provided by the assembler when the immediate data is greater than the memory width and requires two bytes.

MVII	I,DDD	14	0000	000	001	MoVe In Immediate double byte data to register DDD. L's will be low byte and U's upper byte. XX = don't care.	
			1010	111	DDD		
			XXLL	LLL	LLL		
			XXUU	UUU	UUU		
ADDI	I,DDD	14	0000	000	001	ADD Immediate double byte data to contents of register DDD. Results to DDD. L's indicate low byte of literal, U's upper byte.	S, Z, C, OV
			1011	111	DDD		
			XXLL	LLL	LLL		
			XXUU	UUU	UUU		
SUBI	I,DDD	14	0000	000	001	SUBtract Immediate double byte data from contents of register DDD. Results to DDD. L's indicate low byte of literal, U's upper byte.	S, Z, C, OV
			1100	111	DDD		
			XXLL	LLL	LLL		
			XXUU	UUU	UUU		
CMPI	I,SSS	14	0000	000	001	CoMPare Immediate double byte data with contents of register SSS by subtraction. Results not stored. L's indicate low byte of literal, U's upper byte.	S, Z, C, OV
			1101	111	SSS		
			XXLL	LLL	LLL		
			XXUU	UUU	UUU		
ANDI	I,DDD	14	0000	000	001	logical AND Immediate double byte data with the contents of Register DDD. Results to register DDD. L's indicate low byte of literal, U's upper byte.	S, Z
			1110	111	DDD		
			XXLL	LLL	LLL		
			XXUU	UUU	UUU		
XORI	I,DDD	14	0000	000	001	eXclusive OR Immediate double byte data with the contents of register DDD. Results to Register DDD. L's indicate low byte of literal, U's upper byte.	S, Z
			1111	111	DDD		
			XXLL	LLL	LLL		
			XXUU	UUU	UUU		

GLOSSARY OF TERMS

SSS - Source Register
DDD - Destination Register
n - Number of Shifts
RR - Register to Shift (only 0-3 allowed)
AAAAAA Memory address for Jump.
AAAAAAAAAA (new Program Counter)
BB - Register to save old PC in for Jump. (Reg = 1BB, 4, 5, or 6)
S - Sign of address displacement for Branch (PC relative).
pppppp PPPPPPPPPP - Address displacement for Branch
pppppp is dependent on the memory word size.
aaaaaa AAAAAAAAAA - Direct address of data word.
aaaaaa is dependent on the memory word size.
iiiiii IIIIIIIII - Immediate data word. iiiiii is dependent on memory word size.
LLLLLLLL Lower 8 bits of double byte data.
UUUUUUUU Upper 8 bits of double byte data.

MMM - Address Mode
000 - direct address in location following instruction.
001 - indirect address for Register 1
010 - indirect address for Register 2
011 - indirect address for Register 3
100 - indirect address for Register 4, post increment
101 - indirect address for Register 5, post increment
110 - indirect address for Register 6, post increment for MVO only
indirect address for Register 6, pre decrement for all instructions except MVO.
111 indirect address for Register 7, post increment.
(Immediate data in location following instruction.)

3.4 PROGRAM EXAMPLES

The following examples demonstrate common program coding techniques for use on the CP-1600. These include loop control, array addressing, table look-up, use of the stack, reentrant interrupt service, subroutine calls, and other miscellaneous ideas. All are intended to be useful examples but in many cases are designed to show the use of particular instructions as opposed to minimizing code or optimizing speed of execution.

3.4.1 LOOP OR ITERATION CONTROL

Counting down using a general register:

```
        MVII    n, R3           ; set loop count
LOOP    .
        .
        .
        DECR    R3              ; decrement count
        BNZE    LOOP            ; if not zero, continue loop
                                ; loop complete
```

Counting up using a general register:

```
        CLRR    R2              ; zero count
LOOP    .
        .
        .
        .
        INCR    R2              ; increment count
        CMPI    n, R2           ; compare to limit
        BNEQ    LOOP            ; if not equal, continue loop
                                ; loop complete
```

Counting down using a storage cell:

```
        MVII    n, R0           ; set loop count
LOOP    MVO     R0, CNTR        ; store count
        .
        .
        .
        .
        MVI     CNTR, R0        ; fetch count
        DECR    R0              ; decrement count
        BNZE    LOOP            ; if not zero, continue loop
        .                       ; loop complete
        .
        .
```

Loop control by address limiter:

```
        MVII    ADDR, R5        ; set starting address
LOOP    ADD@    R5, R0          ; fetch data, increment pointer
          .
          .
          .
        CMPI    ADDR+n, R5      ; compare for final address
        BNEQ    LOOP            ; if not equal, continue loop
```

3.4.2 TABLE ACCESS

Sequential ascending access:

```
        MVII    n, R3           ; set count
        MVII    TBL, R4         ; set table base address
LOOP    MVI@    R4, R0          ; fetch table item & increment address
          .
          .
        DECR    R3              ; decrement count
        BNZE    LOOP            ; continue until all items accessed
          .                     ; complete
          .
          .
```

Sequential descending access:

```
        MVII    n, R3           ; set count
        MVII    TBL+n, R2       ; set table end address
LOOP    MVI@    R2, R0          ; fetch table item
        DECR    R2              ; decrement table address
          .
          .
        DECR    R3              ; decrement count
        BNZE    LOOP            ; continue until all items accessed
          .
          .                     ; complete
```

Random access; ie, TBL (I):

```
          .
          .
          .
        MVI     I, R1           ; fetch index
        ADDI    TBL-1, R1       ; compute table pointer
        MVI@    R1, R0          ; fetch table item
          .
          .
          .
```

3.4.3 TRANSFER VIA AN INDEXED DISPATCH TABLE

GO TO (a, b, c, d), I

```
      MVI     I, R1          ; fetch dispatch index
      ADDI    TBL -1, R1     ; compute pointer to table
      MVI@    R1, R7         ; move dest addr to prog counter
TBL   WORD    a, b, c, d     ; destination addresses
```

3.4.4 EVALUATE VARIABLES AND TRANSFER

IF (I . LT . J) GO TO a

```
      MVI     J, R0          ; fetch J variable
      CMP     I, R0          ; compare I to J
      BLT     a              ; branch if less than
```

3.4.5 ARITHMETIC OPERATIONS

Add variables ; ie, I = J + K + L :

```
      MVI     J, R0          ; fetch J
      ADD     K, R0          ; add K
      ADD     L, R0          ; add L
      MVO     R0, I          ; move sum to I
```

Add and subtract variables; ie, I = (J + K) - (L - M) :

```
      MVI     J, R0          ; fetch J
      ADD     K, R0          ; add K
      MVI     L, R1          ; fetch L
      SUB     M, R1          ; subt. M
      SUBR    R1, R0         ; subtract L-M from J+K
      MVO     R0, I          ; move result to I
```

Double precision register to register add:

```
      ADDR    R3, R1         ; add lower words
      ADCR    R0             ; add in any carry
      ADDR    R2, R0         ; add upper words
```

Double precision memory to register subtract:

```
      DECR    R0             ; correct upper for add carry
      SUB@    R4, R1         ; subtract lower halves, increment pntr.
      ADCR    R0             ; add in any carry
      SUB@    R4, R0         ; subtract upper halves
```

3.4.6 STACK OPERATIONS

When register R6 is used to address storage indirectly, post-incrementing and pre-decrementing are performed by the CP 1600 automatically. When data is moved out of the CPU via R6, it is placed in the storage cell specified by the contents of the register and then the contents of R6 is incremented by one. When data is moved into the CPU via R6, the contents of R6 is first decremented then data is fetched from the storage cell specified by R6. Thus, R6 can be used to manage a dynamic last in, first out storage area commonly called a stack.

The stack is used by the CP 1600 as a temporary storage area to hold interrupt return addresses. It can also be used to hold nested subroutine return addresses, data during general programming activities, and operands to be evaluated using polish notation techniques. For user convenience, the Series 1600 assemblers recognize the mnemonics PSHR r and PULR r as the equivalent of MVO@ r, R6 and MVI@ R6, r respectively. PSHR r causes the contents of register r to be pushed onto the top of the stack maintained by R6 and conversely PULR r pulls the contents of the top of the stack into register r.

Save registers:

```
PSHR Rn
```

Restore registers:

```
PULR Rn
```

Add top two elements in stack and compare to third element:

```
PULR    R0          ; fetch top element
ADD@    R6, R0      ; add second element
CMP@    R6, R0      ; compare to third
BEQ     MATCH       ; if equal, go to MATCH
```

3.4.7 INTERRUPT PROCESSING

Reentrant interrupt processing; i.e., recognizing and processing higher priority interrupts while currently servicing a lower priority one, requires totally reentrant CPU context save and restore routines.

Context save:

```
CNXSAV  PSHR   R0          ; save R0
        GSWD   R0          ; fetch CPU status
        PSHR   R0          ; save status
        PSHR   R1          ; save all registers
        PSHR   R2          ; required for
                           ; interrupt service
                           ; routine up
        PSHR   Rn          ; to Rn
```

Context restore:

```
CNXRST   PULR    Rn          ; restore Rn
           .
           .
           .
         PULR    R2          ; restore R2
         PULR    R1          ; restore R1
         PULR    R0          ; get status
         RSWD    R0          ; replace status
         PULR    R0          ; restore R0
```

3.4.8 SUBROUTINES

A subroutine is a common sequence of instructions which can be executed many times by being called from many different places in a program. The JSR (jump and save return) instruction provides for entry into subroutines while saving the return address in register 4, 5, or 6. Data passed to a subroutine by the calling program (arguments) may be transferred by values or addresses held in registers or as items stored sequentially following the JSR instruction.

Subroutine call with no arguments:

```
         JSR     R5, SUBR        ; transfer to subroutine
           .
           .
           .
SUBR       .
           .
           .
        MOVR     R5, R7          ; return via addr in R5
```

Note: The Series 1600 assemblers recognize JR r as the equivalent of MOVR r, R7

Subroutine call with argument passed via a general register:

```
         MVI     J, R0           ; fetch argument
         JSR     R4, SUBR        ; transfer to subroutine
           .
           .
SUBR       .
           .
           .
           .
         JR      R4              ; return via R4
```

Subroutine call with argument value following the JSR instruction:

```
         JSR     R5, SUBR        ; transfer to SUBR
         WORD     6              ; arg 1 value
         WORD     2              ; arg 2 value
           .
           .
           .
```

```
SUBR    MVI@    R5, R0          ; fetch arg 1 & update return addr
        MVI@    R5, R1          ; fetch arg. 2 & update return addr
          .
          .
          .
        JR      R5              ; return
```

Subroutine call with argument address following the JSR instruction; ie, call SUBR (I):

```
        JSR     R5, SUBR        ; transfer to SUBR
        WORD    I               ; address of variable
          .
          .
          .
SUBR    MVI@    R5, R1          ; fetch arg addr, update return
        MVI@    R1, R1          ; fetch value
          .
          .
          .
        JR      R5              ; return
```

Nested subroutines:

```
        JSR     R5, SUB 1       ; transfer to SUB 1

SUB 1   PSHR    R5              ; save return addr on stack
          .
          .
          .
        JSR     R5, SUB 2       ; transfer to next subroutine level
          .
          .
          .
        PULR    PC              ; exit subroutine SUB 1

SUB 2   PSHR    R5              ; save return addr on stack
          .
          .
          .
        JSR     R5, SUB 3       ; transfer to next subroutine level
          .
          .
        PULR    PC              ; exit subroutine SUB 2

SUB 3     .
          .
          .
          .
```

3.4.9 BEXT INSTRUCTION (POLLING)

The BEXT instruction allows up to 16 external conditions to be easily tested by software. These conditions can be switches on a control panel, operator command inputs, data ready signals from peripherals, or any other type of digital status signal. This instruction allows polling schemes to be easily implemented.

Poll inputs and vector to service routines:

```
LOOP    BEXT    SRV1, 1          ; test condition 1
        BEXT    SRV2, 2          ; test condition 2
        BEXT    SRV3, 3          ; test condition 3
          .
          .
          .
        B       LOOP             ; branch back to start again
```

Wait loop for I/O device:

```
LOOP    BEXT    LOOP, 5          ; test data ready & wait
        MVI     IODATA, R0       ; get data when ready
          .
          .
          .
          .
```

```
                                REL   BINOCT
                        ;;;;;;
                        ;
                        ; 1: PROGRAM NAME: S16LIB,BINOCT
                        ;
                        ; 2: ORIGIN DATE: NOV. 20, 1974
                        ;
                        ; 3: LAST REVISION DATE:
                        ;
                        ; 4: AUTHOR:  ROBERT B. MUIR
                        ;
                        ; 5: REVISION LEVEL: 1A
                        ;
                        ; 6: PURPOSE: TO CONVERT BINARY DATA IN REGISTER
                        ;            TO ASCII FORM OCTAL NUMBER
                        ;            AND STORE IN BUFFER
                        ;
                        ; 7: METHOD USED:
                        ;
                        ; 8: COMMENTS:
                        ;
                        ; 9: CALLING SEQUENCE:
                        ;            R0 = DATA TO BE CONVERTED
                        ;            R4 = BUFFER BASE ADDRESS
                        ;            JSR R5,BINOCT
                        ;
                        ; 10: RETURN: ASCII FORM OCTAL NUMBER IN BUFFER
                        ;
                        ; 11: STACK LEVEL: 3
                        ;
                        ; 12: SIZE: 20 WORDS
                        ;
                        ; 13: EXECUTION TIME:
                        ;            MAXIMUM:
                        ;            MINIMUM:
                        ;            TYPICAL: 187.2 USEC
                        ;
                        ;;;;;;
                        R0      EQU   0
                        R1      EQU   1
                        R2      EQU   2
                        R3      EQU   3
                        R4      EQU   4
                        R5      EQU   5
                        SP      EQU   6
                        PC      EQU   7

                                GLOB  BINOCT

000000 001163   BINOCT  PSHR  R3          ;SAVE REGS
000001 001162          PSHR  R2
000002 001161          PSHR  R1
000003 001271          MVII  6,R1         ;6 CHR COUNTER:
000004 000006
000005 000722          CLRR  R2            ;OPERATING REG:
000006 000130   LOOP   SLLC  R0,1
000007 000122          RLC   R2,1
000010 001372          ADDI  060,R2        ;ASCII MASK
```

```
     000011 000060
58   000012 001142          MV@@  R2,R4     ;CHR -> BUFFER
59   000013 000722          CLRR  R2
60   000014 000134          SLLC  R0,2
61   000015 000126          RLC   R2,2
62   000016 000021          DECR  R1        ;6 CHR YET?
63   000017 001054          BNZE  L@@P      ;N@
     000020 000012
64   000021 001261          PULR  R1        ;YES, RESTORE REGS,
65   000022 001262          PULR  R2
66   000023 001263          PULR  R3
67   000024 000257          JR    R5
68   000024                 END
```

```
                                REL   OCTBIN
                        ;;;;;;
                        ;
                        ; 1. PROGRAM NAME: S16LIB,OCTBIN
                        ;
                        ; 2. ORIGIN DATE: NOV. 20, 1974
                        ;
                        ; 3. LAST REVISION DATE:
                        ;
                        ; 4. AUTHOR:  ROBERT B. MUIR
                        ;
                        ; 5. REVISION LEVEL: 1A
                        ;
                        ; 6. PURPOSE: TO CONVERT ASCII FORM OCTAL NUMBER IN
                        ;            BUFFER TO BINARY.
                        ;
                        ; 7. METHOD USED:
                        ;
                        ; 8. COMMENTS: CONVERSION TERMINATES ON NON
                        ;             NUMERIC (0-7) OR WHEN SIX CHARACTERS
                        ;             HAVE BEEN CONVERTED. LEADING SPACES
                        ;             IGNORED. BINARY VALUE IS RETURNED
                        ;             IN R0, STRING POINTER IS RETURNED
                        ;             IN R4.
                        ;
                        ; 9. CALLING SEQUENCE:
                        ;             R4 = BUFFER POINTER
                        ;             JSR R5,OCTBIN
                        ;
                        ; 10. RETURN:
                        ;             R0 = BINARY ACCUMULATION
                        ;
                        ; 11. STACK LEVEL: 2
                        ;
                        ; 12. SIZE: 33 WORDS
                        ;
                        ; 13. EXECUTION TIME:
                        ;             MAXIMUM:
                        ;             MINIMUM: 50.0 USEC
                        ;             TYPICAL: 245.6 USEC
                        ;
                        ;;;;;;
                        R0      EQU   0
                        R1      EQU   1
                        R2      EQU   2
                        R3      EQU   3
                        R4      EQU   4
                        R5      EQU   5
                        SP      EQU   6
                        PC      EQU   7

                                GLOB  OCTBIN

000000 001163   OCTBIN  PSHR  R3
000001 001162           PSHR  R2
000002 001271           MVII  6,R1      ; 6 CHR
000003 000006
000004 000700           CLRR  R0
```

```
58  000005 000733          CLRR  R3          ;FLG = 0
59  000006 001242  GET     MVI@  R4,R2       ;GET BUFFER CHR
60  000007 000233          MOVR  R3,R3
61  000010 001014          BNZE  CHK
    000011 000005
62  000012 001572          CMPI  ' ',R2      ;SPACE?
    000013 000040
63  000014 001044          BEQ   GET         ;YES, IGNORE IT
    000015 000007
64  000016 000013          INCR  R3
65  000017 001472  CHK     SUBI  060,R2
    000020 000060
66  000021 001013          BMI   ADJPT
    000022 000013
67  000023 001572          CMPI  '7',R2
    000024 000067
68  000025 001016          BGT   ADJPT
    000026 000007
69  000027 000114          SLL   R0,2        ;POSITION BINARY ACCUM
70  000030 000110          SLL   R0,1        ;FOR NEXT DIGIT
71  000031 000720          XORR  R2,R0       ;INSERT IN ACCUM
72  000032 000021          DECR  R1
73  000033 001054          BNZE  GET
    000034 000026
74  000035 000017          INCR  PC
75  000036 000024  ADJPT   DECR  R4          ;BACK PTR TO RIGHT PLACE.
76  000037 001262  EXIT    PULR  R2
77  000040 001263          PULR  R3
78  000041 000257          JR    R5          ;EXIT
79  000041                 END
```

CHAPTER 4

CP 1600 SYSTEM CONFIGURATION

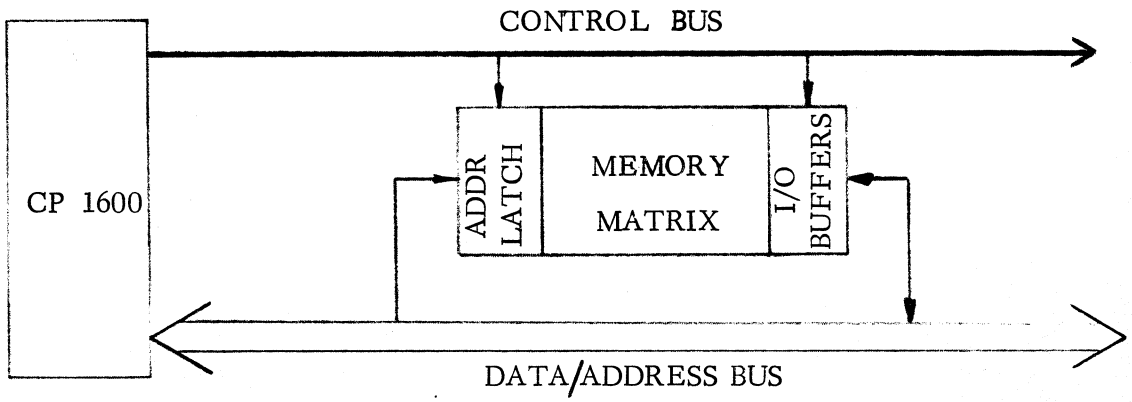
4.0 BASIC IMPLEMENTATION

The CP 1600 is a single chip, 16-bit MOS-LSI microprocessor array with many advanced features. In its most basic form as a semiconductor, it serves as the heart of a high performance microcomputer-based system. As such, it must be configured to meet both the processing and economic requirements of its end product environment. The following sections discuss a number of ways to configure the CP-1600 microprocessor into a working system.

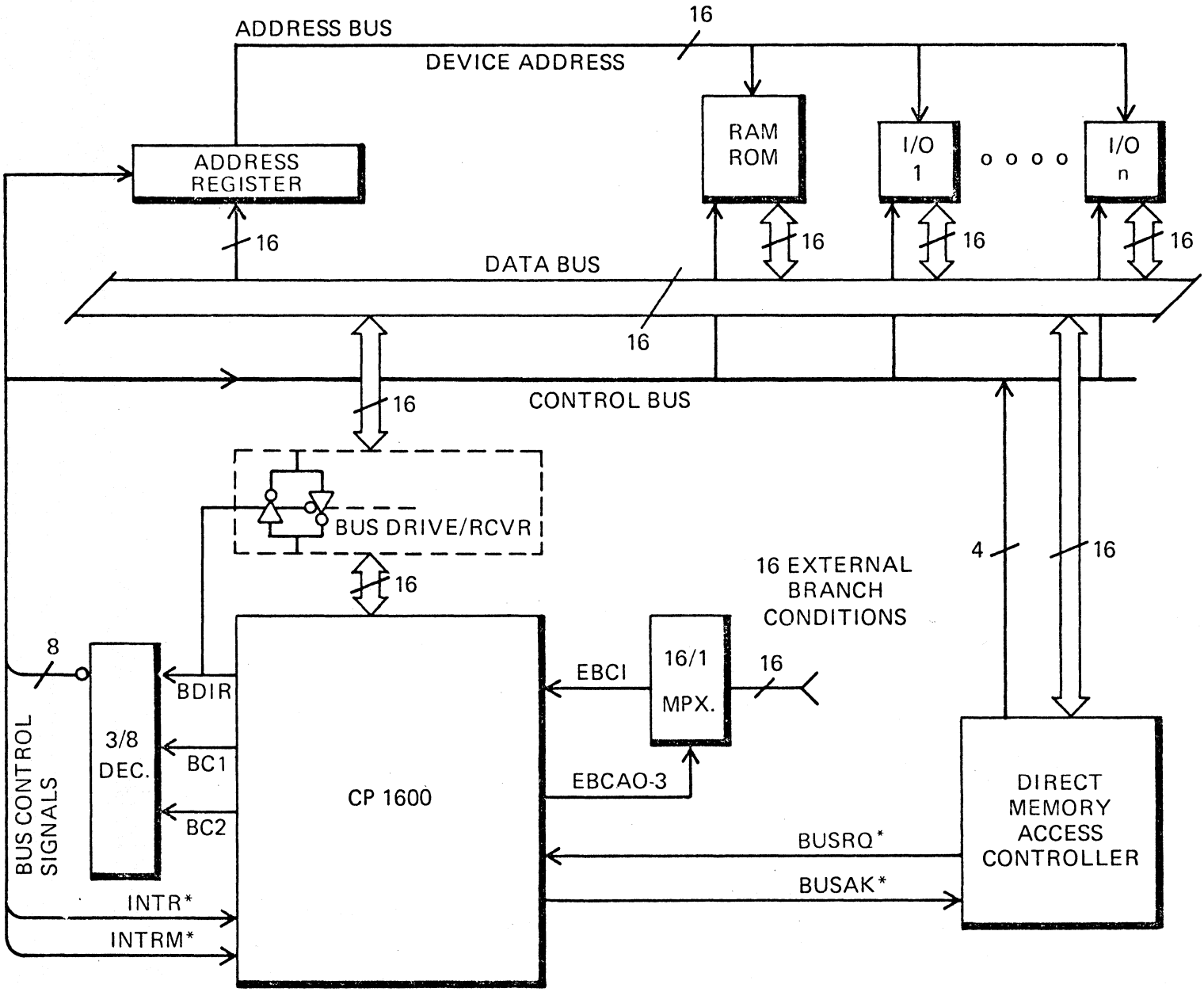
4.1 BUS STRUCTURE

The bus structure of the CP 1600 microprocessor array is based on a single, 16-bit, bidirectional bus used to output both data and addresses and input instructions, data, and interrupt addresses. However, in a microcomputer based product, there are a number of different approaches to implementing the total system bus structure around the single bidirectional bus of the CP 1600.

In small systems with only a few memory chips, limited peripheral interfacing, and minimal expansion capability, the CP 1600 bus can be used directly without buffering. Typically, it can drive 1 TTL and 200 pf at full rated speed. With A.C. loading only, it can typically drive 500 pf at 80% of full rated speed. To accommodate the unbuffered, single bus approach, many of the General Instrument ROM and RAM products contain internal address and chip select registers to trap and hold memory addresses from the CP 1600 bus. This reduces parts count significantly in these small systems. In addition, a convenient and cost-saving technique in these small systems is to utilize the upper bits of the 16-bit address as direct chip select signals. This creates a non-contiguous memory allocation but eliminates the need for full binary decoding of the upper part of the address field. Because the CP 1600 provides a full 16-bit address, there is plenty of Address Space available to accommodate this technique in small systems.

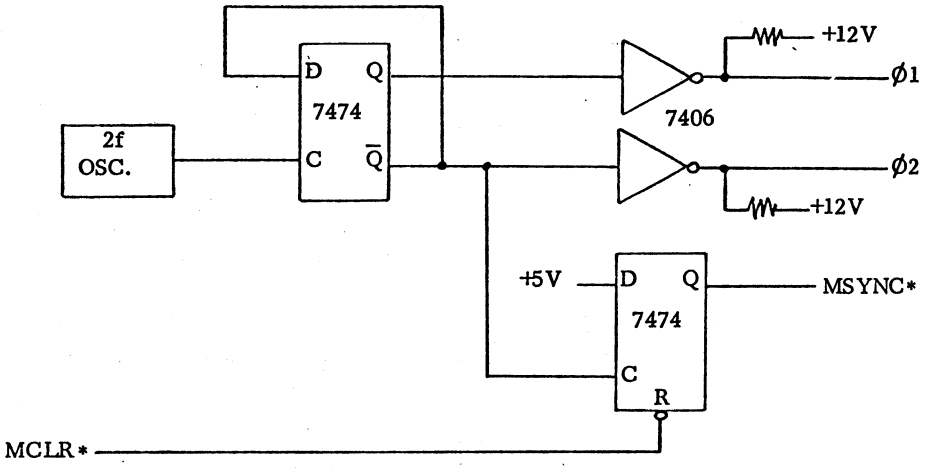


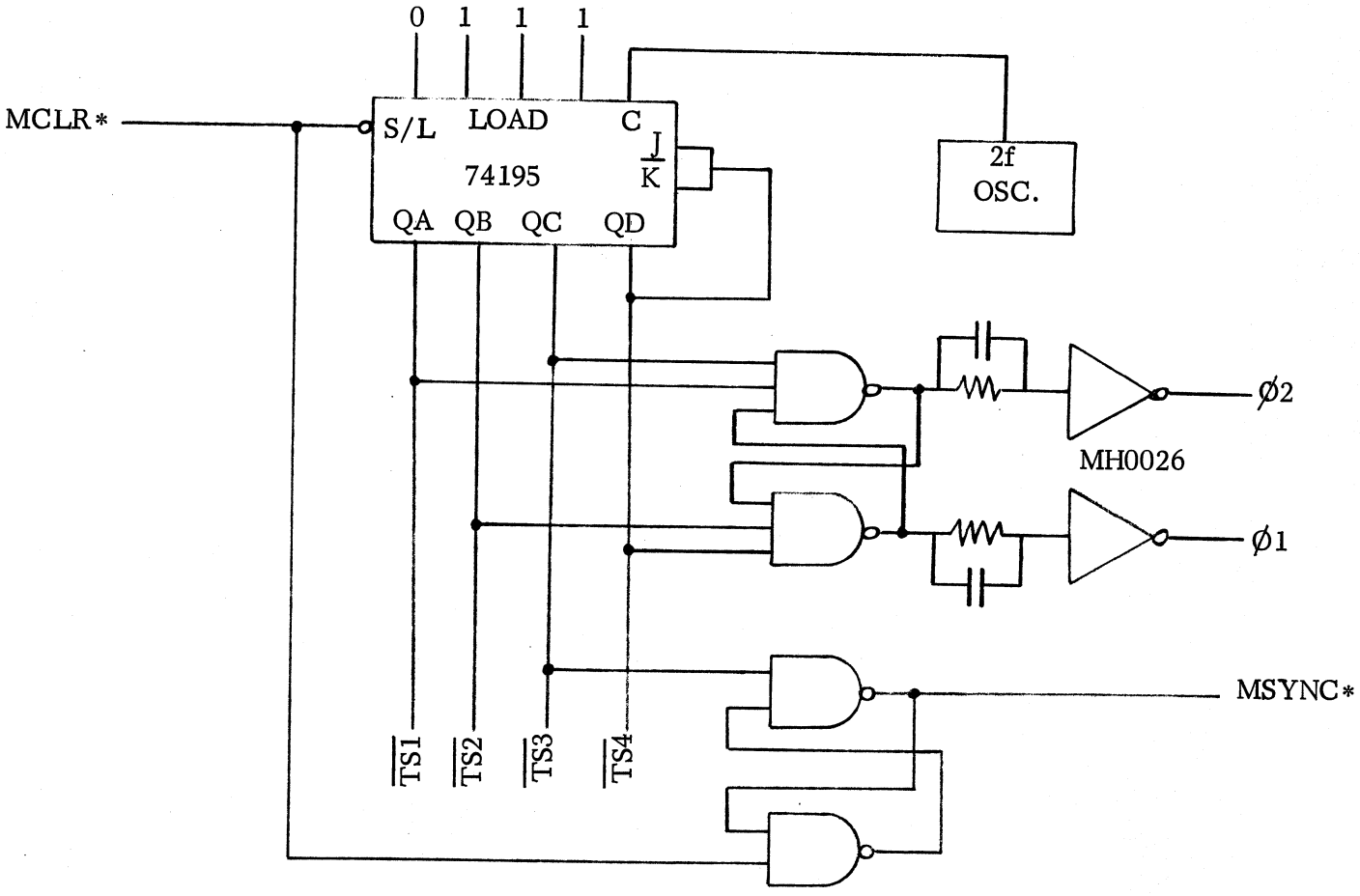
In system with larger memories, more sophisticated peripheral interfacing, and extensive modularity and expandability, the CP-1600 is best utilized with bipolar buffering of the bidirectional bus. Many standard, multiple-sourced, TTL parts are presently available for this purpose and provide the system designer with a number of important configuration options.



4.2 CLOCKS

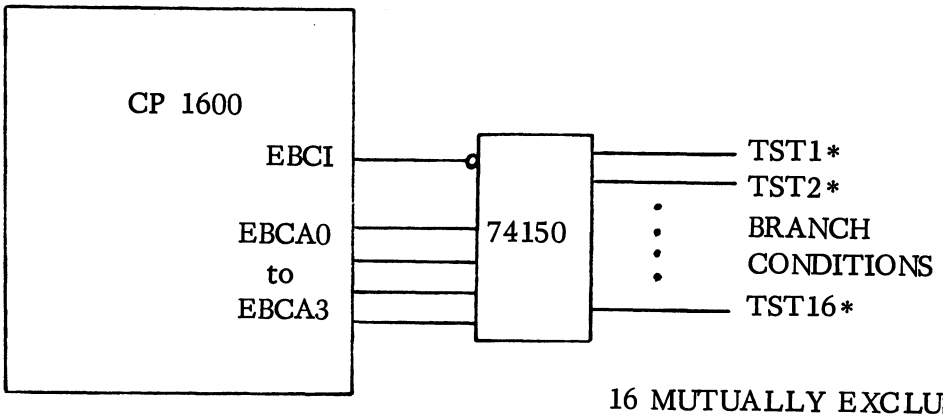
The CP 1600 requires a two phase, non-overlapping, high level clock system for proper operation. Internally, the Ø1/Ø2 clocks are used to generate the four internal time slots TS1, TS2, TS3, TS4, which make up each microcycle. In addition, the power-on/master clear sequence requires that MSYNC* go high at the rising edge of a Ø1 clock phase (TS3). The following examples are typical clock generation circuits.



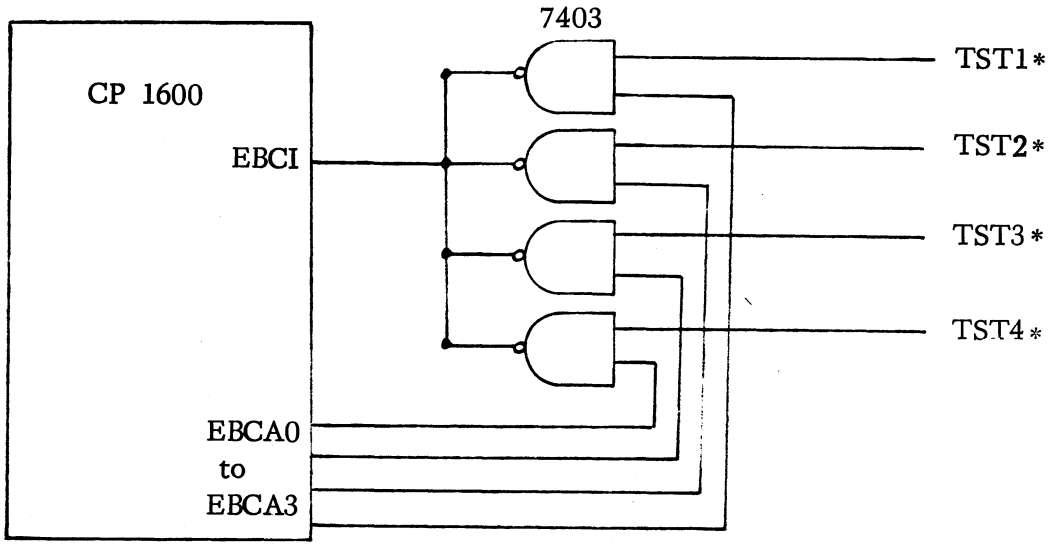


4.3 EXTERNAL SENSE LOGIC

The Branch on EXTernal condition (BEXT) instruction of the CP 1600 allows up to 16 external digital signals to be sampled by the program with a program branch occurring if the test is true. The four lower bits of the Instruction Register (IR) are presented at pins EBCA0-3 and the CPU samples EBCI to determine if the branch should occur. Two simple examples of the implementation of this function are shown below.



16 MUTUALLY EXCLUSIVE
BRANCH TESTS



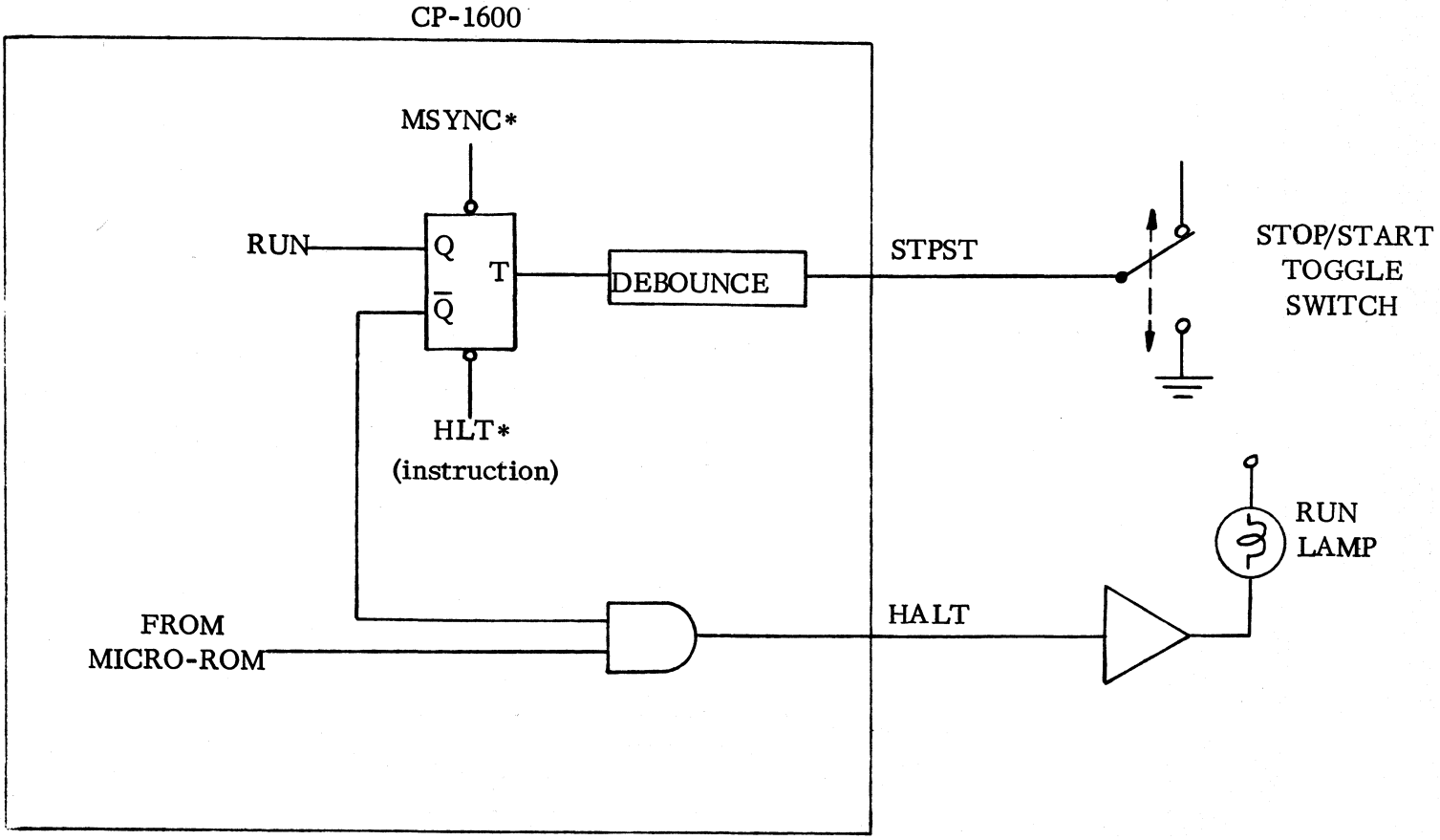
4 TEST CONDITIONS WITH
MULTIPLE "AND" CAPABILITY

4.4 START/STOP AND HALT

Most systems incorporating a microprocessor require some sort of start/stop mechanism and a simple indication of the present operational mode; i.e., running or stopped. The CP-1600 has an internal START/STOP FLIP-FLOP to provide these functions totally on-chip.

The STPST pin is a negative edge trigger to the toggle input of this flip-flop so that alternate pulses cause transistions between the RUN and STOP modes of the CPU. This input also has an internal debounce circuit so that a momentary switch can be directly connected to the CPU to perform the START/STOP function.

The HALT output from the CPU is simply the indication of the RUN/STOP condition of the processor. Because the CPU stops only after the completion of an instruction, the HALT signal is generated from a logical combination of the START/STOP FLIP-FLOP and the microcontrol sequencing logic. It represents the true state of the operational mode of the CP-1600.



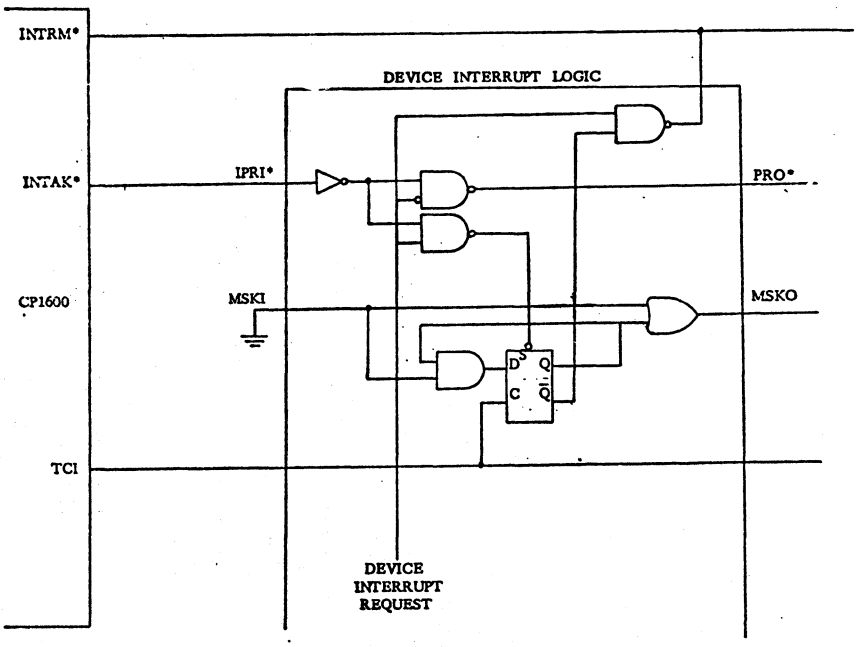
4.5 INTERRUPT SYSTEM

The interrupt system of the CP 1600 microprocessor is based on a generalized multi-line/multi-level configuration which can be used to implement interrupt structures ranging in complexity from simple "polled" priority networks to sophisticated, fully reentrant, nested priority structures. In addition, interrupts can be identified in groups or individually as the system performance demands.

The CP 1600 provides five signals to control the interrupt system. These signals include two request lines, INTR* and INTRM*, two bus control signals, INTAK and IAB, and an interrupt termination signal, TCI. In the simplest case, each device can request interrupt service on the INTRM* line with a fixed interrupt address vector being generated by the IAB signal. The software service program can then disable the INTRM* line and "poll" via BEXT instructions to determine the highest priority device requiring service. At the completion of the appropriate service, the INTRM* line can be reenabled by the EIS command.

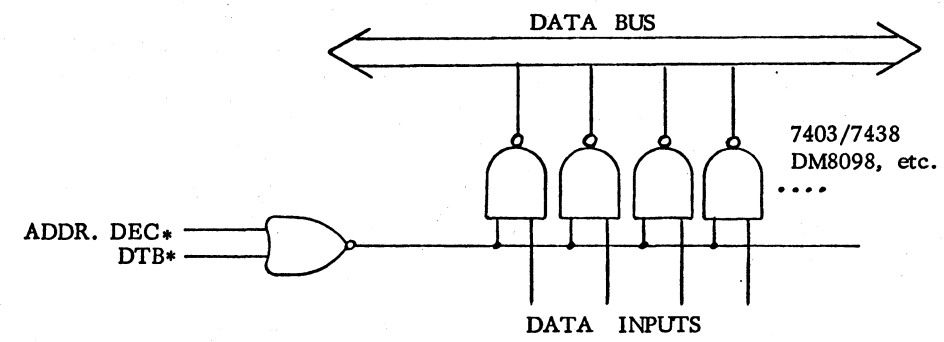
In a more sophisticated interrupt system, the INTAK acknowledge signal can be used to find the highest priority interrupting device via a daisy-chain network. Acknowledgement of the highest priority device masks out all lower priority devices allowing only higher priority devices to generate interrupts and break into device service already in progress. These nested interrupts can be terminated in reverse order by the TCI pulse with the use of a simple daisy-chain network between the ACK FF in each device.

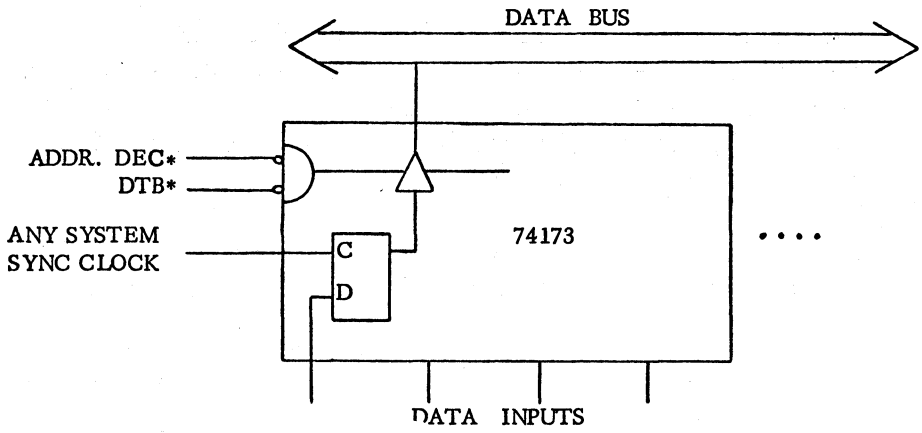
All of the interrupt system signals are discussed fully in Section 2.1 and a simple example of a nested interrupt structure is shown below.

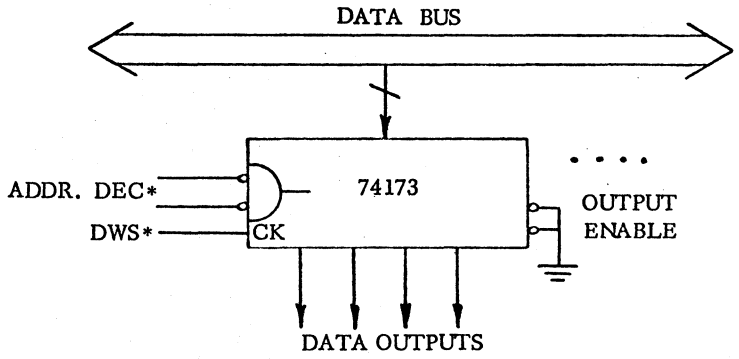


4.6 BASIC INPUT/OUTPUT PORTS

The most fundamental input/output structure of any digital processor is the simple program-controlled I/O PORT. These interfaces are composed of output holding registers and input transfer gates with all peripheral device control performed by software programming.

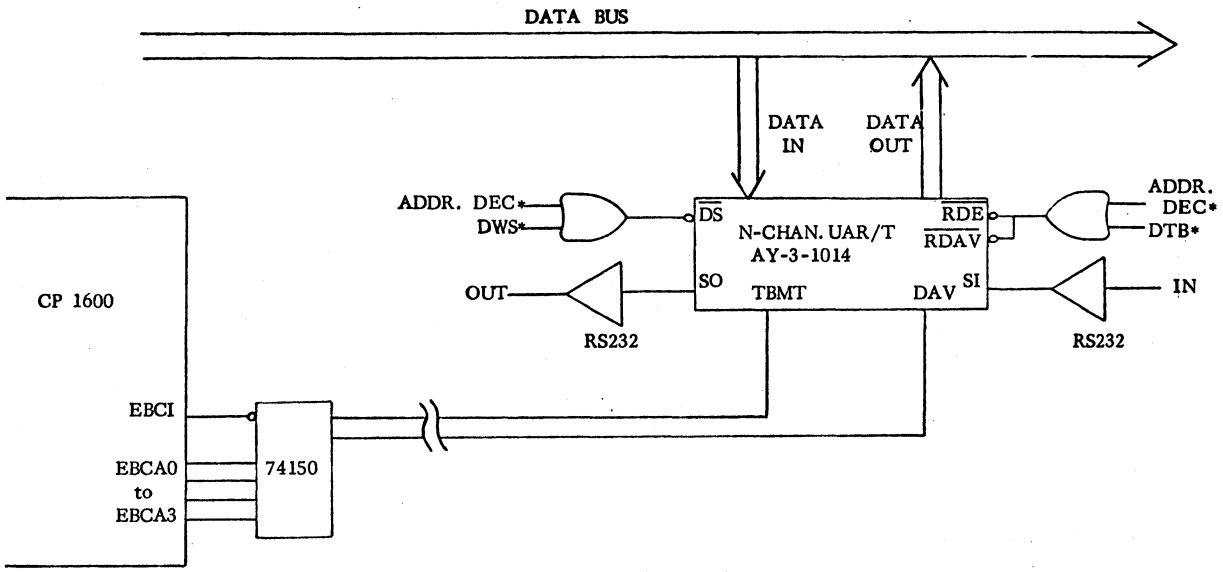






4.7 SIMPLE COMMUNICATIONS INTERFACE

The following example shows a low cost, byte buffered communications interface utilizing a Universal Asynchronous Receiver/Transmitter (UAR/T). This interface allows data rates of 60K Band to be easily handled with very few components. Because the UAR/T is internally double buffered in both the receiver and transmitter sections, the CP 1600 program has a full 170 μsec to process characters.



APPENDIX A - SERIES 1600 MICROPROCESSOR PRODUCTS

The Series 1600 Microprocessor System is a new approach to microprocessor products from General Instrument Corporation. It is the first microprocessor system to be developed around the same total product family concept that has proven so effective in the computer industry. This approach guarantees a wide range of hardware and software products and services that will enjoy long product life while maintaining the latest in semiconductor technology.

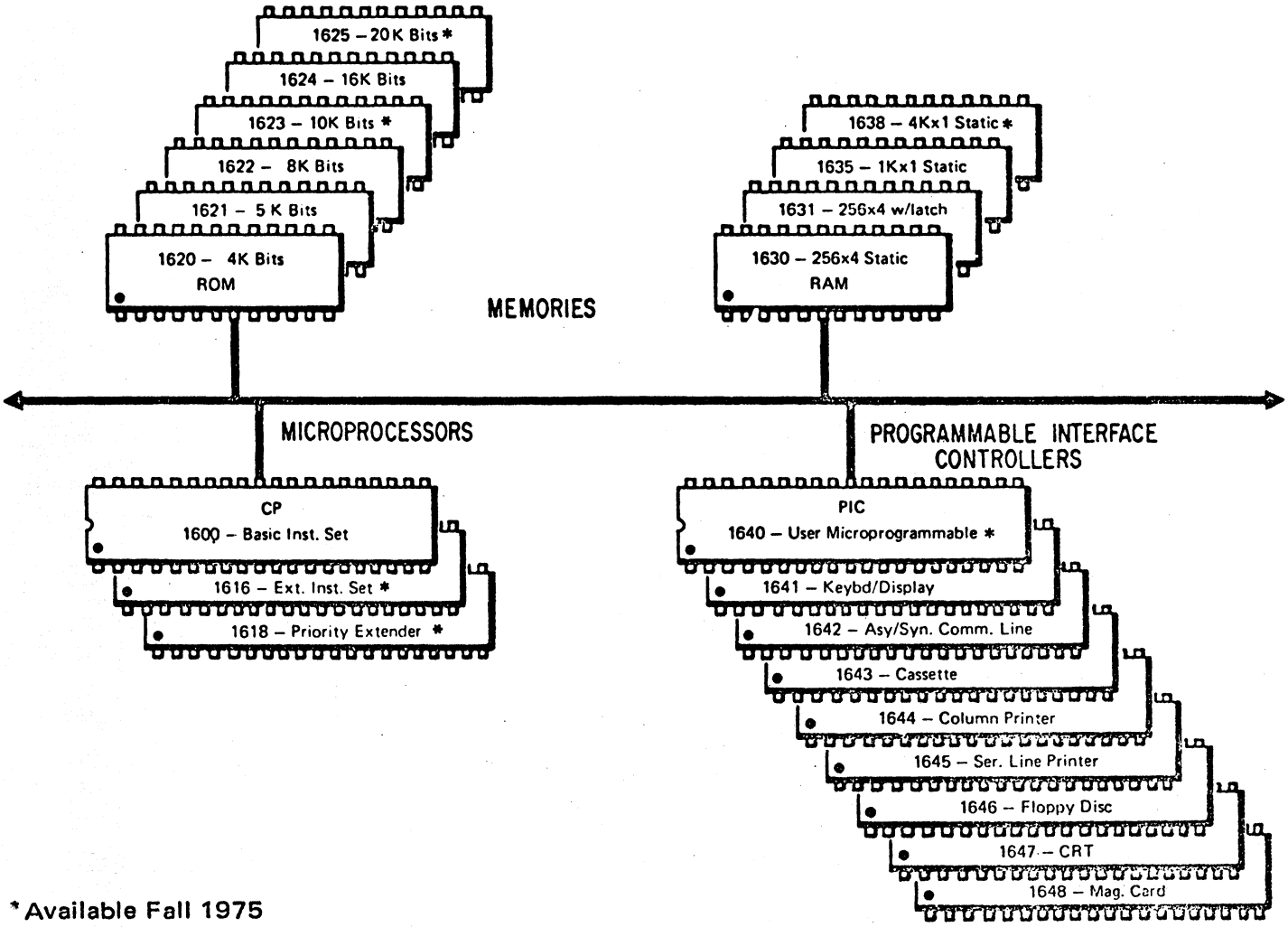
The Series 1600 is based on an advanced 16-bit architecture that provides high performance in conjunction with an easy to use, rich instruction set with built-in growth potential. Supporting this architecture is a growing family of high speed N-Channel MOS-LSI arrays that includes a family of upward compatible processors and a series of standard ROM and RAM memory circuits. In addition, the Series 1600 incorporates a powerful, intelligent I/O interface concept with its complement of Programmable Interface Controllers (PIC) consisting of a generalized user-programmable device and a host of preprogrammed versions aimed at interfacing many of the most popular industry peripherals.

This Series 1600 NMOS-LSI hardware forms the foundation for an extensive software support system that will grow with the product family. Sophisticated program preparation tools include compatible Assembler/Simulator software at three levels - a large machine or time-share, popular minicomputer systems, and on the microprocessor itself. Extensive subroutine libraries, diagnostics, utility programs, and a very complete and easy to use On-Line Debug Program (ODP) to aid in program checkout directly on the microprocessor system are also available. Another unique feature of the Series 1600 software is the Language Generation Program (LGP) which enables a high level language to be developed to match each application environment.

To simplify hardware and software development and speed the users product design cycle, a complete hardware prototype development system is available to support the Series 1600 family. The GIC1600 Microcomputer System provides a test bed for user designed interfaces and related hardware as well as a full program preparation facility with resident, on-line hardware and software debug aids. To make program development fast and efficient, peripheral interfaces and their related software routines include TTY, high speed paper tape equipment, serial line printer, magnetic tape cassette, and floppy disc. In addition, all of the card level modules of this system, ranging from complete microcomputers to memory and I/O modules, are available on an OEM basis for further system integration.

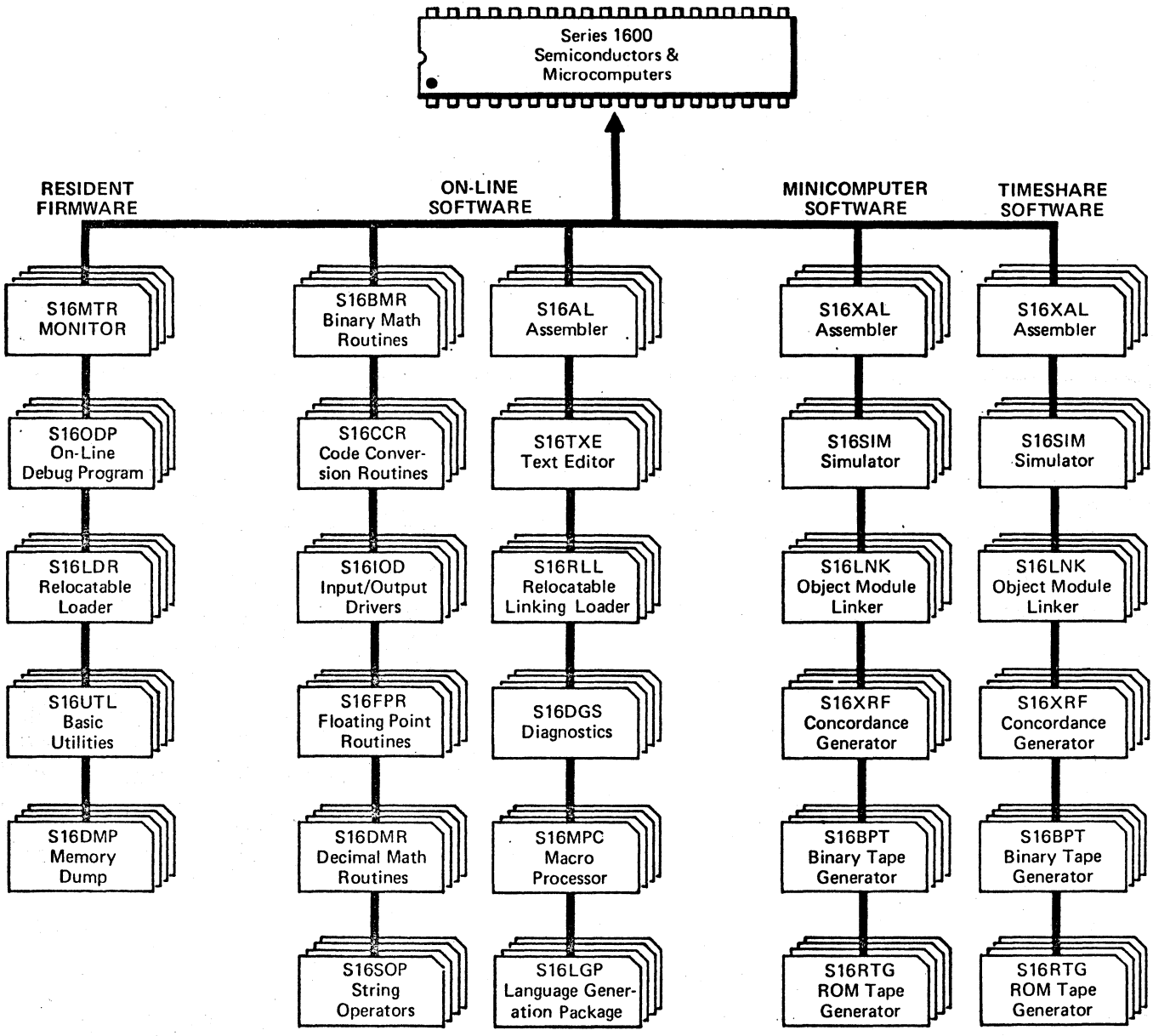
To round out the Series 1600 Microprocessor System, detailed documentation and specifications are available on all parts of the product family. These manuals cover full operational specifications of the hardware and software as well as typical examples of how to use components. In addition, a trained applications staff is available for customer support. The end result is a total product concept designed around a single, high performance architecture with a common software structure and several levels of performance, cost, and flexibility.

THE SERIES 1600 SEMICONDUCTOR LINEUP



*Available Fall 1975

THE SERIES 1600 SOFTWARE LINE UP



RO-3-4096

4096 BIT STATIC READ ONLY MEMORY

GIANT II

GENERAL INSTRUMENT ADVANCED N-CHANNEL ION IMPLANT TECHNOLOGY

FEATURES

- 512 x 8 Organization
- Single +5 Volt Supply
- TTL/DTL Compatible
- Static Operation — no clocks required
- 500 ns. Maximum Access Time
- 150 mW Typical Power
- Tri-State Outputs — under control of 'Output Inhibit' signal
- Totally Automated Custom Programming
- Zener Protected Inputs
- Glass Passivation Protection

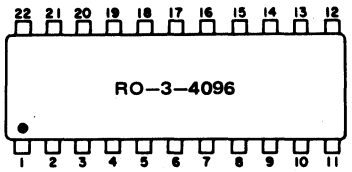
DESCRIPTION

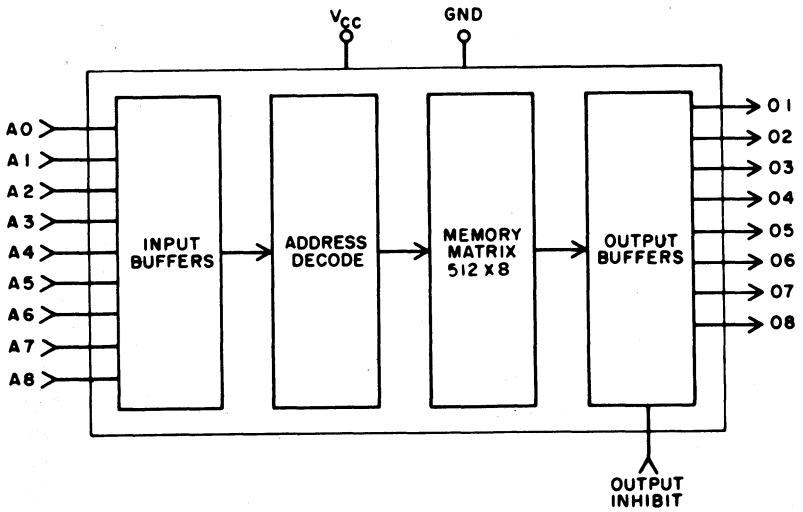
The General Instrument RO-3-4096 is a 4096 bit static Read-Only-Memory. It is organized as 512 eight bit words and requires 9 bits of addressing. An 'Output Inhibit' function is provided to simplify the connection of several ROMs to a common bus. The RO-3-4096 is constructed on a single monolithic chip utilizing low-voltage N-channel Ion Implant technology.

A separate publication, "RO-3-4096 Custom Coding Information," available from GI Sales Offices, describes the punched card and truth table data format for custom programming of the RO-3-4096 memory.

PACKAGE INFORMATION
22 LEAD DUAL IN-LINE

PIN	FUNCTION	PIN	FUNCTION
1	GND	12	Output Inhibit
2	A8	13	N.C.
3	A7	14	N.C.
4	A6	15	Out 8
5	A5	16	Out 7
6	A4	17	Out 6
7	A3	18	Out 5
8	Vcc	19	Out 4
9	A2	20	Out 3
10	A1	21	Out 2
11	A0	22	Out 1





RO-3-5120

5120 BIT STATIC READ ONLY MEMORY

GENERAL INSTRUMENT ADVANCED N-CHANNEL ION IMPLANT TECHNOLOGY

FEATURES

- 512 x 10 Organization
- Single +5 Volt Supply
- TTL/DTL Compatible
- Static Operation — no clocks required
- 500 ns. Maximum Access Time
- 150 mW Typical Power
- Tri-State Outputs — under control of 'Output Inhibit' signal
- Totally Automated Custom Programming
- Zener Protected Inputs
- Glass Passivation Protection

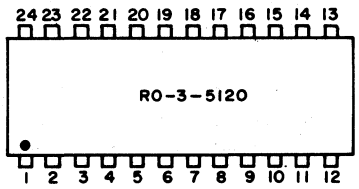
DESCRIPTION

The General Instrument RO-3-5120 is a 5120 bit static Read-Only-Memory. It is organized as 512 ten bit words and requires 9 bits of addressing. An 'Output Inhibit' function is provided to simplify the connection of several ROMs to a common bus. The RO-3-5120 is constructed on a single monolithic chip utilizing low-voltage N-channel Ion Implant technology.

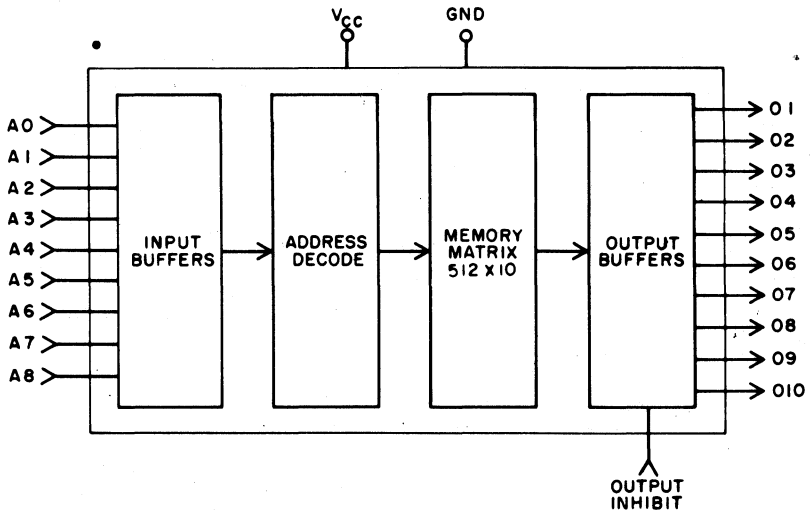
A separate publication, "RO-3-5120 Custom Coding Information," available from GI Sales Offices, describes the punched card and truth table data format for custom programming of the RO-3-5120 memory.

PACKAGE INFORMATION
24 LEAD DUAL IN-LINE

PIN	FUNCTION	PIN	FUNCTION
1	GND	13	Output Inhibit
2	A8	14	Out 10
3	A7	15	Out 9
4	A6	16	Out 8
5	A5	17	Out 7
6	A4	18	Out 6
7	A3	19	Out 5
8	N.C.	20	Out 4
9	Vcc	21	Out 3
10	A2	22	Out 2
11	A1	23	Out 1
12	A0	24	N.C.



BLOCK DIAGRAM



RO-3-16384

16,384 BIT STATIC READ ONLY MEMORY

GIANT II

GENERAL INSTRUMENT ADVANCED N-CHANNEL ION IMPLANT TECHNOLOGY

FEATURES

- 4096 x 4 Organization
- Single +5 Volt Supply
- TTL/DTL Compatible
- Static Operation—no clocks required
- Address/Chip Select Latch Input—may be used to gate in new Address or Chip Select inputs
- 1 μs Maximum Access Time
- 250 mW Typical Power
- Tri-State Outputs—under control of 3 programmable Chip Select inputs
- Totally Automated Custom Programming
- Zener Protected Inputs
- Glass Passivation Protection

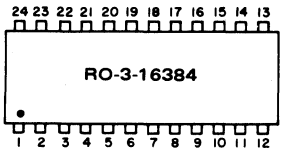
DESCRIPTION

The General Instrument RO-3-16384 is a 16,384 bit static Read-Only-Memory. It is organized as 4096 four bit words and requires 12 bits of addressing. Three programmable Chip Select inputs are provided to simplify the connection of several ROMs to a common bus. The RO-3-16384 is constructed on a single monolithic chip utilizing low-voltage N-channel Ion Implant technology.

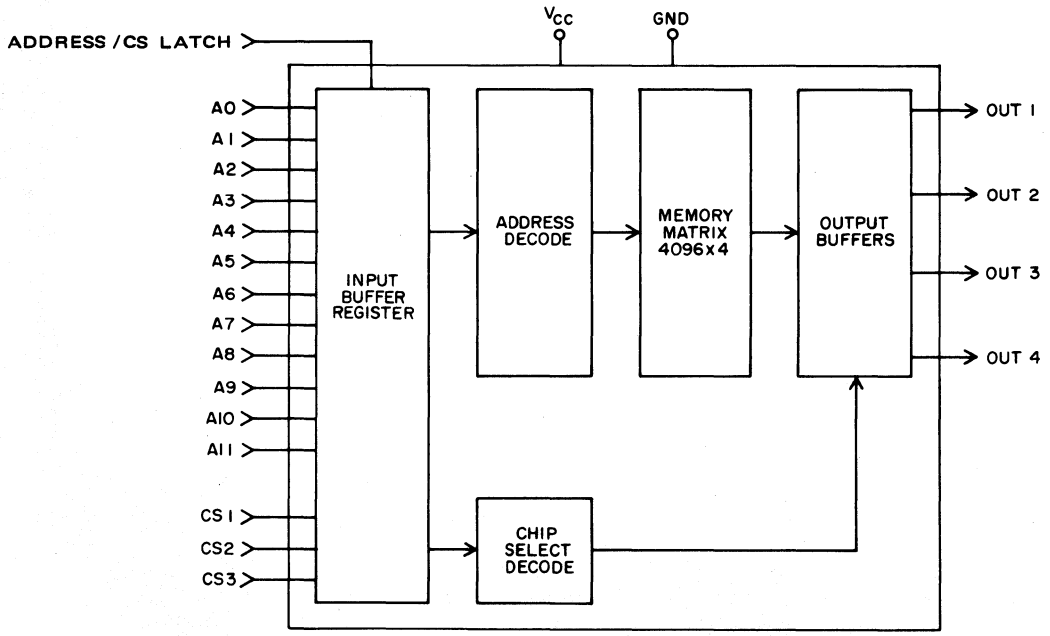
A separate publication, "RO-3-16384 Custom Coding Information," available from GI Sales Offices, describes the punched card and truth table data format for custom programming of the RO-3-16384 memory.

PACKAGE INFORMATION
24 LEAD DUAL IN-LINE

PIN	FUNCTION	PIN	FUNCTION
1	Vcc	13	A9
2	Address/CS Latch	14	A10
3	N.C.	15	N.C.
4	A0	16	Out 4
5	A1	17	Out 3
6	A2	18	Out 2
7	A3	19	Out 1
8	A4	20	A11
9	A5	21	CS3
10	A6	22	CS2
11	A7	23	CS1
12	A8	24	GND



BLOCK DIAGRAM



RO-3-8316A

16384 BIT STATIC READ ONLY MEMORY

GIANT II™

GENERAL INSTRUMENT ADVANCED N-CHANNEL ION IMPLANT TECHNOLOGY

FEATURES

- 2048X8 Organization—ideal for microprocessor memory systems.
- Single +5 Volt Supply
- TTL Compatible—all inputs and outputs.
- Static Operation—no clocks required.
- 850ns Maximum Access Time.
- 200 mW Typical Power
- Three-State Outputs—under the control of three mask-programmable Chip Select inputs to simplify memory expansion.
- Totally Automated Custom Programming.
- Zener Protected Inputs.
- Glass Passivation Protection.

DESCRIPTION

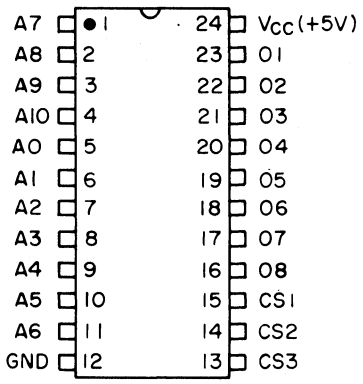
The General Instrument RO-3-8316A is a 16,384 bit static Read-Only Memory organized as 2048 eight bit words and is ideally suited for microprocessor memory applications. Fabricated in GI's advanced GIANT II N-channel Ion-Implant process to enable operation from a single +5 Volt power supply, the RO-3-8316A offers the best combination of high performance, large bit storage, and simple interfacing of any MOS Read-Only Memory available today.

The RO-3-8316A is a direct replacement in pin connection and operation for the Intel 8316A and 2316A.

A separate publication, "RO-3-8316A Custom Coding Information," available from GI Sales Offices, describes the punched card and truth table format for custom programming of the RO-3-8316A memory.

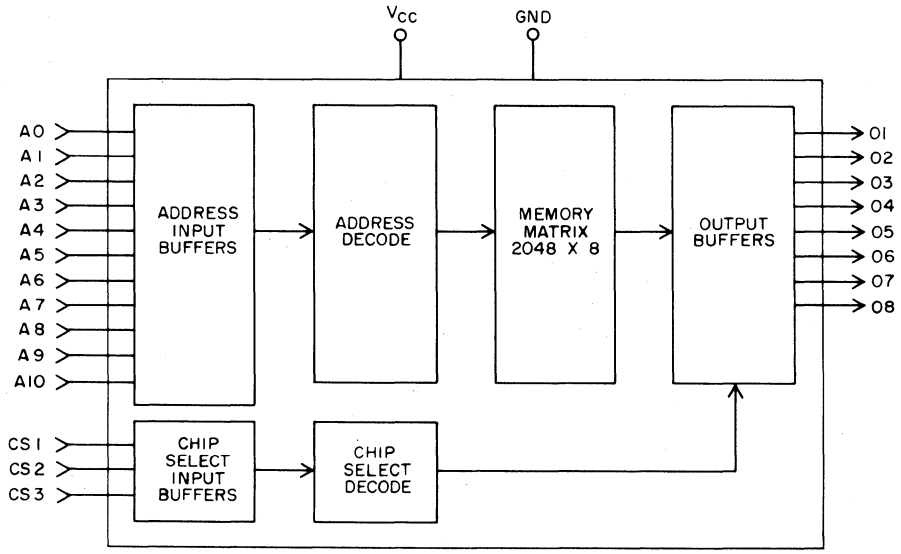
PIN CONFIGURATION

TOP VIEW



A0 – A10	ADDRESS INPUTS
O1 – O8	DATA OUTPUTS
CS1 – CS3	CHIP SELECT INPUTS

BLOCK DIAGRAM



GENERAL INSTRUMENT CORPORATION

PRODUCT INFORMATION

RO-3-20480
20480 BIT STATIC READ ONLY MEMORY

FEATURES

- 2048X10 Organization-ideal for microprocessor memory systems.
- Single +5 Volt Supply
- TTL/DTL Compatible
- Static Operation-no clocks
- 500 ns Maximum Access Time
- 250mW Typical Power
- Tri-State Outputs-under control of Output Inhibit.

- Available 4th Quarter, 1975

PACKAGE INFORMATION

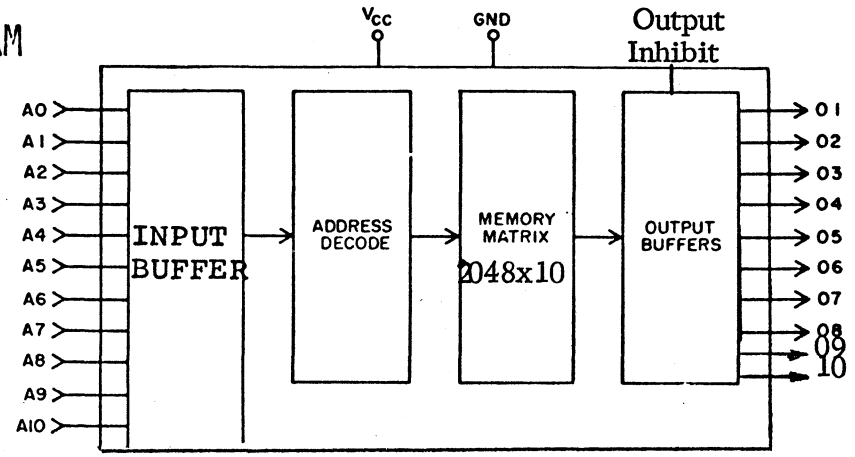
24 LEAD DUAL IN-LINE

PIN	FUNCTION	PIN	FUNCTION
1	GND	13	Output Inhibit
2	A8	14	OUT 10
3	A7	15	OUT 9
4	A6	16	OUT 8
5	A5	17	OUT 7
6	A4	18	OUT 6
7	A3	19	OUT 5
8	A10	20	OUT 4
9	Vcc	21	OUT 3
10	A2	22	OUT 2
11	A7	23	OUT 1
12	A0	24	A9

DESCRIPTION

The General Instrument RO-3-20480 is a 20,480 bit static Read-Only Memory ideally suited for microprocessor memory applications. Fabricated in GI's advanced GIANT II N-channel Ion-Implant process to enable operation from a single +5 Volt power supply, the RO-3-20480 offers the best combination of high performance, large bit storage, and simple interfacing of any MOS Read-Only Memory available today.

BLOCK DIAGRAM



GENERAL INSTRUMENT CORPORATION

PRODUCT INFORMATION

RA-3-4256/RA-3-4256A
1024 BIT STATIC RANDOM ACCESS MEMORIES

FEATURES

- 256X4 Organization
- Single +5Volt Supply
- TTL/DTL Compatible
- True Static-no clocks or refresh
- Power Down Input-together with split +5V inputs, allows for power reduction in a standby mode.
- Choice of Two Memory Speeds-
  RA-3-4256:1μs read/1μs write
  RA-3-4256A:650ns read/650ns write
- Tri-State Outputs-under control of two Chip Select inputs.
- RA-3-4256:Available Now
  RA-3-4256A: April/May 1975

PACKAGE INFORMATION

24 LEAD DUAL IN-LINE

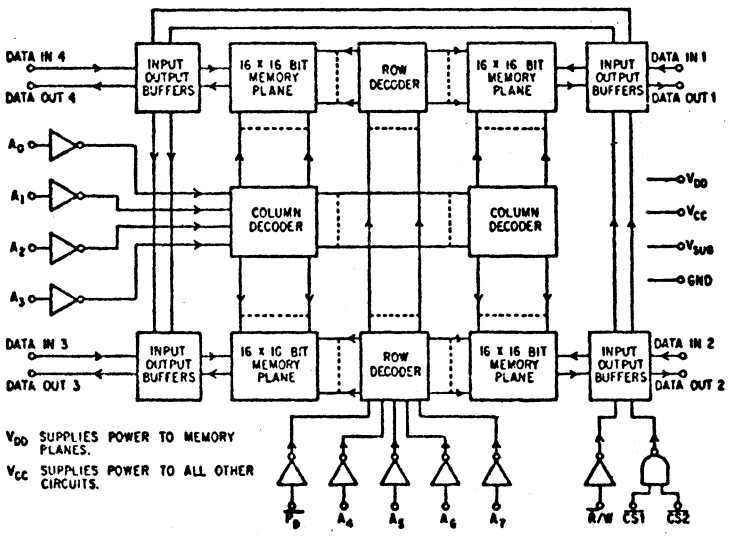
PIN	FUNCTION	PIN	FUNCTION
1	V_{SS} (GND)	13	DATA OUT 3
2	DATA IN 1	14	DATA IN 3
3	$\overline{CS2}$	15	$\overline{POW. DOWN}$
4	$\overline{CS1}$	16	V_{DD} (+5V)
5	A0	17	A7
6	A1	18	A6
7	A2	19	A5
8	A3	20	A4
9	$\overline{R/W}$	21	V_{CC} (+5V)
10	GND	22	DATA IN 4
11	DATA IN 2	23	DATA OUT 4
12	DATA OUT 2	24	DATA OUT 1

DESCRIPTION

The General Instrument RA-3-4256 and RA-3-4256A are 1,024 bit static Random Access Memories ideally suited for small memory systems applications. Fabricated in GI's advanced GIANT II N-channel Ion-Implant process to enable operation from a single +5 Volt power supply, the RA-3-4256 memory series offers the best combination of performance, features, and ease of use of any MOS static Random Access Memories available today.

In addition to the advantage of a very flexible 256X4 memory organization, the RA-3-4256 and RA-3-4256A feature a 'power down' mode in which standby memory power is reduced to half its normal operating level.

BLOCK DIAGRAM



GENERAL INSTRUMENT CORPORATION

PRODUCT INFORMATION

RA-3-4256B/RA-3-4256C
RA-3-4256D/RA-3-4256E
1024 BIT STATIC RANDOM ACCESS MEMORIES

FEATURES

- 256X4 Organization
- Single +5 Volt Supply
- TTL/DTL Compatible
- True Static-no clocks or refresh
- Address/CS Latch(RA-3-4256B,C)
- Output Latch(RA-3-4256D,E)
- Choice of Two Memory Speeds-
  RA-3-4256B,D:1μs read/1μs write
  RA-3-4256C,E:650ns read/650ns write
- Tri-State Outputs-under control of a 'Chip Select' input.
- RA-3-4256B:Available Now
  RA-3-4256C,D,E: April/May 1975

PACKAGE INFORMATION

22 LEAD DUAL IN-LINE

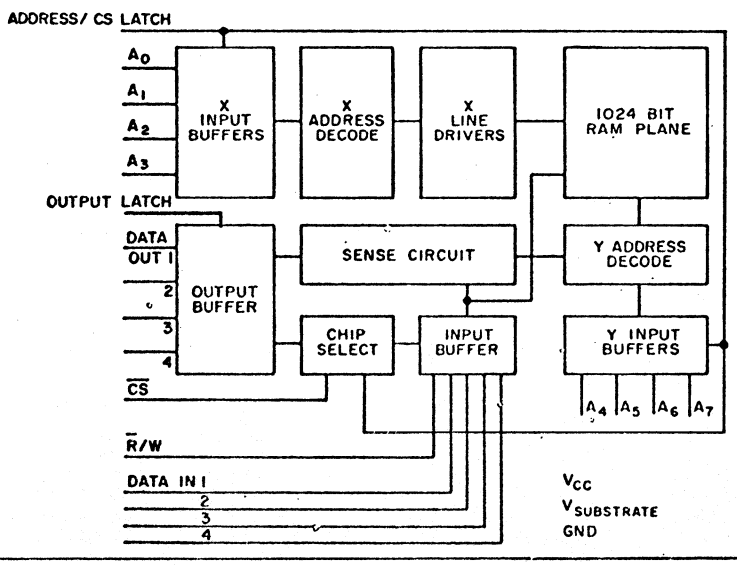
PIN	FUNCTION	PIN	FUNCTION
1	V_{SS} (GND)	12	DATA OUT 1
2	DATA OUT3	13	DATA IN 1
3	DATA IN 3	14	$\overline{\text{CHIP SEL.}}$
4	A7	15	A0
5	A6	16	A1
6	A5	17	A2
7	A4	18	A3
8	LATCH	19	$\overline{\text{R/W}}$
9	V_{CC} (+5V)	20	GND
10	DATA IN 4	21	DATA IN 2
11	DATA OUT 4	22	DATA OUT 2

DESCRIPTION

The General Instrument RA-3-4256B, C, D, and E are 1,024 bit static Random Access Memories ideally suited for small memory systems applications. Fabricated in GI's advanced GIANT II N-channel Ion-Implant process to enable operation from a single +5 Volt power supply, the RA-3-4256 memory series offers the best combination of performance, features, and ease of use of any MOS static Random Access Memories available today.

In addition to the advantage of a very flexible 256X4 memory organization, the RA-3-4256B and C feature an 'Address/Chip Select Latch' and the RA-3-4256D and E feature an 'Output Latch'-each providing even more system design flexibility.

BLOCK DIAGRAM



GENERAL INSTRUMENT CORPORATION

PRODUCT INFORMATION

RA-3-1801/RA-3-1802
1024 BIT STATIC RANDOM ACCESS MEMORIES

FEATURES

- 1024X1 Organization
- Static Memory-no refresh required
- TTL Compatible Inputs(except CS)
- Data Output-TTL(RA-3-1801) or Differential(RA-3-1802)
- Wire-Or'able Outputs-under control of a 'Chip Select' input.
- Choice of Two Memory Speeds-
  RA-3-1801:90ns read/175ns write
  RA-3-1802:70ns read/165ns write
- Available Now

PACKAGE INFORMATION

18 LEAD DUAL IN-LINE

PIN	FUNCTION	PIN	FUNCTION
1	$\overline{R}/W$	10	V_{SX}(-3.5V)
2	DATA OUT	11	A0
3	A9	12	A1
4	V_{RF}(+5V)	13	A4
5	A8	14	GND
6	A7	15	A3
7	A6	16	A2
8	A5	17	V_{DD}(+15V)
9	CHIP SEL.	18	DATA IN

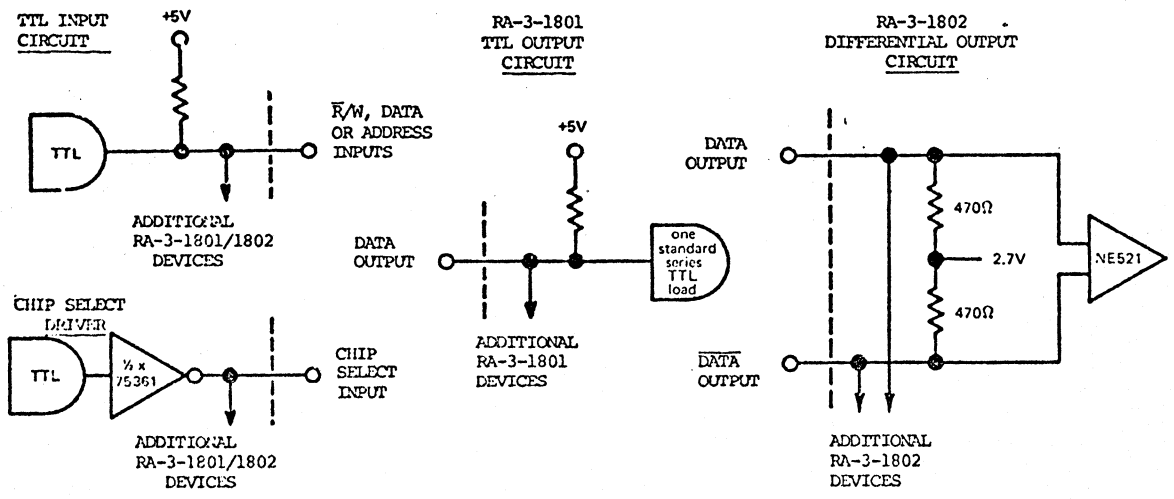
NOTE:Above only for RA-3-1801. RA-3-1802 has $\overline{\text{DATA OUT}}$ on pin 3 and A9 on pin 4.

DESCRIPTION

The General Instrument RA-3-1801 and RA-3-1802 are 1,024 bit static Random Access Memories ideally suited for small and medium size memory system applications where there is an advantage in utilizing a static memory without sacrificing the high speeds that current dynamic memories offer. Featuring an efficient design, TTL compatible except for a +15Volt Chip Select which dynamically accesses the memory, the RA-3-1801 and RA-3-1802 are fabricated in GI's advanced GIANT II N-channel Ion Implant process.

The RA-3-1801 and RA-3-1802 are direct replacements in pin connection and operation for the EM&M/SEMI 1801 and 1802.

INTERFACE CIRCUITS



GENERAL INSTRUMENT CORPORATION

PRODUCT INFORMATION

RA-3-4402
4096 BIT STATIC RANDOM ACCESS MEMORY

FEATURES

- 4096X1 Organization
- Static Memory-no refresh required
- TTL Compatible Inputs(except CS)
- Differential Output-two complementary Data Output signals are provided.
- Wire-Or'able Outputs-under control of a 'Chip Select' input.
- High Speed:200ns access time, 350ns cycle time.
- Low Power:typically 400mW
- Available March/April 1975

PACKAGE INFORMATION

22 LEAD DUAL IN-LINE

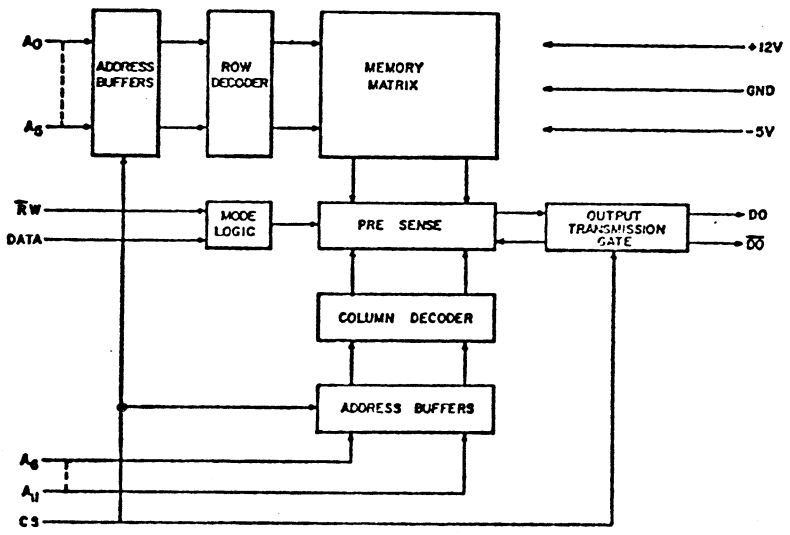
PIN	FUNCTION	PIN	FUNCTION
1	V_{SX}(-5V)	12	$\overline{R}/W$
2	A9	13	A3
3	A10	14	A4
4	A11	15	A5
5	N.C.	16	DATA OUT
6	DATA IN	17	CHIP SEL.
7	$\overline{\text{DATA OUT}}$	18	V_{DD}(+12V)
8	A0	19	A6
9	A1	20	A7
10	A2	21	A8
11	N.C.	22	GND

DESCRIPTION

The General Instrument RA-3-4402 is a 4,096 bit static Random Access Memory ideally suited for memory system applications where there is an advantage in utilizing a static memory without sacrificing the high speeds that current dynamic memories offer. The RA-3-4402 is fabricated in GI's advanced GIANT II N-channel Ion Implant process and features a fully static memory cell to eliminate the need for any refresh or charge-pump circuitry and TTL compatibility except for a +12Volt Chip Select which dynamically accesses the memory.

The RA-3-4402 is a direct replacement in pin connection and operation for the EM&M/SEMI 4402.

BLOCK DIAGRAM



GENERAL INSTRUMENT CORPORATION

PRODUCT INFORMATION

RA-3-4401
4096 BIT STATIC RANDOM ACCESS MEMORY

FEATURES

- 4096X1 Organization
- Static Memory-no refresh required
- TTL Compatible Inputs(except CS)
- TTL Compatible Output
- Wire-Or'able Outputs-under control of a 'Chip Select' input
- High Speed:250ns access time, 400ns cycle time.
- Low Power:typically 400mW
- Available June/July 1975

PACKAGE INFORMATION

22 LEAD DUAL IN-LINE

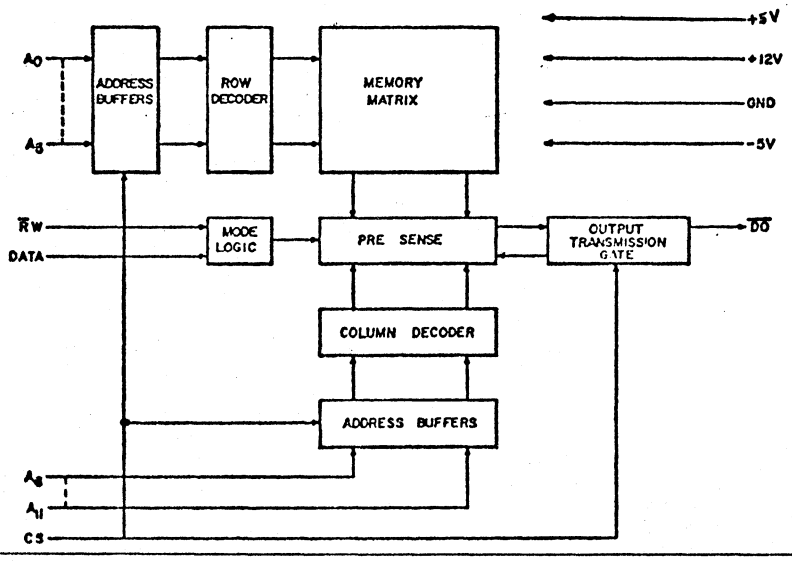
PIN	FUNCTION	PIN	FUNCTION
1	V_{SX}(-5V)	12	$\overline{R}/W$
2	A9	13	A3
3	A10	14	A4
4	A11	15	A5
5	N.C.	16	N.C.
6	DATA IN	17	CHIP SEL.
7	$\overline{\text{DATA OUT}}$	18	V_{DD}(+12V)
8	A0	19	A6
9	A1	20	A7
10	A2	21	A8
11	V_{REF}(+5V)	22	GND

DESCRIPTION

The General Instrument RA-3-4401 is a 4,096 bit static Random Access Memory ideally suited for memory system applications where there is an advantage in utilizing a static memory without sacrificing the high speeds that current dynamic memories offer. The RA-3-4401 is fabricated in GI's advanced GIANT II N-channel Ion Implant process and features a fully static memory cell to eliminate the need for any refresh or charge-pump circuitry and TTL compatibility except for a +12Volt Chip Select which dynamically accesses the memory.

The RA-3-4401 is a direct replacement in pin connection and operation for the EM&M/SEMI 4401.

BLOCK DIAGRAM



GIC1600 MICROCOMPUTER HARDWARE

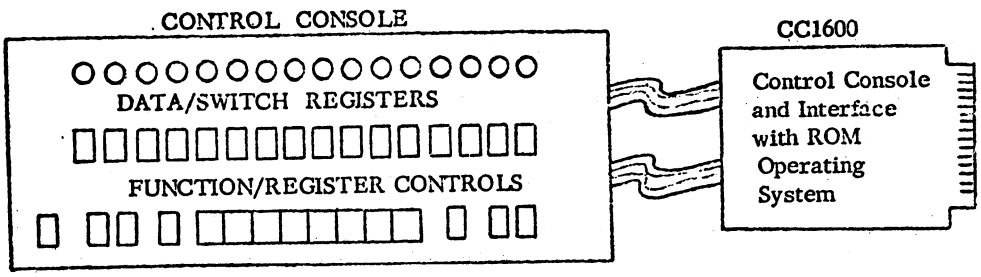
GENERAL DESCRIPTION

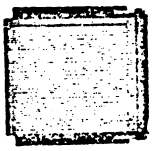
To simplify microprocessor hardware and software development, speed the product design cycle, and support product prototyping, a microcomputer development system and its associated components are a must. The Series 1600 family fills these requirements with the GIC1600 Microcomputer - a versatile, general purpose, stand alone computer system built with the Series 1600 Semiconductor Components.

The GIC1600 utilizes a totally modular design allowing the system designer maximum configurability. The system provides direct addressing to 65K words, unlimited DMA channels, and a multi-line/multi-level nested interrupt system with full priority resolution and self-identifying addresses. All control and timing signals as well as data and address buses are fully buffered and available for use in expanding memory or designing specialized I/O interfaces.

The basic hardware includes a card cage, front panel, and four printed circuit boards: MC1600 Microcomputer, RM1600 RAM Memory, CC1600 Control Console Interface, and I/O1600 Input/Output Interface. Up to 10 additional cards of any types can be added as required. With the additon of a TTY and/or a high speed reader/punch unit, the GIC1600 becomes a test bed for customer designed interfaces and related hardware as well as a full program preparation facility. Its resident firmware operating system provides many sophisticated utility functions. A comprehensive S16ODP On-Line Debug Program allows testing of hardware and software directly on the system in real time while the resident S16LDR Relocating Loader totally eliminates the annoying bootstrap procedure. The On-Line Software Package provides the necessary propram preparation aids such as the S16AL Assembler and S16TXE Text Editor.

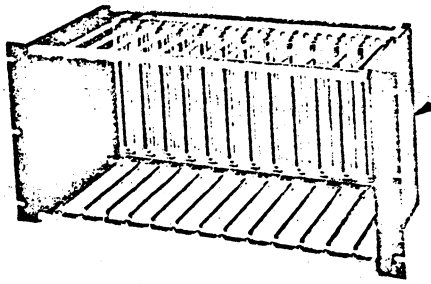
For the small volume user or those that want to avoid tooling and testing costs, all of the card level modules of the GIC1600 are available on an OEM basis for further system integration. These card level subsystems can be easily mixed and matched to meet any system requirement and additional modules are continually being added to the family. In many cases, this is an effective way to reap the benefits of low cost computer hardware created by the microprocessor revolution.





EX1600 EXTENDER CARD

WW1600 WIRE WRAP CARD



CF1600 CARD FILE AND PRINTED CIRCUIT BACKPLANE

I/O1600 - Peripheral Interface for TTY, RS232, and High Speed Paper Tape Reader/Punch

I/O1601 - Peripheral Interface as above with RS232 compatibility

RM1600 - Memory Card with 2Kx16 RAM for program and data storage

RM1601 - Memory Card with 8Kx16 RAM for program and data storage

MC1600 - Microcomputer with buffered buses, priority interrupts, and DMA channels

MC1601 - Microcomputer as above with Real Time Clock & Power Fail Interrupt

GENERAL INSTRUMENT CORPORATION
MICROELECTRONICS

EASTERN AREA SALES HEADQUARTERS, 600 W. John St., Hicksville, N.Y. 11802, (516) 733-3107
CENTRAL AREA SALES HEADQUARTERS, 3101 West Pratt Blvd., Chicago, Ill. 60645, (312) 388-9200
WESTERN AREA SALES HEADQUARTERS, 7120 Havenhurst Ave., Van Nuys, Calif. 91406, (213) 781-0489

GENERAL INSTRUMENT CANADA LTD., 61 Industry St., Toronto 337, Ontario, Canada, Tel: (416) 763-4133
GENERAL INSTRUMENT MICROELECTRONICS LTD., 57/61 Mortimer St., London, W1N 7TD, England, Tel: 01-636-7022
GENERAL INSTRUMENT EUROPE S.P.A., Piazza Amendola 9, 20149 Milano, Italy, Tel: 469-7751
GENERAL INSTRUMENT FRANCE SA, 11-13 Rue Gandon, 75-Paris-13eme, France, Tel: 588-74-31
GENERAL INSTRUMENT DEUTSCHLAND GMBH, Neumarkter Strasse 61, 8 Munich 80, West Germany, Tel: 452239/450181
GENERAL INSTRUMENT INTERNATIONAL CORP., Fukide Building, 17 Fukide-cho, Minato-ku, Tokyo 105, Japan, Tel: (03) 437-0281-5
GENERAL INSTRUMENT OF TAIWAN LTD., P.O. Box 22226, Taipei, Taiwan, Tel: 933861-3

MC1600/1601 MICROCOMPUTER MODULE

Complete microcomputer module with system clocks, memory interface, and fully buffered Address, Data, and Control Buses

- Built with General Instrument's CP1600 MOS N-Channel microprocessor
- 87 Basic Instructions
- 8 Internal Registers
- Two Phase CPU Clock - 5 MHz
- Direct Addressing up to 65K memory space
- Register Addressing up to 65K memory space
- Memory stack pointer
- Two Programmable Interrupt Lines/ Multi-Level and Self Identifying
- DMA Channel Capability
- Real Time Clock (MC1601 only)
- Power Down Monitor (MC1601 only)
- 16 External Sense Conditions for Conditional Branching
- Generalized Initialization Logic

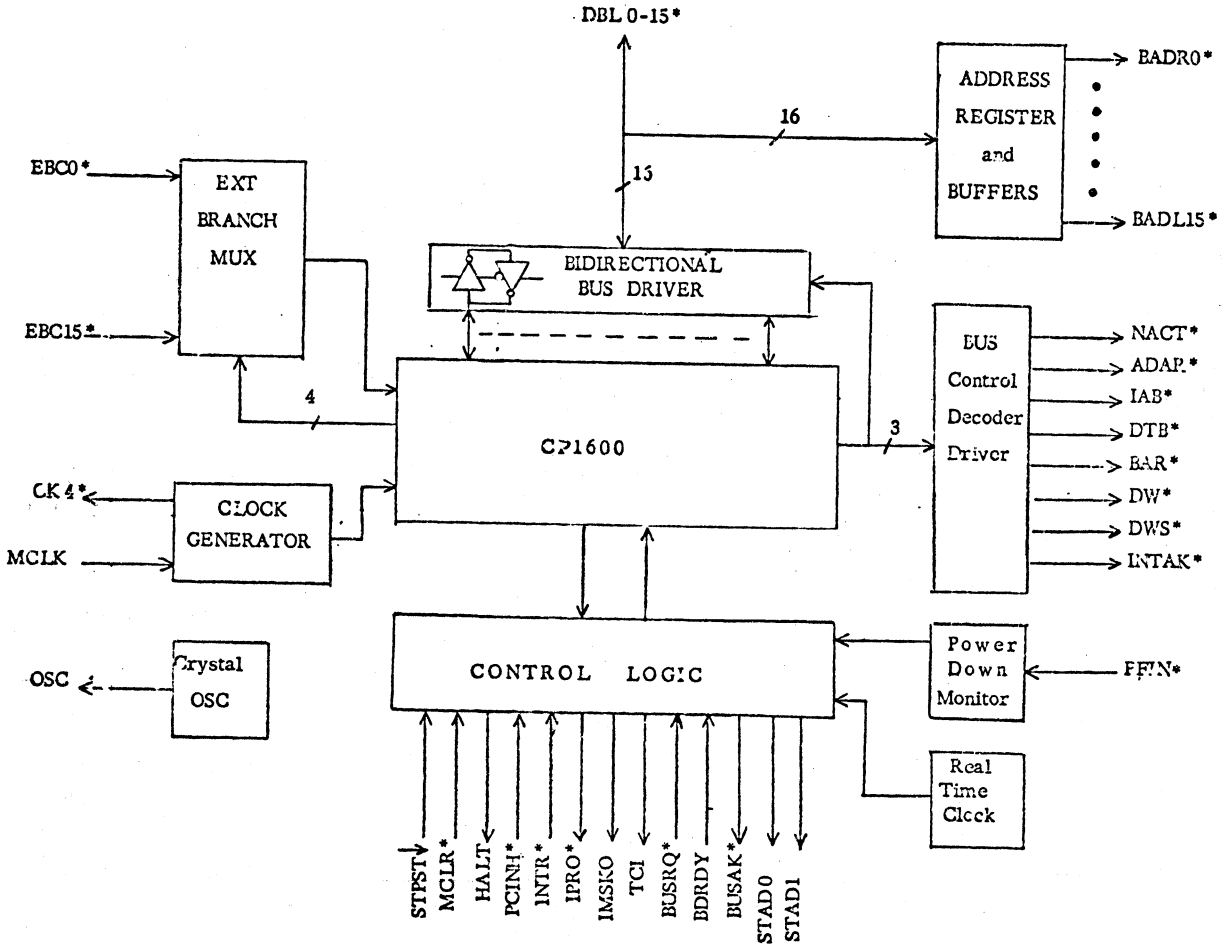
The MC1600/1601 Microcomputer Module is a complete 16-bit parallel processing unit. It contains the hardware necessary to interface with memory and I/O. This is the main module in the GIC1600 System.

The Microcomputer Module is designed around the CP1600, a 16-bit microprocessor on a chip. The MC1600 contains a 16-bit wide bidirectional bus driver, Address Register and Driver, Bus Control Decoder Driver, 10 Megahertz Crystal Oscillator, Clock Driver, and an External Branch Multiplexer. The MC1601 contains, in addition, a Real Time Clock and a Power Down Monitor.

Two line, multi-level interrupt capability and Direct Memory access are provided on this module. In response to an interrupt, the microcomputer automatically saves the current Program Counter on the Memory Stack, resolves interrupt priority and vectors to the device's interrupt service address. The direct memory access capability allows an alternate source to access memory or I/O while temporarily suspending processor operation. At the completion of a DMA operation, normal program execution continues in normal fashion.

Microcomputer Module Specifications

Central Processor:	CP1600 16-Bit microprocessor with 8 general registers, 65K addressing, and memory stacking
Instruction Set:	87 basic with many variations
Memory Space:	65K - 16-bit words
System Clock:	Crystal controlled, 10 MHz $\pm$0.01% Processor cycle time 400 ns
Connector:	Dual 70 pin connector
Board Dimension:	9.75" x 9.25" x .062" printed circuit board
DC Power Requirements:	V_{CC} = +5V $\pm$5% at 0.5A typical V_{DD} = +12V $\pm$5% at 0.1A typical V_{BB} = -12V $\pm$5% at 4 mA
Operating Temperature:	0°C to 55°C



GENERAL INSTRUMENT CORPORATION
MICROELECTRONICS

EASTERN AREA SALES HEADQUARTERS, 600 W. John St., Hicksville, N.Y. 11802, (516) 733-3107
CENTRAL AREA SALES HEADQUARTERS, 3101 West Pratt Blvd., Chicago, Ill. 60645, (312) 388-9200
WESTERN AREA SALES HEADQUARTERS, 7120 Havenhurst Ave., Van Nuys, Calif. 91406, (213) 781-0489

GENERAL INSTRUMENT CANADA LTD., 61 Industry St., Toronto 337, Ontario, Canada, Tel: (416) 763-4133
GENERAL INSTRUMENT MICROELECTRONICS LTD., 57/61 Mortimer St., London, W1N 7TD, England, Tel: 01-636-7022
GENERAL INSTRUMENT EUROPE S.P.A., Piazza Amendola 9, 20149 Milano, Italy, Tel: 469-7751
GENERAL INSTRUMENT FRANCE SA, 11-13 Rue Gandon, 75-Paris-13eme, France, Tel: 588-74-31
GENERAL INSTRUMENT DEUTSCHLAND GMBH, Neumarkter Strasse 61, 8 Munich 80, West Germany, Tel: 452239/450181
GENERAL INSTRUMENT INTERNATIONAL CORP., Fukide Building, 17 Fukide-cho, Minato-ku, Tokyo 105, Japan, Tel: (03) 437-0281-5
GENERAL INSTRUMENT OF TAIWAN LTD., P.O. Box 22226, Taipei, Taiwan, Tel: 933861-3

I/O 1600/1601 INPUT/OUTPUT MODULE

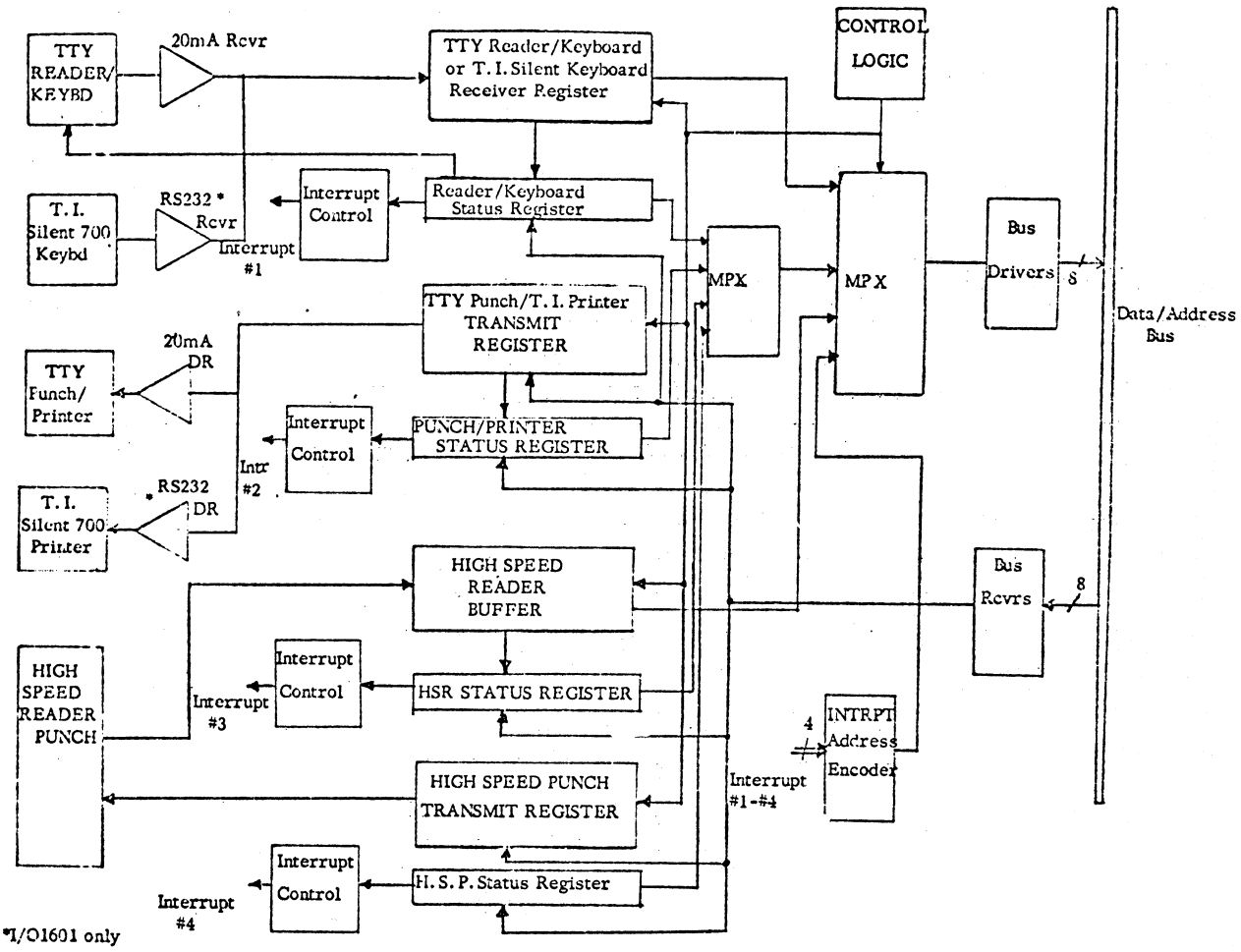
- Teletype Asynchronous Transmitter/Receiver and Control (UAR/T)
- High Speed Reader/Punch Controller
- Silent 700 DATA Terminal Interface (RS-232) *
- Interfaces directly with MC1600/1601 Microcomputer Module
- TTL Compatible

The I/O 1600 Input Output Module handles full duplex communication between a Teletype, High Speed Reader/Punch combination or any RS232*compatible device and the MC1600/1601 Microcomputer Module. The I/O Module has complete interrupt capability with four separate channels: two for the receiver section, High Speed Reader and TTY Reader/Keyboard; and two for the transmitter section, High Speed Punch and TTY Punch/Printer. These four interrupt channels operate independently with the receiver sections taking priority over the transmitter sections on simultaneous interrupts. The High Speed Reader/Punch has a higher priority than the TTY. Electrically, the I/O 1600 Module has a 20 mA current loop for TTY operation and a TTY reader control line which allows the microprocessor to control the Teletype reader during on-line operation. The High Speed Reader/Punch interface controls a Remex Model 6375 Reader/Punch combination capable of reading paper tape at 300 characters per second and punching tape at 75 characters per second. The I/O 1601 module provides the additional capability of interfacing with any RS-232 compatible terminal.

I/O Module Specification

I/O Interface:	TTY: 20 mA current loop TTY Reader Control: Discrete relay interface High Speed Reader: 300 character per sec. High Speed Punch: 75 character per sec. Serial Line: RS-232 compatible*
Peripheral Connection:	10 wire flexible cable connector for TTY. 34 wire flexible cable connector for High Speed Reader/Punch.
Data:	8-bit bytes
Board Dimensions:	9.75" x 9.25" x .062" Printed Circuit Board
Operating Temperature:	$0^{\circ}C$ to $55^{\circ}C$
DC Power Requirement:	V_{CC} +5V $\pm$5% at .5A typical +12V $\pm$5% at .2A typical -12V $\pm$5% at .2A typical

*I/O1601 only



GENERAL INSTRUMENT CORPORATION
MICROELECTRONICS

EASTERN AREA SALES HEADQUARTERS, 600 W. John St., Hicksville, N.Y. 11802, (516) 733-3107
CENTRAL AREA SALES HEADQUARTERS, 3101 West Pratt Blvd., Chicago, Ill. 60645, (312) 388-9200
WESTERN AREA SALES HEADQUARTERS, 7120 Havenhurst Ave., Van Nuys, Calif. 91406, (213) 781-0489

GENERAL INSTRUMENT CANADA LTD., 61 Industry St., Toronto 337, Ontario, Canada, Tel: (416) 763-4133
GENERAL INSTRUMENT MICROELECTRONICS LTD., 57/61 Mortimer St., London, W1N 7TD, England, Tel: 01-636-7022
GENERAL INSTRUMENT EUROPE S.P.A., Piazza Amendola 9, 20149 Milano, Italy, Tel: 469-7751
GENERAL INSTRUMENT FRANCE SA, 11-13 Rue Gandon, 75-Paris-13eme, France, Tel: 588-74-31
GENERAL INSTRUMENT DEUTSCHLAND GMBH, Neumarkter Strasse 61, 8 Munich 80, West Germany, Tel: 452239/450181
GENERAL INSTRUMENT INTERNATIONAL CORP., Fukide Building, 17 Fukide-cho, Minato-ku, Tokyo 105, Japan, Tel: (03) 437-0281-5
GENERAL INSTRUMENT OF TAIWAN LTD., P.O. Box 22226, Taipei, Taiwan, Tel: 933861-3

RM1600 MEMORY MODULE

- 2048 — 16-bit words per Module
- Static memory, no clocks required
- Single +5 Volt Supply
- Byte or Word Capability
- Module decoding for 65K memory expansion
- 550 ns Read/Write Cycle Time
- Open Collector TTL Output - 30 Loads
- Buffered TTL Inputs - 1 Load

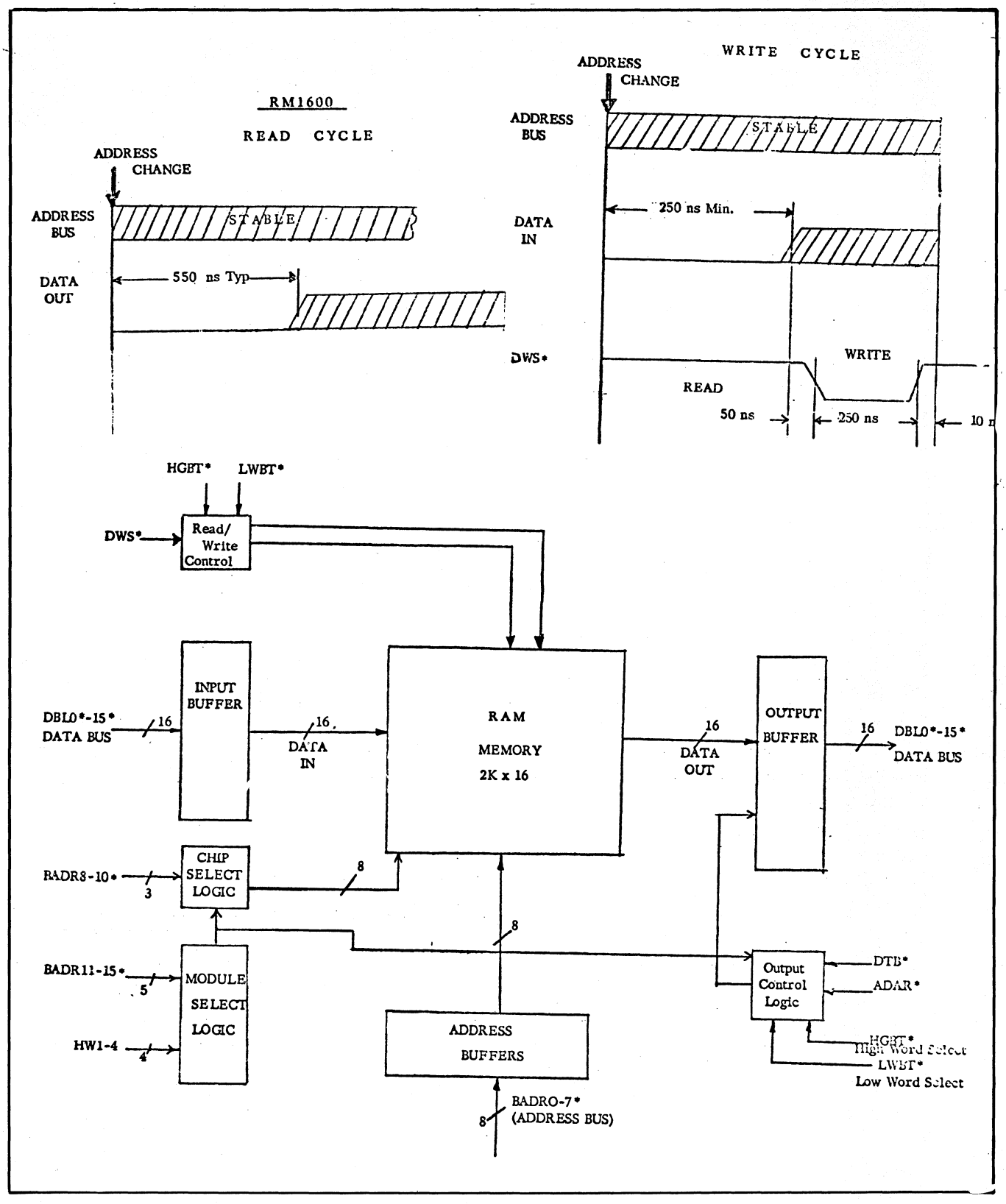
The RM1600 Memory Card is a standard 2Kx16 memory module for use in the GIC1600 Microcomputer System. This memory card contains address and data buffers, read/write circuits, low or high byte word selection logic, and is implemented with General Instrument RA-3-4256 1024 bit static Random Access Memory. There are thirty-two 22 pin 256x4 static RAM's packaged on a 9.75"x9.25"x.062 printed circuit board.

The Address bus inputs from the GIC1600 to the memory card are buffered to provide the necessary address inputs to the RAM's. The sixteen data outputs from the RAM are buffered onto the Data bus of the GIC1600 System.

If more than one 2K memory card is used in the GIC1600 System, provisions are provided for proper selection of 2K increments.

RM1600 Module Specification

Memory Capacity	2K x 16 bits
Cycle Time	550 ns
Memory Expansion	65 K (32 modules)
Board Dimensions	9.75" x 9.25" x .062" printed circuit board with 140 pin I/O connector
Interface	TTL compatible inputs: open collector outputs
Operating Temperature	$0^{\circ}C$ to $55^{\circ}C$
Power Requirements	+5V $\pm$ 5% at 2.0 A typical



GENERAL INSTRUMENT CORPORATION
MICROELECTRONICS

EASTERN AREA SALES HEADQUARTERS, 600 W. John St., Hicksville, N.Y. 11802, (516) 733-3107
CENTRAL AREA SALES HEADQUARTERS, 3101 West Pratt Blvd., Chicago, Ill. 60645, (312) 388-9200
WESTERN AREA SALES HEADQUARTERS, 7120 Havenhurst Ave., Van Nuys, Calif. 91406, (213) 781-0489

GENERAL INSTRUMENT CANADA LTD., 61 Industry St., Toronto 337, Ontario, Canada, Tel: (416) 763-4133
GENERAL INSTRUMENT MICROELECTRONICS LTD., 57/61 Mortimer St., London, W1N 7TD, England, Tel: 01-636-7022
GENERAL INSTRUMENT EUROPE S.P.A., Piazza Amendola 9, 20149 Milano, Italy, Tel: 469-7751
GENERAL INSTRUMENT FRANCE SA, 11-13 Rue Gandon, 75-Paris-13eme, France, Tel: 588-74-31
GENERAL INSTRUMENT DEUTSCHLAND GMBH, Neumarkter Strasse 61, 8 Munich 80, West Germany, Tel: 452239/450181
GENERAL INSTRUMENT INTERNATIONAL CORP., Fukide Building, 17 Fukide-cho, Minato-ku, Tokyo 105, Japan, Tel: (03) 437-0281-5
GENERAL INSTRUMENT OF TAIWAN LTD., P.O. Box 22226, Taipei, Taiwan, Tel: 933861-3

CC1600 CONTROL CONSOLE MODULE

- 16-bit Data/Address Display
- 16-bit Switch Register
- Easy to use Control Panel

  Display/Modify all 8 internal registers
  Display/Modify the CPU Status Word
  Display/Modify all 65K Memory Space
  Single Instruction operation
  Program Counter Inhibit capability

  ROM based Operating System

  Conversational Monitor
  On-Line Debug Program/Software Breakpoints
  Relocating Loader (Eliminates Bootstrap)
  Memory Dump Program
  General Utilities/Input-Output Drivers

  Standard 19" rack mountable Control Panel

The CC1600 Control Console Module is designed to provide a convenient method of controlling and monitoring the GIC1600 System. The CC1600 module consists of a front panel and printed circuit card that are connected with two flex cables. The card contains the control logic to handle all front panel commands as well as the required interrupt logic to interface with the Microcomputer Module.

The Control Console Module consists of six control ROMs, scratch pad memory (256x16), 16-bit Switch Regist;er, 16-bit Display Register, several action switches and the control logic to service any front panel request.

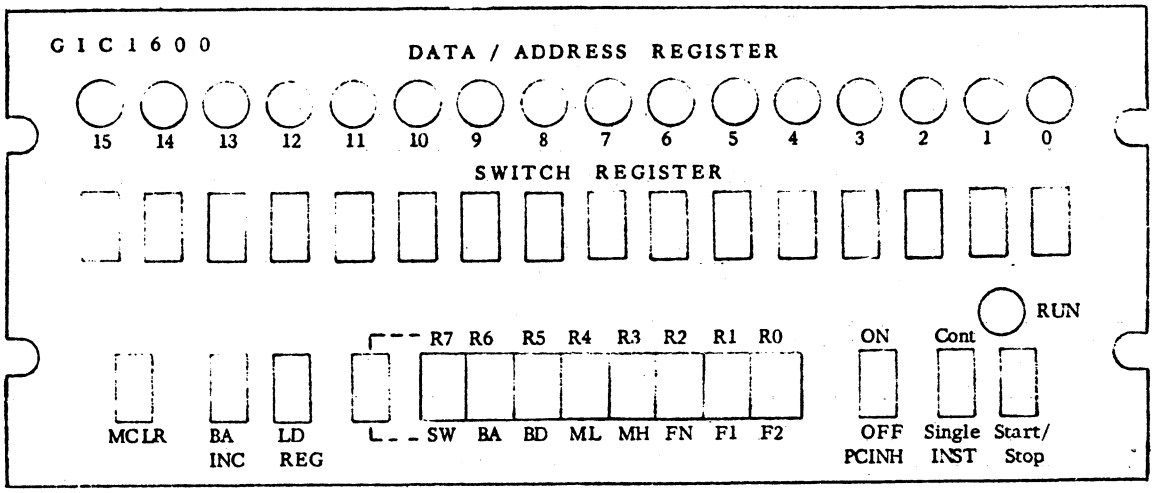
All functional operations for the Control Console are performed by the execution of program stored in the control ROMs. Pressing any action switch on the Control Console results in an interrupt request to the CPU. After this interrupt is acknowledged the CC1600 supplies the starting address of the Control Console service routine which performs the required function. In addition, the program automatically stores all CPU registers in the scratch pad memory which is accessible via front panel selection. Consequently, whenever the CPU is in the HALT mode, the Control Console has direct access to all updated CPU information.

The control ROMs also contain all the firmware necessary for the development of microprocessor based systems. An On-Line Debug program is included so that software breakpoints and memory search routines may be executed. The system monitor allows the user to maintain conversational control via teletype interaction. The Relocating Loader can be used to input data from either a TTY or High Speed Reader while the

Memory Dump program allows any block of memory to be transformed onto paper tape media.

Control Console Module Specifications

Word size:	16 bits
Front panel:	19" x 7" rack mountable panel/two 36 wire flex cable connection to interface board
Board Dimensions:	9.75" x 9.25" x .062" printed circuit board
Connector:	Dual 70 Pin
Operating temperature:	$0^{o}C$ to $55^{o}C$
DC Power Requirement:	V_{CC} = +5V $\pm$ 5% at 1A typical



GENERAL INSTRUMENT CORPORATION
MICROELECTRONICS

EASTERN AREA SALES HEADQUARTERS, 600 W. John St., Hicksville, N.Y. 11802, (516) 733-3107
CENTRAL AREA SALES HEADQUARTERS, 3101 West Pratt Blvd., Chicago, Ill. 60645, (312) 388-9200
WESTERN AREA SALES HEADQUARTERS, 7120 Havenhurst Ave., Van Nuys, Calif. 91406, (213) 781-0489

GENERAL INSTRUMENT CANADA LTD., 61 Industry St., Toronto 337, Ontario, Canada, Tel: (416) 763-4133
GENERAL INSTRUMENT MICROELECTRONICS LTD., 57/61 Mortimer St., London, W1N 7TD, England, Tel: 01-636-7022
GENERAL INSTRUMENT EUROPE S.P.A., Piazza Amendola 9, 20149 Milano, Italy, Tel: 469-7751
GENERAL INSTRUMENT FRANCE SA, 11-13 Rue Gandon, 75-Paris-13eme, France, Tel: 588-74-31
GENERAL INSTRUMENT DEUTSCHLAND GMBH, Neumarkter Strasse 61, 8 Munich 80, West Germany, Tel: 452239/450181
GENERAL INSTRUMENT INTERNATIONAL CORP., Fukide Building, 17 Fukide-cho, Minato-ku, Tokyo 105, Japan, Tel: (03) 437-0281-5
GENERAL INSTRUMENT OF TAIWAN LTD., P.O. Box 22226, Taipei, Taiwan, Tel: 933861-3

GP 1600 GENERAL PURPOSE I/O MODULE

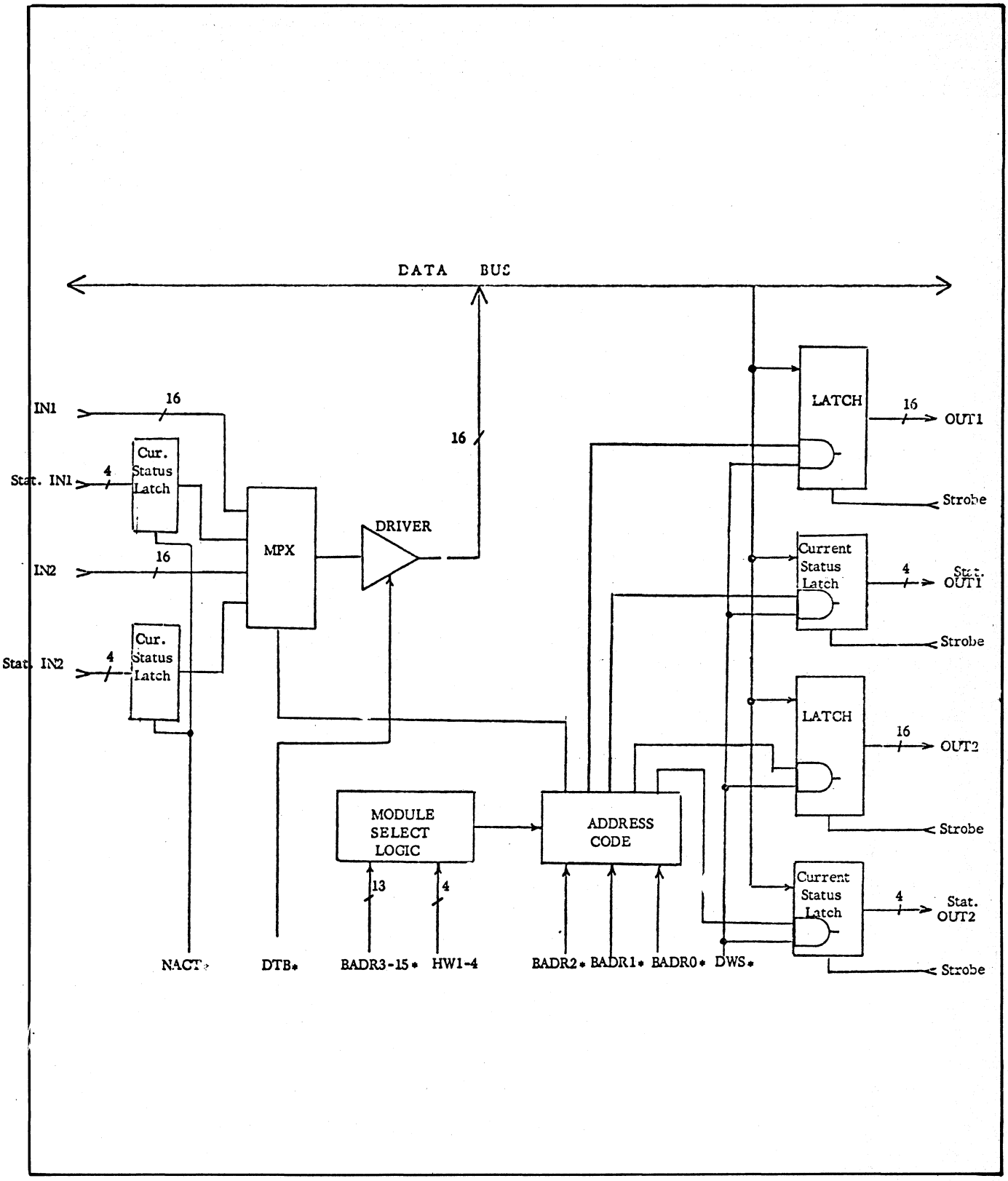
- 4 Addressable Input Ports
- 4 Addressable Output Ports
- Interface directly with MC1600/1601 Module
- Space provided for I/O Control Logic

The GP1600 Module is a general purpose Input/Output Module with four 16-bit and four 4-bit software addressable ports. Each port can be used either for control or data purposes and is accessible under software control. Address decoding is provided on this card although specific port assignment is determined by backplane selection within a defined address space. It is therefore possible to use more than one GP1600 in a given system.

Connection to a given peripheral device is accomplished by a flat ribbon cable as connectors are mounted at the top of the module. Space has been provided to accommodate wire wrap sockets so that specific interface circuitry may be incorporated on the module. Control signals have been brought out to wire wrap pins to facilitate prototype development.

GP1600 MODULE SPECIFICATIONS

Word Size	16 bits
Capacity	2 16-bit latching output ports 2 4-bit latching output ports 2 4-bit latching input ports 2 16-bit input ports Module will accept dual-in-line packaged components mounted in standard wirewrap sockets. Locations for 14, 16, 22, 24 pin sockets are available.
Board Dimensions	9.75"x 9.25"x .062" printed circuit board.
Peripheral Connection	Dual 50 wire flexible cable connector on board
Operating Temperature	0^{o}C to 55^{o}C



GENERAL INSTRUMENT CORPORATION
MICROELECTRONICS

EASTERN AREA SALES HEADQUARTERS, 600 W. John St., Hicksville, N.Y. 11802, (516) 733-3107
CENTRAL AREA SALES HEADQUARTERS, 3101 West Pratt Blvd., Chicago, Ill. 60645, (312) 388-9200
WESTERN AREA SALES HEADQUARTERS, 7120 Havenhurst Ave., Van Nuys, Calif. 91406, (213) 781-0489

GENERAL INSTRUMENT CANADA LTD., 61 Industry St., Toronto 337, Ontario, Canada, Tel: (416) 763-4133
GENERAL INSTRUMENT MICROELECTRONICS LTD., 57/61 Mortimer St., London, W1N 7TD, England, Tel: 01-636-7022
GENERAL INSTRUMENT EUROPE S.P.A., Piazza Amendola 9, 20149 Milano, Italy, Tel: 469-7751
GENERAL INSTRUMENT FRANCE SA, 11-13 Rue Gandon, 75-Paris-13eme, France, Tel: 588-74-31
GENERAL INSTRUMENT DEUTSCHLAND GMBH, Neumarkter Strasse 61, 8 Munich 80, West Germany, Tel: 452239/450181
GENERAL INSTRUMENT INTERNATIONAL CORP., Fukide Building, 17 Fukide-cho, Minato-ku, Tokyo 105, Japan, Tel: (03) 437-0281-5
GENERAL INSTRUMENT OF TAIWAN LTD., P.O. Box 22226, Taipei, Taiwan, Tel: 933861-3

RM1601 MEMORY MODULE

- 8192 - 16-bit words per module
- Module decoding for 65K memory expansion
- Byte or Word Capability
- 350 ns read access time
- 450 ns cycle time
- Open collector TTL outputs- 30 loads
- Buffered TTL inputs - 1 load

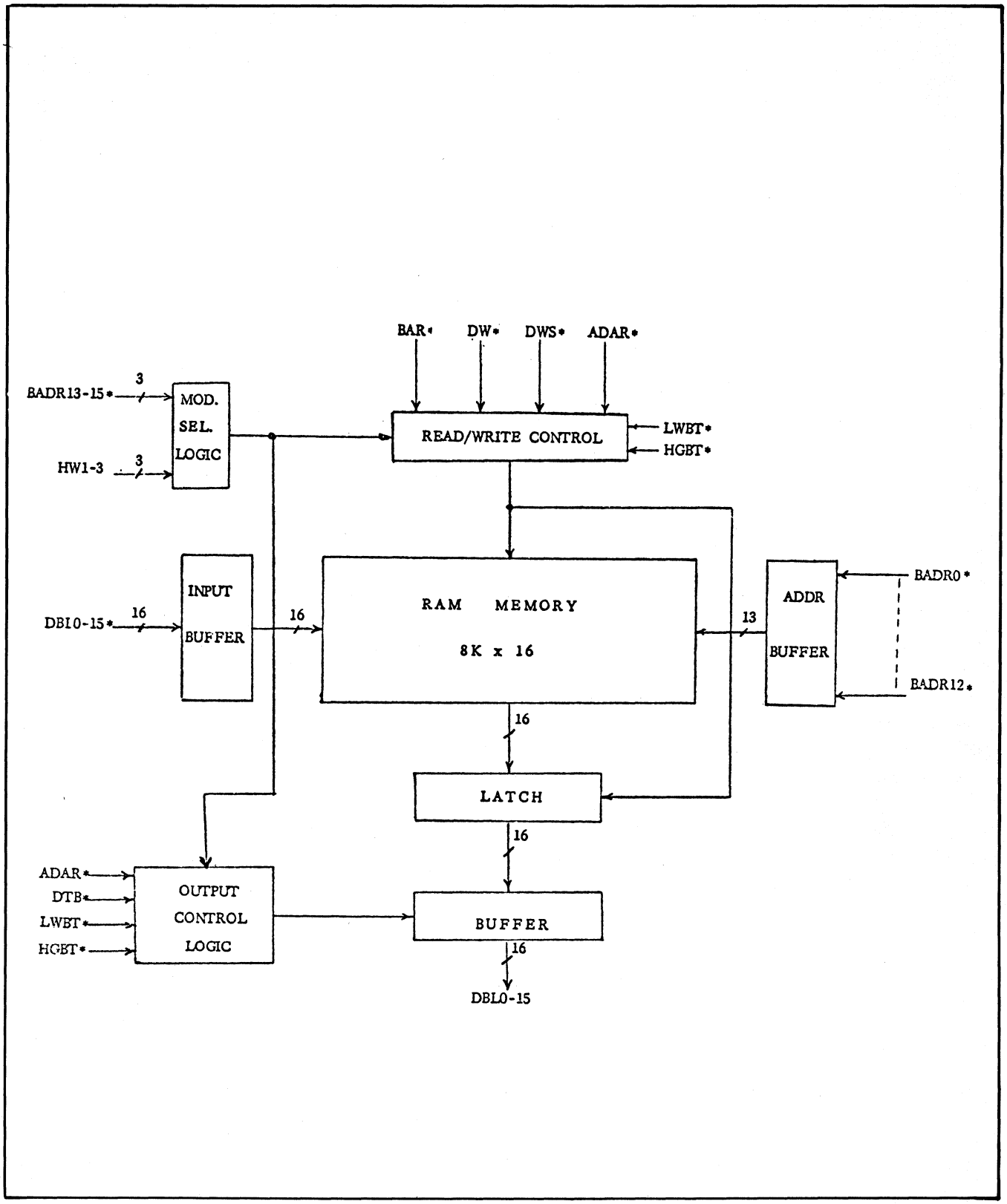
The RM1601 Memory Card is a standard 8K x 16 memory module for use in the GIC1600 Microcomputer System. This memory card contains address and data buffers, read/write circuits, low or high byte word selection logic, and module selection circuitry. There are thirty-two 22 pin 4K x 1 RAMs packaged on a 9.75" x 9.25" x .062 printed circuit board.

The address bus inputs from the GIC1600 to the memory card are buffered to provide the necessary address inputs to the RAMs. The sixteen data outputs from the RAM are latched and buffered onto the Data Bus of the GIC1600 System.

If more than one 8K memory card is used in the GIC1600 System, provisions are provided for proper selection of 8K increments.

RM1601 Module Specification

Memory Capacity	8K x 16 bits
Cycle Time	450 ns
Memory Expansion	65K
Board Dimensions	9.75" x 9.25" x .062 printed circuit board with 140 pin I/O connector
Interface	TTL compatible inputs: Open collector outputs
Operating Temperature	$0^{\circ}C$ to $55^{\circ}C$
Power Requirements	+5V $\pm$ 5% at 0.5 A typical +12V $\pm$ 5% at 0.5 A typical -12V $\pm$ 5% at 0.1 A typical



GENERAL INSTRUMENT CORPORATION
MICROELECTRONICS

EASTERN AREA SALES HEADQUARTERS, 600 W. John St., Hicksville, N.Y. 11802, (516) 733-3107
CENTRAL AREA SALES HEADQUARTERS, 3101 West Pratt Blvd., Chicago, Ill. 60645, (312) 388-9200
WESTERN AREA SALES HEADQUARTERS, 7120 Havenhurst Ave., Van Nuys, Calif. 91406, (213) 781-0489

GENERAL INSTRUMENT CANADA LTD., 61 Industry St., Toronto 337, Ontario, Canada, Tel: (416) 763-4133
GENERAL INSTRUMENT MICROELECTRONICS LTD., 57/61 Mortimer St., London, W1N 7TD, England, Tel: 01-636-7022
GENERAL INSTRUMENT EUROPE S.P.A., Piazza Amendola 9, 20149 Milano, Italy, Tel: 469-7751
GENERAL INSTRUMENT FRANCE SA, 11-13 Rue Gandon, 75-Paris-13eme, France, Tel: 588-74-31
GENERAL INSTRUMENT DEUTSCHLAND GMBH, Neumarkter Strasse 61, 8 Munich 80, West Germany, Tel: 452239/450181
GENERAL INSTRUMENT INTERNATIONAL CORP., Fukide Building, 17 Fukide-cho, Minato-ku, Tokyo 105, Japan, Tel: (03) 437-0281-5
GENERAL INSTRUMENT OF TAIWAN LTD., P.O. Box 22226, Taipei, Taiwan, Tel: 933861-3

PM1600 PROM MEMORY MODULE

- Provides sockets for up to sixteen PROMS (4096 x 16)
- Static Memory - no clocks required
- Field programmable
- Erasable with short wave ultra-violet light
- 1 μs max. access time
- Buffered TTL inputs - 1 load
- Open collector TTL output - 30 loads
- Module decoding for 65K memory expansion

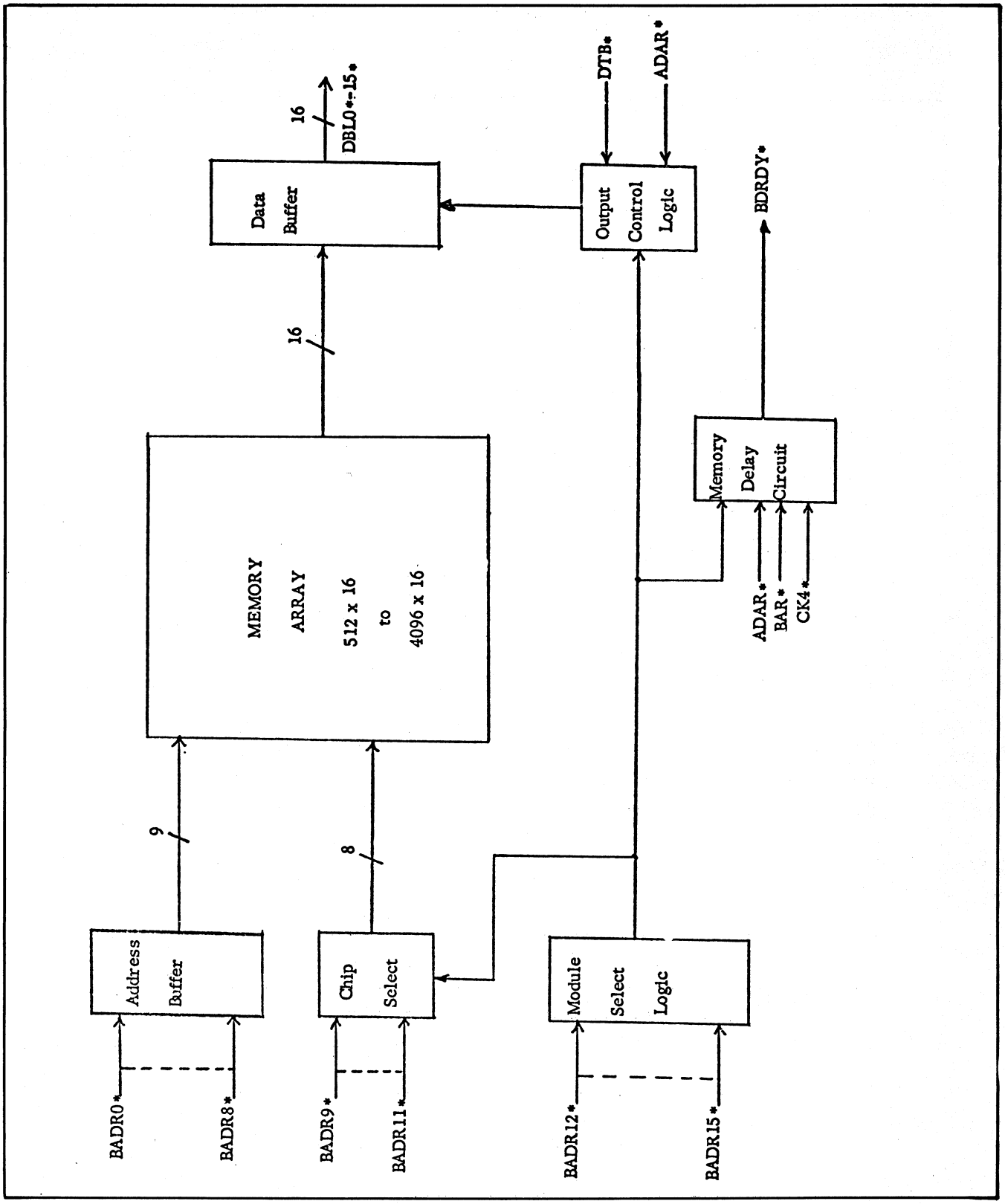
The PM1600 PROM Memory Module Card is a standard 4096 x 16 memory module for use in the GIC1600 Microcomputer System. This memory card has sixteen sockets for 4096-bit static read-only memories organized as 512 x 8 words.

The Address bus inputs from the GIC1600 to the memory card are buffered to provide the necessary address inputs to the PROMS The sixteen data outputs from the PROMS are buffered onto the Data bus of the GIC1600 System. For memories larger than 4K x 16, decoding on the module allows addressing for a total of 65K memory.

A special memory delay circuit is also provided on the board to provide synchronization between the CP1600 microprocessor and PROMS

PROM 1600 MODULE SPECIFICATION

Memory Capacity:	4K x 16 Bits
Access Time:	1 μsec. max.
Memory Expansion:	65K (16 modules)
Board Dimensions:	9.75" x 9.25" x .062" printed circuit board with 140 pin I/O connector
Interface:	TTL compatible inputs; open collector outputs
Operating Temperature:	$0^{\circ}C$ to $55^{\circ}C$
Power Requirements:	+5V ± 5% .5 A typical (board loaded with all 16 PROMS)
	-12V ± 5% .5 A typical (board loaded with all 16 PROMS)



GENERAL INSTRUMENT CORPORATION
MICROELECTRONICS

EASTERN AREA SALES HEADQUARTERS, 600 W. John St., Hicksville, N.Y. 11802, (516) 733-3107
CENTRAL AREA SALES HEADQUARTERS, 3101 West Pratt Blvd., Chicago, Ill. 60645, (312) 388-9200
WESTERN AREA SALES HEADQUARTERS, 7120 Havenhurst Ave., Van Nuys, Calif. 91406, (213) 781-0489

GENERAL INSTRUMENT CANADA LTD., 61 Industry St., Toronto 337, Ontario, Canada, Tel: (416) 763-4133
GENERAL INSTRUMENT MICROELECTRONICS LTD., 57/61 Mortimer St., London, W1N 7TD, England, Tel: 01-636-7022
GENERAL INSTRUMENT EUROPE S.P.A., Piazza Amendola 9, 20149 Milano, Italy, Tel: 469-7751
GENERAL INSTRUMENT FRANCE SA, 11-13 Rue Gandon, 75-Paris-13eme, France, Tel: 588-74-31
GENERAL INSTRUMENT DEUTSCHLAND GMBH, Neumarkter Strasse 61, 8 Munich 80, West Germany, Tel: 452239/450181
GENERAL INSTRUMENT INTERNATIONAL CORP., Fukide Building, 17 Fukide-cho, Minato-ku, Tokyo 105, Japan, Tel: (03) 437-0281-5
GENERAL INSTRUMENT OF TAIWAN LTD., P.O. Box 22226, Taipei, Taiwan, Tel: 933861-3

GIC 1600 ACCESSORY

CF1600 CARD FILE

- 13-position
- P.C. backplane with wirewrap capability
- Rack-mountable
- Cards are keyed to connectors
- 10.5" high x 19.0" wide x 12.0" deep

EX1600 EXTENDER CARD

- For use with all GIC1600 cards
- Two 70-pin connectors

WW1600 WIREWRAP CARD

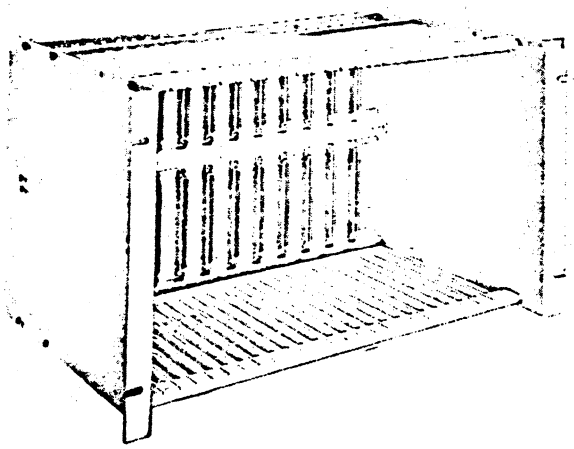
- 126 16-pin positions
- Power and ground planes provided
- 10 Test points on edge of card

The CF1600 Card File is designed to house up to 13 cards of the GIC1600 family. The MC1600/1601 Microcomputer Module, the CC1600 Control Console Module, and the I/O1600/1601 Input/Output Module each have one assigned position. The 10 remaining positions are available for memory modules, general purpose input/output cards, or special interface cards.

The printed circuit backplane parallels the power supply rails and the data, address and control buses for all 13 cards. There are separate voltage and voltage sense lines on the P.C. backplane for the +12v, -12v and +5v supplies. The bus system can be extended to another card file by wirewrapping or soldering a ribbon cable to one of the rear connectors.

The EX1600 Extender Card can be used with any other card of the GIC1600 family.

The WW1600 Wirewrap Card contains 126 16-pin sockets for prototyping special interface cards. Power and ground planes are provided.



CF1600 CARD FILE

Slot	Card
1	MC1600/1601 Microcomputer Module
2	CC1600 Control Console Module
3-12	Memory or I/O Cards
13	I/O 1600/1601 Input/Output Module

GENERAL INSTRUMENT CORPORATION
MICROELECTRONICS

EASTERN AREA SALES HEADQUARTERS, 600 W. John St., Hicksville, N.Y. 11802, (516) 733-3107
CENTRAL AREA SALES HEADQUARTERS, 3101 West Pratt Blvd., Chicago, Ill. 60645, (312) 388-9200
WESTERN AREA SALES HEADQUARTERS, 7120 Havenhurst Ave., Van Nuys, Calif. 91406, (213) 781-0489

GENERAL INSTRUMENT CANADA LTD., 61 Industry St., Toronto 337, Ontario, Canada, Tel: (416) 763-4133
GENERAL INSTRUMENT MICROELECTRONICS LTD., 57/61 Mortimer St., London, W1N 7TD, England, Tel: 01-636-7022
GENERAL INSTRUMENT EUROPE S.P.A., Piazza Amendola 9, 20149 Milano, Italy, Tel: 469-7751
GENERAL INSTRUMENT FRANCE SA, 11-13 Rue Gandon, 75-Paris-13eme, France, Tel: 588-74-31
GENERAL INSTRUMENT DEUTSCHLAND GMBH, Neumarkter Strasse 61, 8 Munich 80, West Germany, Tel: 452239/450181
GENERAL INSTRUMENT INTERNATIONAL CORP., Fukide Building, 17 Fukide-cho, Minato-ku, Tokyo 105, Japan, Tel: (03) 437-0281-5
GENERAL INSTRUMENT OF TAIWAN LTD., P.O. Box 22226, Taipei, Taiwan, Tel: 933861-3

SERIES 1600 SOFTWARE

GENERAL DESCRIPTION

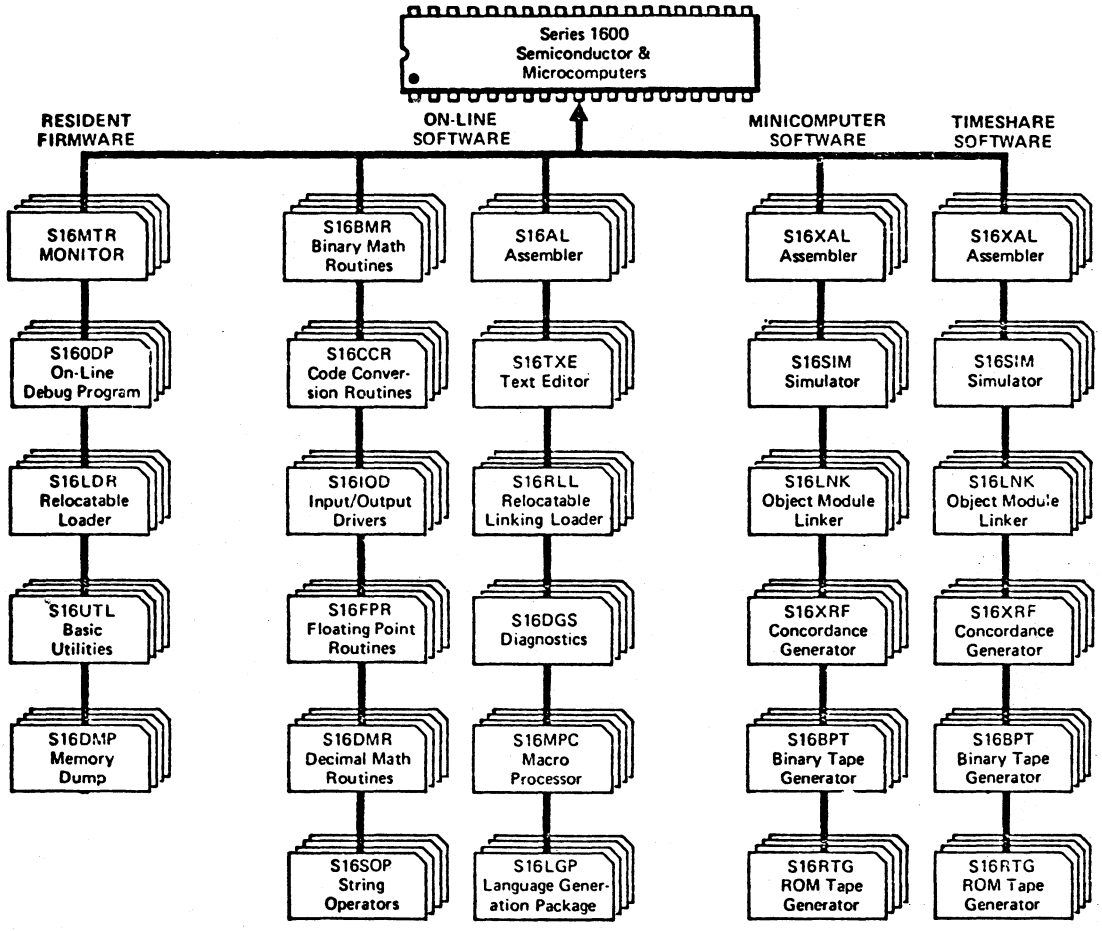
Software is fundamental to making every microprocessor come alive and the Series 1600 is no exception. The entire product family is supported by an extensive software system designed to make program development fast and efficient. Most important, the software structure is designed to grow with the hardware to insure long term product continuity.

The Series 1600 Cross Software Package contains a versatile set of program preparation tools including compatible Assembler/Simulator programs operating at two different computer system levels - large machine or time share, or popular minicomputer systems. Each accepts Series 1600 assembly language statements as input and produce relocatable, linkable object code as output. In addition, the full microprocessor environment, including I/O operations, is simulated on the host machine so that complete program debugging and testing can be performed before committing to hardware. The combination of these features along with the ability to use any minicomputer as a host processor results in the lowest cost, easiest to use, Cross Software Package in the industry.

The Series 1600 Microcomputer System (GIC 1600) also serves as a program preparation and hardware debug facility with the aid of its resident firmware and the On-Line Software Package. The resident firmware consists of a basic operating system containing a Monitor, the On-Line Debug Program, the Relocating Loader, the Memory Dump Program, and a number of other basic utility routines. The firmware also supports the system I/O with generalized routines for input/output from a TTY, high speed paper tape reader/punch, or any RS232 compatible device.

The On-Line Software Package includes the Symbolic Assembler, the Text Editor, the Relocating/Linking Loader, and the Macro Processor, with more to come. The next addition will be the Language Generation Package which enables high level language elements to be developed to match each application environment.

THE SERIES 1600 SOFTWARE LINE UP



GENERAL INSTRUMENT CORPORATION
MICROELECTRONICS

EASTERN AREA SALES HEADQUARTERS, 600 W. John St., Hicksville, N.Y. 11802, (516) 733-3107
CENTRAL AREA SALES HEADQUARTERS, 3101 West Pratt Blvd., Chicago, Ill. 60645, (312) 388-9200
WESTERN AREA SALES HEADQUARTERS, 7120 Havenhurst Ave., Van Nuys, Calif. 91406, (213) 781-0489

GENERAL INSTRUMENT CANADA LTD., 61 Industry St., Toronto 337, Ontario, Canada, Tel: (416) 763-4133
GENERAL INSTRUMENT MICROELECTRONICS LTD., 57/61 Mortimer St., London, W1N 7TD, England, Tel: 01-636-7022
GENERAL INSTRUMENT EUROPE S.P.A., Piazza Amendola 9, 20149 Milano, Italy, Tel: 469-7751
GENERAL INSTRUMENT FRANCE SA, 11-13 Rue Gandon, 75-Paris-13eme, France, Tel: 588-74-31
GENERAL INSTRUMENT DEUTSCHLAND GMBH, Neumarkter Strasse 61, 8 Munich 80, West Germany, Tel: 452239/450181
GENERAL INSTRUMENT INTERNATIONAL CORP., Fukide Building, 17 Fukide-cho, Minato-ku, Tokyo 105, Japan, Tel: (03) 437-0281-5
GENERAL INSTRUMENT OF TAIWAN LTD., P.O. Box 22226, Taipei, Taiwan, Tel: 933861-3

S16XSFT CROSS SOFTWARE PACKAGE

The Series 1600 Cross Software Package is coded in F level Fortran IV and is specifically designed to operate in a 16-bit minicomputer environment. Versions are available for many popular minicomputer systems such as DGC NOVA and DEC PDP11. The Cross Software package is easily installed on larger time-sharing computer systems.

S16XAL Symbolic Cross Assembler
- Symbolic representation of all instructions
- User defined six character symbols
- Octal, decimal, hexadecimal and ASCII literals
- Expression evaluation
- Extensive assembly directives
- Absolute, Relocatable or Relocatable/Linkable assembly
- Full program and sorted symbol listing
- Extensive error detection

S16XRF Concordance Generator
- Assembly symbol cross reference map

S16LNK Object Module Linker
- Resolves global/external symbol linkages
- Relocates and merges object modules
- Produces relocatable load module
- Produces load module map

S16SIM Simulator
- Full Series 1600 Instruction set simulation
- Full 65K word memory simulation
- I/O and interrupt simulation
- Memory and/or Register breakpoints
- Memory and/or Register traces
- Simulated program execution time accumulation
- Program execution time and stack size limits
- Inspection and modification of memory and registers
- Symbolic memory addressing

S16BPT Binary Paper Tape Generator

S16RTG ROM Pattern Tape Generator

SERIES 1600 ON-LINE SOFTWARE PACKAGE

The Series 1600 On-Line Software Package is written in assembly language and runs on any Series 1600 microprocessor. All programs are designed to be directly input/output compatible with the S16XSFT Cross Software Package so that either means of program preparation can be used interchangeably.

S16AL Symbolic Assembler
- Same features as S16XAL

S16TXE Text Editor
- Multiple line buffering
- Symbol search
- Character, line, string editing

S16RLL Relocating/Linking Loader
- Global and external symbol resolution
- Full relocation capability
- Loads and links multiple object modules
- Memory map

S16DGS Diagnostics
- Memory diagnostic
- Instruction test
- I/O Controller exerciser

S16MPC Macro Processor
- Unlimited definition capability
- System defined macros
- User defined macros

GENERAL INSTRUMENT CORPORATION
MICROELECTRONICS

EASTERN AREA SALES HEADQUARTERS, 600 W. John St., Hicksville, N.Y. 11802, (516) 733-3107
CENTRAL AREA SALES HEADQUARTERS, 3101 West Pratt Blvd., Chicago, Ill. 60645, (312) 388-9200
WESTERN AREA SALES HEADQUARTERS, 7120 Havenhurst Ave., Van Nuys, Calif. 91406, (213) 781-0489

GENERAL INSTRUMENT CANADA LTD., 61 Industry St., Toronto 337, Ontario, Canada, Tel: (416) 763-4133
GENERAL INSTRUMENT MICROELECTRONICS LTD., 57/61 Mortimer St., London, W1N 7TD, England, Tel: 01-636-7022
GENERAL INSTRUMENT EUROPE S.P.A., Piazza Amendola 9, 20149 Milano, Italy, Tel: 469-7751
GENERAL INSTRUMENT FRANCE SA, 11-13 Rue Gandon, 75-Paris-13eme, France, Tel: 588-74-31
GENERAL INSTRUMENT DEUTSCHLAND GMBH, Neumarkter Strasse 61, 8 Munich 80, West Germany, Tel: 452239/450181
GENERAL INSTRUMENT INTERNATIONAL CORP., Fukide Building, 17 Fukide-cho, Minato-ku, Tokyo 105, Japan, Tel: (03) 437-0281-5
GENERAL INSTRUMENT OF TAIWAN LTD., P.O. Box 22226, Taipei, Taiwan, Tel: 933861-3

GIC1600 RESIDENT FIRMWARE

The resident firmware in the GIC1600 Microcomputer System creates an efficient, easy to use, prototyping tool for the development of microprocessor based products. The firmware performs all front panel functions as well as creating a terminal driven operating environment. Features include the following:

S16MTR Monitor
- Conversational system control
- TTY communications

S16ODP On-Line Debug Program
- Eight program breakpoints
- Register/Memory display and modify
- Memory search and initialize
- Single step/Execute commands
- Modify Branch and Jump destinations
- Module Relocation Origins

S16LDR Relocating Loader
- Full relocation capability
- TTY or H.S. Paper Tape Reader input

S16MDP Memory Dump Program
- Punches in S16LDR format
- TTY or H.S. Paper Tape Punch output

S16UTL Utility Programs
- Generalized Code Conversions
- TTY input/output driver
- H.S. Paper Tape Reader/Punch driver

SERIES 1600 SUBROUTINE LIBRARY

The Series 1600 Microprocessor System is supported by an extensive and growing library of useful subroutines designed to relieve the user of many time consuming software chores. All of the Subroutine Library programs are written in Series 1600 Assembly Language making them both fast and efficient. They are compatible with both the Series 1600 Symbolic Cross Assembler (S16XAL) and the Series 1600 On-Line Assembler (S16AL). In addition, all library programs are designed to be directly compatible with hardware extensions to the Series 1600 product family so that increased performance can be achieved without software complications.

S16 BMR Binary Math Routines
- Signed Multiply/Divide
- Square Root
- Double Precision Multiply/Divide
- Double Precision Square Root
- SIN/COS Functions
- Double Precision Comparison
- Absolute Value

S16FPR Floating Point Routines
- Floating Add/Subtract
- Floating Multiply/Divide
- Square Root
- Logarithms
- Exponentiation
- Trigonemetric Functions

S16DMR Decimal Math Routines
- Decimal Add/Subtract
- Decimal Multiply/Divide
- Decimal Square Root
- Decimal Compare

S16CCR Code Conversion Routines
- Binary to BCD - BCD to Binary
- Binary to ASCII - ASCII to Binary
- Binary to HEX - HEX to Binary
- Binary to OCTAL - OCTAL to Binary
- Fixed to Floating - Floating to Fixed

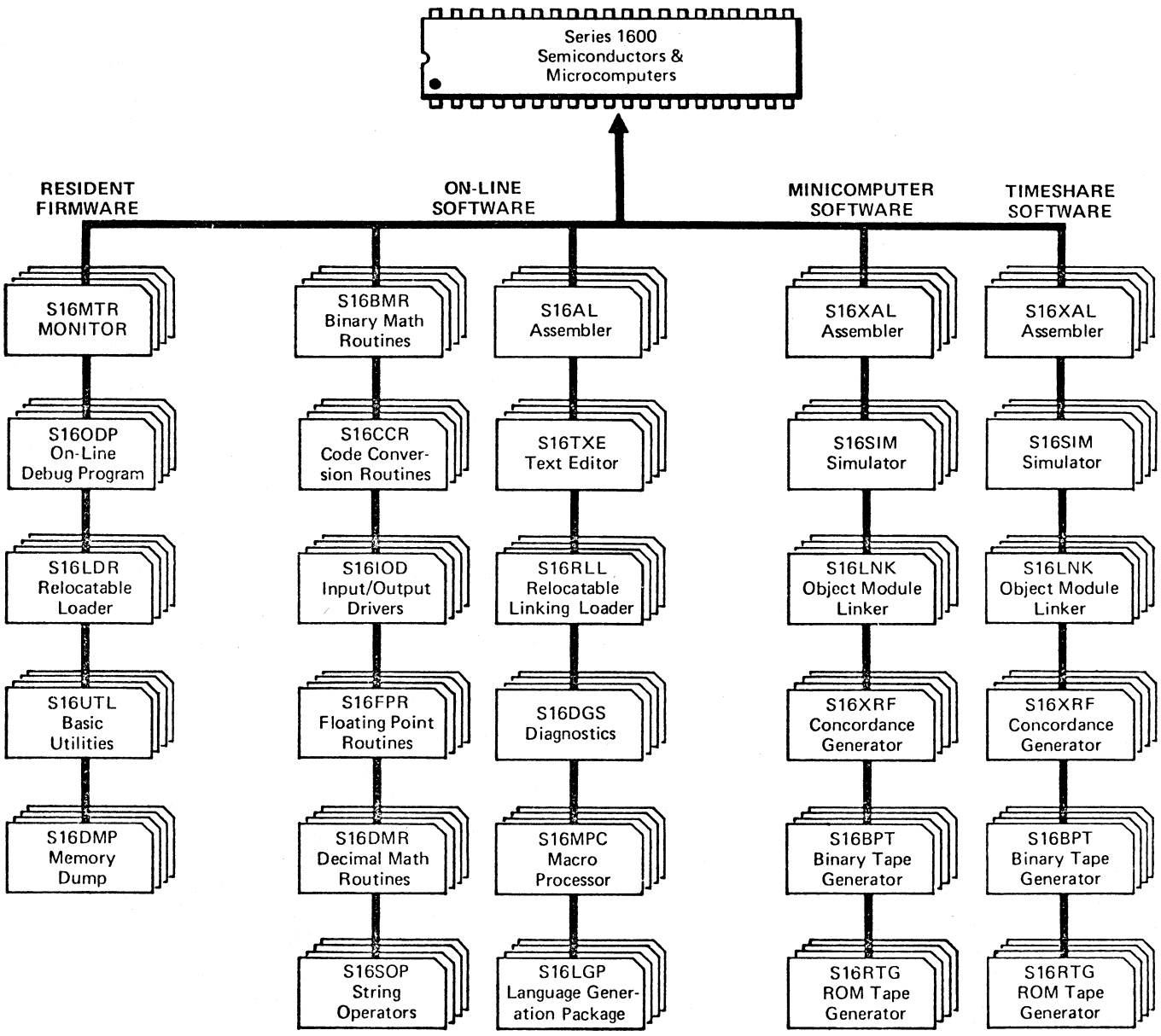
S16SOP String Operators
- Byte String Add/Subtract/Compare
- Byte String Move
- Byte String Search
- Word String Add/Subtract/Compare
- Word String Move
- Word String Search

S16IOD Input/Output Drivers
- TTY Input/Output
- H.S. Paper Tape Reader/Punch Input/Output
- Byte Table Pack - Byte Table Unpack

GENERAL INSTRUMENT CORPORATION
MICROELECTRONICS
EASTERN AREA SALES HEADQUARTERS, 600 W. John St., Hicksville, N.Y. 11802, (516) 733-3107
CENTRAL AREA SALES HEADQUARTERS, 3101 West Pratt Blvd., Chicago, Ill. 60645, (312) 388-9200
WESTERN AREA SALES HEADQUARTERS, 7120 Havenhurst Ave., Van Nuys, Calif. 91406, (213) 781-0489

GENERAL INSTRUMENT CANADA LTD., 61 Industry St., Toronto 337, Ontario, Canada, Tel: (416) 763-4133
GENERAL INSTRUMENT MICROELECTRONICS LTD., 57/61 Mortimer St., London, W1N 7TD, England, Tel: 01-636-7022
GENERAL INSTRUMENT EUROPE S.P.A., Piazza Amendola 9, 20149 Milano, Italy, Tel: 469-7751
GENERAL INSTRUMENT FRANCE SA, 11-13 Rue Gandon, 75-Paris-13eme, France, Tel: 588-74-31
GENERAL INSTRUMENT DEUTSCHLAND GMBH, Neumarkter Strasse 61, 8 Munich 80, West Germany, Tel: 452239/450181
GENERAL INSTRUMENT INTERNATIONAL CORP., Fukide Building, 17 Fukide-cho, Minato-ku, Tokyo 105, Japan, Tel: (03) 437-0281-5
GENERAL INSTRUMENT OF TAIWAN LTD., P.O. Box 22226, Taipei, Taiwan, Tel: 933861-3

THE SERIES 1600 SOFTWARE LINE UP



GENERAL INSTRUMENT CORPORATION
MICROELECTRONICS

EASTERN AREA SALES HEADQUARTERS, 600 W. John St., Hicksville, N.Y. 11802, (516) 733-3107
CENTRAL AREA SALES HEADQUARTERS, 3101 West Pratt Blvd., Chicago, Ill. 60645, (312) 338-9200
WESTERN AREA SALES HEADQUARTERS, 7120 Hayvenhurst Ave., Van Nuys, Calif. 91406, (213) 781-0489

GENERAL INSTRUMENT CANADA LTD., 61 Industry St., Toronto 337, Ontario, Canada, Tel: (416) 763-4133
GENERAL INSTRUMENT MICROELECTRONICS LTD., 57/61 Mortimer St., London, W1N 7TD, England, Tel: 01-636-2022
GENERAL INSTRUMENT EUROPE S.P.A., Piazza Amendola 9, 20149 Milano, Italy, Tel: 469-7751
GENERAL INSTRUMENT FRANCE SA, 11-13 Rue Gandon, 75-Paris-13eme, France, Tel: 588-74-31
GENERAL INSTRUMENT DEUTSCHLAND GMBH, (Mos Produkigruppe) Nordendstrasse, 1A 8000 Munchen 40 Tel: 28-40-31
GENERAL INSTRUMENT INTERNATIONAL CORP., Fukide Building, 17 Fukide-cho, Minato-ku, Tokyo 105, Japan, Tel: (03) 437-0281-5
GENERAL INSTRUMENT OF TAIWAN LTD., P.O. Box 22226. Taipei. Taiwan, Tel: 933861-3

Printed in U.S.A.