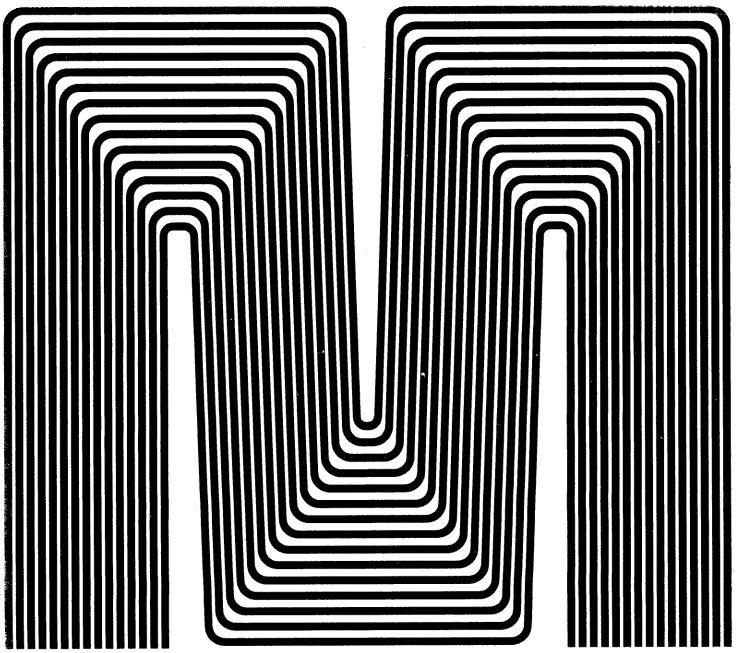
Microdata

MICRO 1600 COMPUTER

REFERENCE MANUAL

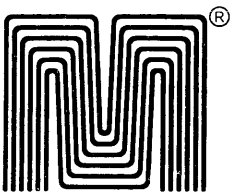


MICRO 1600 COMPUTER

REFERENCE MANUAL

71-1-1600-001
October 1971

Microdata®



Microdata Corporation
644 East Young Street
Santa Ana, California 92705

Printed in USA

TABLE OF CONTENTS

CHAPTER 1

MICRO 1600 DESIGN FEATURES

INTRODUCTION

The MICRO 1600 is a microprogrammable digital computer with the capability to satisfy a broad range of application requirements through the use of both expandable high-speed control memory and magnetic core main memory.

The high-speed control memory continuously sequences preprogrammed microcommands which generate control and timing signals to perform all control operations and data manipulations in the computer.

Using application programming at the micro level, the MICRO 1600 can be used directly as a hard-wired controller. When the 1600 emulates the operation of a general purpose computer which executes software instructions stored in core memory, macroinstructions are fetched and interpreted by the microprogram with corresponding operations carried out by execution of microprogrammed routines in the control memory.

Basic macroinstruction sets are available which are significantly more powerful than conventional minicomputers. Individual users can add to or modify the basic macroinstruction set and basic input/output structure by a simple addition or change of firmware in the control memory to increase the flexibility and utility of the machine.

Control memory can be implemented in bipolar read-only memory (BROM), programmable read-only memory (PROM) or alterable read-only memory (AROM) devices. BROM is low cost and is appropriate for volume production of field-proven firmware. PROM permits microprograms to be installed at the factory or in the field with fast turn-around time and low initial set-up costs. It is intended for use in low production volume. AROM permits dynamic microprogramming and/or the debugging of firmware in a real-time environment before implementation into the more permanent BROM and PROM control memories.

There are 30 general-purpose file registers which are implemented with MSI/LSI semiconductor devices. Under program control, these registers are typically assigned functions such as I/O buffer registers, accumulators, index registers, and program counters.

The internal data paths and I/O are byte-oriented with eight-bit word lengths. Under control of microcommands, effective word lengths are variable.

Owing to its inherent flexibility, the MICRO 1600 can be applied as a direct function processor, general-purpose computer, special-purpose computer, emulator or language processor.

The MICRO 1600's control memory can be expanded to 16,384 16-bit words. This will permit implementation of languages such as BASIC, COBOL, FORTRAN, or equivalent complex firmware requirements. Direct language implementation will eliminate intermediate compile operations and results in an interpretive processor or a compile-and-go capability with performance exceeding conventional minicomputers.

Packaging variations permit operations ranging from a stripped-down low-cost minicomputer with three printed circuit boards and a card cage to a "super" computer with multiprocessor capability.

GENERAL CHARACTERISTICS

Advanced features and operating characteristics include:

- 65,536 bytes of memory in basic enclosure
- 4096 and 8192 byte core memory modules
- 1 microsecond main core memory speed (full cycle)
- Dual processors with common main memory
- Macro processors

  Standard macro processors, the 1600/10, 1600/20 and 1600/21 are available. These permit users to apply the machine using conventional software programming while providing advanced system features.

- Alterable read-only memory

  Alterable read-only memory permits firmware programmers to operate new microprograms in a true on-line environment.

- Supporting standard software including special firmware development packages.

  AP1600 – Micro language cross-assembler written in FORTRAN

  MAP1600 – Micro language assembler written for use on the MICRO 1600/20 and 1600/21 computers.

  SIM1600 – MICRO 1600 simulator written for use on the MICRO 1600/20 and 1600/21 computers.

  ICM1600 – Integrated circuit memory MAP generator permits direct conversion of AP/MAP1600 outputs to control memory bit patterns.

  ROM Diagnostics – Pluggable standard CPU diagnostics in read-only memory.

  AROS1600 – Alterable read-only memory operating system for control of AROM used for firmware checkout and debug.

- Direct memory access (DMA)
- 30 general purpose eight-bit file registers plus eight-bit status register
- Up to 16,384 words of read-only memory in 256-word modules
- 200-nanosecond microcommand execution time
- Real-time clock (optional)
- Standard automatic shutdown in event of power failure and automatic startup when power returns
- Operating temperature range 0 to 50°C
- Dimensions: 10-1/2 inches high, 19 inches wide, 20 inches deep
- Power: 115/230 VAC, 47-63 Hz, 350 watts
- Optional 16-level stack for recursive firmware

CHAPTER 2

SYSTEM DESCRIPTION

The MICRO 1600 is an eight-bit computer employing microprogram control. The computer incorporates eight-bit registers and data paths, executing with every clock pulse a 16-bit microcommand stored in a high-speed semiconductor control memory. The major elements of the system are shown in Figure 1.

This chapter describes the registers, data flow, memory and control. The microcommands are described in Chapter 3 and the input/output in Chapter 4.

REGISTERS

All registers except the file registers have specific functions in the machine. The file registers are used by the programmer for general purpose use. Description of the processor's registers follow.

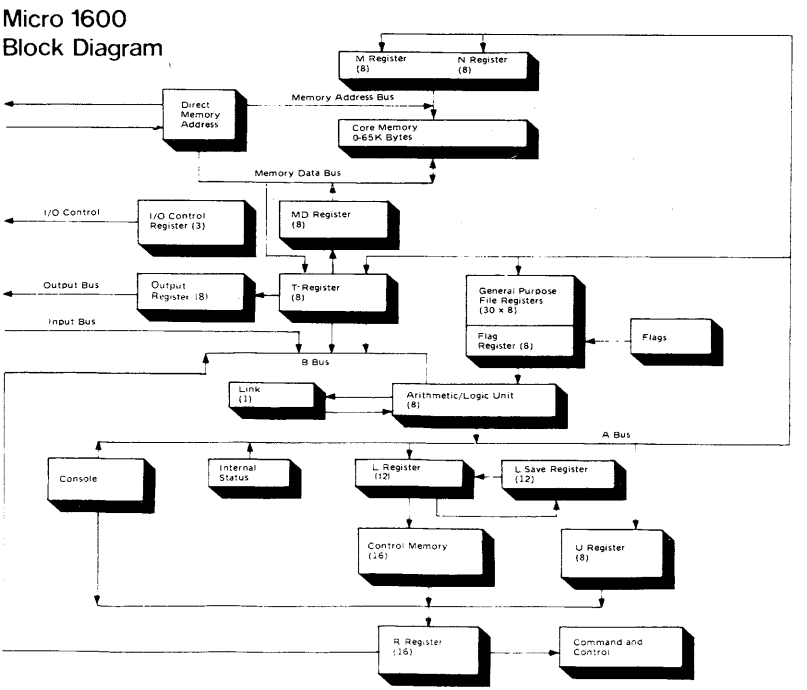


Figure 1. MICRO 1600 Block Diagram

T Register

The eight-bit T Register serves as an operand register for most operate commands and a buffer register for data being written into the memory and output on the byte I/O bus. The contents of the T Register are transferred to the MD or OD Registers on memory write or output respectively, thus freeing the T Register for other uses. Memory read operations cause the accessed data to be placed in the T Register 400 nanoseconds after the memory read is initiated.

M Register

The eight-bit M Register holds the 8 high-order bits of the processor's 16-bit memory address.

N Register

The eight-bit N Register holds the 8 low-order bits of the processor's 16-bit memory address.

U Register

The eight-bit U Register is used to modify the 8 high-order bits of the control memory output. The contents of the U Register are ORed with the Control Memory output on the R-Bus as it is gated into the R Register for those commands which have zeros in bits 15-12 or zeros in bits 2-0 and a 1 in bit 15. The U Register permits efficient use of control memory by allowing a common routine to be used for different operations, when the operations differ by only a few commands.

File Registers

Two files of 16 registers each provide storage for internal flags, and user's data. Typical assignments include program counters, accumulators, index registers, temporary buffers, etc. The primary or secondary file is selected by command and the primary file is selected after reset or power-on.

Only registers 1-15 are available to the user. Register 0 is common to both files and contains flags as described later. This register cannot be written into but can be used as a source of data. Readout of register 0 does not alter its contents.

Link Register

The two-bit LINK Register holds the carryout of the high-order bit position of the adder for Add, Increment, Subtract, Decrement and Compare commands and the shifted off end bit for Shift commands. The ML link bit is selected and/or set when the M or N Registers have been selected as a destination, otherwise the AL link bit is used.

IC Register

The three-bit IC Register is the I/O Control Register which specifies the I/O bus control signal to be enabled. The output of this register is decoded within I/O device controllers into three output control signals and four input control signals. The register is loaded and cleared by micro-commands, therefore the timing of control signals on the bus is up to the microprogram. When the IC Register contains a value of 4-7 one of the

input modes is specified and the Input bus is substituted for the T Register on any commands which select T or the complement of the T.

L Register

The 12-bit L Register holds the address of the next control word in sequence and provides for direct addressing of 4096 words of control memory. The register is incremented by one as each instruction is executed unless it is loaded by a new value which will effect a jump in the sequence. There are no restrictions in incrementing the L Register. The register can be loaded by a Jump command which alters the 10 low-order bits, a Jump Extended command which alters all 12 bits, a Return command which alters all 12 bits, or by specifying the L Register as the destination which alters the 9 low-order bits.

When the extended control memory option is included in the machine a Bank Select Register allows addressing to 32K. In this case the 3 high-order bits are set by the Bank Select command.

L Save Register

The 12-bit L Save Register saves the incremented contents of the L Register when a Jump Extended command is executed, unless this has been inhibited by prior execution of an Inhibit L Save command. Execution of a Return command causes the contents of the L Save Register to be transferred back to the L Register. In this manner the L Save Register acts as a linkage register allowing one level of subroutine.

When the L Save stack option is included in the machine the L Save Register is replaced by an L Save Stack of 16 12-bit registers which perform the same function. Selection of the current stack register is under command control.

R Register

The 16-bit R Register holds the microcommand currently being executed. The input to the R Register is from the R-Bus which is normally the output of the control memory, but may also be the console switches.

MD Register

This eight-bit register is a buffer which holds data being written into memory. It is not directly accessible to the programmer, but automatically copies the contents of the T Register 350 nanoseconds after the write operation is initiated. This frees the T Register for other uses.

OD Register

This eight-bit register is a buffer which holds data being output on the Byte I/O Bus. It is not directly accessible to the programmer, but automatically copies the contents of the T Register when the IC Register is set to a non-zero value. This frees the T Register for other uses.

DATA FLOW

Arithmetic/Logic Unit

The arithmetic/logic unit (ALU) is the heart of the processor's data flow. All transfers and manipulation of data are done through this unit. The operations that it performs include: add, subtract, AND, OR, exclusive-OR, transfer and shifting. The two inputs to the ALU are the selected file register and the operand placed on the B Bus.

A and B Buses

The output of the ALU is placed on the A Bus where it can be routed to all the processor's registers. Other inputs to the A Bus are the internal status byte, the four sense switches on the front panel and a selected file register shifted four bits to the right. The information on the A Bus is routed to a register implied by the operation code of the command, or optionally to the selected file register and a designated register.

The B Bus is the second input to the ALU. Sources of data on the B Bus are the literal in the eight low-order bits of the command and the true or complement of the contents of the T Register or the data on the Input Bus. The Input Bus is substituted for the T Register as a selected source when the I/O control IC Register is in an input mode. After a switchable delay after placing the I/O control in an input mode, data from an I/O unit will be available on the Input Bus.

Arithmetic and Memory Link Bits

Two link bits are provided in a register commonly referred to as LINK. These bits store the carry out of the high-order bit position of the ALU on arithmetic and compare commands, and the shifted off end bit in shift commands. The Memory Link (ML) is used when the M or N Registers are selected as the destination register; otherwise the Arithmetic Link (AL) is used. The two link bits permit intermixed memory address and data arithmetic operations.

T Register

The T Register is a transient working register used to hold operands for the ALU, receive data from memory, write data to memory and output data to external units. The T Register is backed up by the MD and OD Registers which buffer data being written in memory and output respectively. The purpose of these two buffers is to free the T Register for other operations during the relatively lengthly write and output operations. The T Register, and sometimes its complement constitute selectable operands on the B Bus. If the T Register is not selected in those commands which have that option the operand on the B Bus will be zero. If both the true and complement of the contents of the T Register are selected, the operand is all 1's. Data read from memory is set into the T Register two clock intervals after the read is initiated. Commands which select T Register as the source of operand during the first two clocks of the read operation are delayed until the third clock.

REGISTER 0 FLAGS AND INTERNAL STATUS

Register 0 Flags

Register 0 which is common to both the primary and secondary files contains a set of flags which reflect the result of a previous operation and external conditions which require frequent testing. The flags in bits 0-2 are the result condition flags and are updated when the C modifier (bit 4) of the operate instructions is a one. A description of the Register 0 flags follows:

0 – Overflow Condition: The overflow condition flag stores the overflow condition of an Add, Increment, Subtract, Decrement, or Copy command. Arithmetic overflow occurs when carry out of the high-order bit of the adder differs from the carry into it. The over-flow also stores the shifted off end bit of shift commands.

1 – Negative Condition: The negative condition flag stores the high-order bit of the result. When overflow occurs this flag will be the complement of the true sign.

2 – Zero Condition: The zero condition flag stores the zero condition of the result. The zero test can be linked over multiple byte operations under control of the L modifier (bit 7) of operate instructions. When this bit is 1, the zero condition flag may not be set to indicate the zero condition of the current byte, but may only be reset to indicate a non-zero result. For this flag to indicate zero over multiple bytes it must be set by a zero result on the first operation which will have the L modifier zero and not be reset by non-zero conditions on succeeding bytes which will have the L modifier a one.

3 – I/O Request: The I/O request flag is turned on by one or more external I/O units requesting an I/O operation.

4 – Internal Interrupt: The internal interrupt flag is turned on when an internal interrupt condition is present. The internal interrupt is identified in the internal status (Table 1).

Table 1. Internal Status

Bit	Status Meaning
0	Panel Interrupt
1	DMA Termination
2	Real Time Clock Interrupt
3	Spare
4	Spare
5	Spare
6	Panel Step Switch
7	Power fail (Restart Interrupt)

5 – I/O Reply: The I/O reply flag is turned on by the external I/O unit currently communicating with the processor. This flag is normally not used in MICRO 1600 I/O units.

6 – Serial TTY or Stack Overflow: The serial teletype input flag indicates the state of the serial teletype input. A zero indicates that the input is in a MARK state. This flag is used for MICRO 800 compatibility. The stack overflow flag is turned on when the L Save Stack has overflowed.

7 – External Interrupt: The external interrupt flag is turned on by one or more external I/O units requesting an interrupt. This flag must result in a command which reads the address of the interrupt vector and resets the request.

Internal Status

The internal status reflects the state of the internal interrupts. When any of these functions is requesting an interrupt, bit 4 of the register 0 flag is turned on. Most of the internal interrupt status signals are turned off when the status is read. The status can be read by an Enter Internal Status command. The assignment of the internal status bits is as shown in Table 1.

CONTROL

The sequential control of the processor is obtained from a microprogram stored in a 16-bit control memory. Each word accessed from the control memory is placed in the R Register where it directly controls selection of operands, ALU function, register enables and the determination of the next control memory location. In effect the contents of control memory control the action of the processor at each clock. Because the Micro 1600 is a microprogrammed type of computer there is a minimal amount of command decoding and each command is executed in a single clock period, unless there is a delay imposed by the memory.

Control Memory

The control memory is a 16-bit high-speed memory implemented with semiconductor read-only memory (ROM) devices, or read-write memory providing an alterable control memory (ACM). The standard MICRO 1600 can be expanded to 4096 words of control memory. Control memory can be added in 256 word pages. An option allows expansion to 32,768 words of control. The control memory can be randomly accessed with an access time, including logic delays of less than 200 nanoseconds.

The execution of commands and the accessing of the control memory are overlapped. While one command is in the R Register being executed the next command in sequence is being accessed and the L Register has been incremented to the address of the command being accessed. When the normal sequence is altered by a jump a delay of 200 nanoseconds takes place to allow reading of the first command at the start of a new sequence.

Two types of ROM devices are used for control memory. The first has the ones and zeros pattern of the microprogram built into the device by a special masking used in the manufacturing process. These devices are used for standard Microdata firmware. The other type is a similar ROM device, but can be programmed after device manufacturing by selectively burning out fuses corresponding to the bits of the desired program. This type of device is commonly called a programmable read-only memory (PROM) and costs considerably more than the masked ROM. Each board of control memory can accommodate up to 2048 words in 256-word increments.

The ACM provides for a dynamically alterable control memory. This type of memory is very useful when debugging firmware. The memory initially is treated as an external unit on the I/O bus when it is loaded and then treated as an extension of the control memory in the processor. ACM modules may contain up to 1024 words. Use of ACM modules in the system require additional cooling and +5-volt power supply current. Each 256 words of ACM requires approximately 2.0 ampere.

L Save Stack and Extended Addressing Option

The L Save Register can be expanded to 16 15-bit L Save Registers by means of a stack option permitting multilevel subroutines. When this option is included the L Save Register in the processor is disabled. Along with this feature is a 3-bit Bank Select Register permitting the full 32,768 words of control memory to be addressed. The 3-bits of the Bank Select Register are controlled by the Bank Select command which must be followed by a Jump Extended. Associated with the L Save Stack registers is a 4-bit stack pointer which points to the next available stack register. Incrementing and decrementing of this pointer is done by command. It is reset to register zero on power-up and with the panel reset switch. This stack and extended addressing option is included with a 2048-word control memory module. Only one stack may be used in a system.

Timing

The processor is controlled by a 10 MHz crystal oscillator which is divided by two to produce a 5 MHz clock. This frequency may be reduced if desired. However, all timing will be delayed including timing delays which may be coded in the program.

MEMORY

Memory Interface

The memory modules receive their address over a set of 16 address lines on the printed circuit backplane. This address is derived from the M and N Registers of the processor or the Direct Memory Access (DMA) option. A set of eight data lines on the backplane provides for transfer of data between the memory and processor or between the memory and DMA. Data read from the memory under control of the processor is transferred to the processor's T Register which is cleared one clock time after the start of the memory read operation. Data to be written into the memory is placed on the data lines by the processor's MD Register which receives its input from the T Register.

Memory Modules

The standard memory is a 4096 or 8192-byte core memory module built on a single printed circuit board. The memories make use of lithium cores for operation over a wide temperature range. The memories may be operated in either a full or half cycle mode. Full cycle operation provides for a read-restore or clear-write type of operation with the program initiating only one operation. The half cycle operation allows the read and write operations to be programmed separately. This is normally used to effect a read-modify-write type of operation. After performing a half cycle read the addressed location is left in an all ones state.

Memory Addressing

Addressing is available to a maximum of 65,536 bytes. Normally the high-order address line is disabled and forced to zero limiting addressing to 32,768 bytes. This is done because standard Microdata macro computer configurations use the high-order bit of address words for indexing control, leaving only 15 bits for address. Full 16 bits of address may be used for special applications. Core memory modules of different sizes may be intermixed within a system. However, only one 4196-byte module can be used in conjunction with one or more 8192-byte modules.

Memory Timing

The memory cycle time is 1.0 microseconds with an access time of 400 nanoseconds until data is in the T Register. The half cycle time is 600 nanoseconds with an access time of 400 nanoseconds. Execution of commands which specify that the M or N Register are to receive data is delayed if the memory is busy. These commands are executed on the last clock of the cycle, i.e., the fifth clock on a full cycle operation and the third clock on a half cycle operation. Operate commands which select the contents of the T Register are not executed at the first or second clock following the initiation of a read operation. This allows the accessed data to be placed in the T Register before the command is executed. Two exceptions to this imposed delay occur when either an OR, Exclusive OR, or AND command selects both the T Register and its complement or when a command selects either the T Register or its complement with the input buss enabled. For these operations the command will execute in one machine cycle and the memory data will be loaded into the T Register as previously specified.

Direct Memory Access (DMA)

A Direct Memory Access (DMA) channel allows external devices to directly communicate with memory at data transfer rates up to 1 million bytes/second.

POWER FAIL – AUTO RESTART

Standard with the MICRO 1600 is a feature which provides for detection of loss of AC power and an orderly startup when power is turned on. Both power fail and restart are indicated by a 1 in bit 7 of the internal status, which in turn turns on the Internal Interrupt flag of file register 0.

Power Fail

The standard power supplies are equipped with circuitry for detecting low AC line voltage. When low AC line is detected the Internal Status bit 7 is turned on. The microprogram must periodically test for internal interrupts. After the low line is detected the power supply will hold all voltages within operating range for a minimum of one millisecond. After sensing the power fail interrupt the microprogram should save all volatile registers containing valid data in core memory and bring the processor to a halt. When the processor halts a reset is applied to the system to provide for an orderly loss of power.

Restart

When power is applied a reset is applied to the system until the power supply voltages stabilize at their operating values. This reset initializes the system, sets the L Register to zero, sets internal status bit 7 and prevents accidental operation of the core memory. The microprogram can distinguish the restart from the power fail by testing internal status bit 7 soon after starting execution when the reset is removed. This test should not be used later when there is a chance that the status bit will reflect power fail.

CHAPTER 3

MICROCOMMAND REPERTOIRE

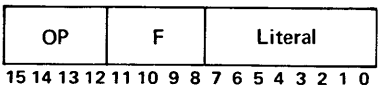
This chapter contains a description of all MICRO 1600 microcommands. With each description is a diagram showing the format of the command and its operation code given in hexadecimal. Above the diagram is the command's mnemonic and the name of the command. Under each diagram is a description of the command, followed by a list of the registers and indicators that can be affected by the command. The timing of each command is 200 nanoseconds except as noted, or if memory timing delays described in Chapter 2 are encountered.

COMMAND FORMATS

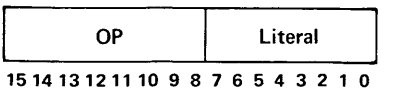
There are five basic command formats. Each command is 16 bits in length and is stored in a single control memory location.

Literal Commands

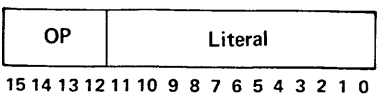
The literal classes of commands have the following formats:



The operation code occupies the four high-order bits. Bits 11-8 contain the file register designator. Bits 7-0 contain an eight-bit literal which is transferred as an operand.



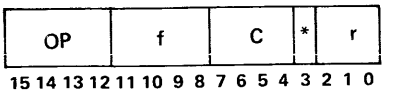
The operation code occupies the eight high-order bits. Bits 7-0 contain an eight-bit literal which is transferred as an operand.



The operation code occupies the four high-order bits. Bits 11-0 contain a 12-bit literal which is transferred as a control memory address. (This format is used for Jump Extended only.)

Operate Commands

The operate class of commands have the following format:



The operation code occupies the four high-order bits. Bits 11-8 contain the file register designator (f). Bits 7-4 contain the control (C) field designator.

C-Field Designators

C-Field Bit Position	Designator	Definition
1000	L	Link control/Add Link
0001	C	Modify Cond. Codes
0010	T	Select T
0100	F	Select T Complement
0100	I	Add 1/Increment
0100	D	Decrement

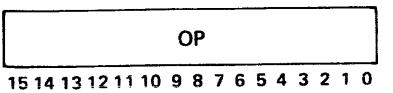
Bit 3 is file inhibit. When bit 3 is set to a one the resultant operation of the command is inhibited from being transferred to the designated file. Symbolically, this is specified to the assembler programs by appending an * to the command mnemonic. The destination register (r) is specified in the three low-order bits, 2-0. When the designator is L or K the command requires 400 nanoseconds to execute.

r Field Destination Register Designators

Bit Configuration	Designator	Register Designated
000		None
001	T	T Register
010	M	M Register
011	N	N Register
100	L	L Register (adds 200 nanoseconds)
101	K	K Register (adds 200 nanoseconds)
110	U	U Register
111	S	U Register ORed into upper 8 bits of commands with OP codes 8 through F.

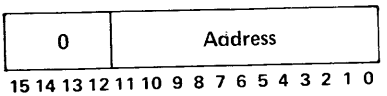
Generic Commands

The generic class of commands have the following Formats:



The operation code occupies all 16 bits of the microcommand.

JE JUMP EXTENDED

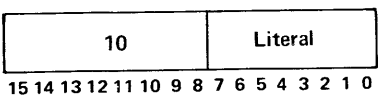


The contents of the 12-bit address field are placed in the L Register. If an inhibit L Register Save command has not been executed since the last Jump Extended, the following operation will also take place. The old contents of the L Register are stored in the L Save Register, or when the L Save Stack option is present, the L Register is stored in the 12 low-order bits of the current stack level, and the contents of the control memory bank Select Register is stored in the upper three bits of the current stack level. A Jump Extended command requires 400 nanoseconds to execute.

This command permits jumping anywhere within 4096 words of control memory in the basic machine. When the L Save Stack and extended control memory option is included, this command controls jumping anywhere within the current selected 4096-word bank of control memory. To jump from one bank of control memory to another, a Bank Select command (BSL) is executed followed by a Jump Extended command to the desired location in the selected 4096-word bank of control memory.

The Jump Extended command is assigned an operation code of 0 and is a special form of the Execute command which OR's the contents of the U Register into the eight high-order bits of the command. In order to obtain an operation code of 0 in the R Register, the four high-order bits of the U Register must be set to zero. If desired the four low-order bits of the U Register may be used to set bits 8-11 of the L Register; otherwise they should also be set to zero.

LZ LOAD ZERO CONTROL



The load zero control command's eight-bit literal field is used to specify combined operations of the load zero group of generic instructions. If the literal field is 00 no operation (NOP) occurs. When multiple bits are specified, the designated control functions will execute.

This single command in a "vertical" sequence of microprogramming, contains "horizontal" microprogramming characteristics with the ability to perform multiple operations in a single 200 nanosecond clock interval.

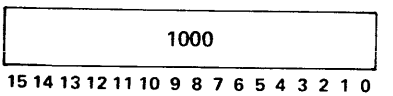
Load Zero Literal Field

Bit Configuration	Code	Mnemonic – Operation
0000 0000	00	NOP – No Operation
0000 0001	01	ECR – Enable Communication Rate Generators
0000 0010	02	DCR – Disable Communication Rate Generators
0000 0100	04	ICR – Input Communication Rate Generators
0000 1000	08	Unassigned
0001 0000	10	Unassigned
0010 0000	20	RTN – Return
0100 0000	40	SPF – Select Primary Files
1000 0000	80	SSF – Select Secondary Files

Example: LZ 62 combines the three operations – DCR (disable rates), RTN (return) and SPF (SELECT primary files).

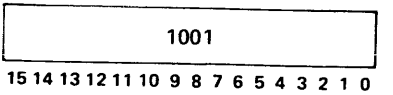
Note: If two commands with opposite functions are executed the existing state is changed to the opposite. Example: LZ 03 defines both enable and disable communication rate generators. When this command is executed the status of the communication rate generator control is complemented. If it was on, it is turned off. If it was off, it is turned on.

NOP **NO OPERATION**



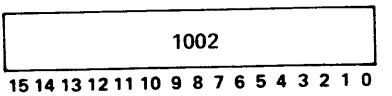
This command performs no operation and can be used to insert a delay of 200 nanoseconds.

ECR **ENABLE COMMUNICATION RATE GENERATORS**



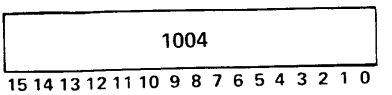
The optional communication rate generators are enabled. These four hardware strappable time interval counters may be used for input bit sampling and output bit updating by microprograms performing serial to parallel and parallel to serial conversions.

DCR DISABLE COMMUNICATION RATE GENERATORS



The optional communication rate generators are disabled.

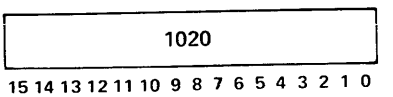
ICR INPUT COMMUNICATION RATE GENERATORS



If the communication rate generators are enabled, and one or more rate is requesting service; bit 5 of the internal status byte will be set to a one and file zero, bit 3 will be set to a 1 indicating the presence of an internal interrupt.

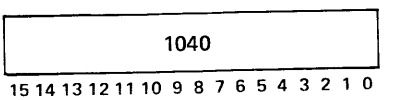
This command allows the four communication rate generator flags to be read and reset by following it immediately with an Enter Internal Status (EIS) command. Bits 5-2 of this status byte will contain the current state of the four rate generators, and bits 7, 6, 1, and 0 should be ignored. The state of the internal status byte is not affected by this two command sequence.

RTN RETURN



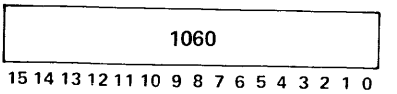
The contents of the L Save Register are placed in the L Register or when the L Save Stack option is present, the low-order 12 bits of the current stack level are placed in the L Register and the high-order three bits of the current stack level are placed in the control memory bank select register. Execution requires 600 nanoseconds and the Inhibit L Save mode is cleared, if set, causing further Jump Extended commands to save the current contents of the L Register.

SPF SELECT PRIMARY FILE



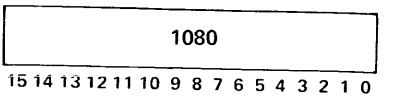
This command causes the primary file of registers to be selected for further file register operations. The primary file is also selected after a power on or by pressing master reset on the front panel.

RSP **RETURN AND SELECT PRIMARY FILE**



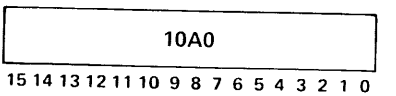
This command combines the functions of Return (RTN), and Select Primary File (SPF), and executes in 600 nanoseconds.

SSF **SELECT SECONDARY FILE**



This command causes the secondary file of registers to be selected for further file register operations. This set will remain selected until execution of a Select Primary File command, or the occurrence of a master reset.

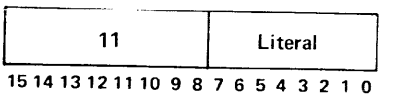
RSS **RETURN AND SELECT SECONDARY FILE**



This command combines the functions of Return (RTN), and Select Secondary File (SSF), and executes in 600 nanoseconds.

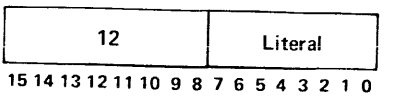
The following commands provide for loading eight-bit literals into registers.

LT **LOAD T**



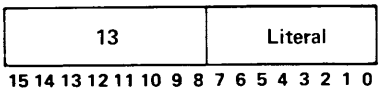
The contents of the eight-bit literal field are placed in the T Register. The condition flags and LINK are not affected.

LM **LOAD M**



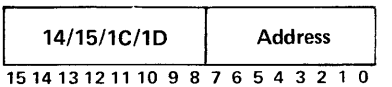
The contents of the eight-bit literal field are placed in the M Register at a time when the memory is not busy. The condition flags and LINK are not affected.

LN LOAD N



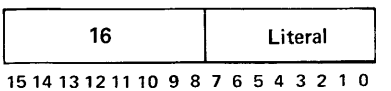
The contents of the eight-bit literal field are placed in the N Register and the M Register is cleared at a time when the memory is not busy. The condition flags and LINK are not affected.

JP JUMP



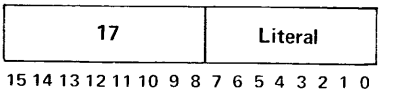
The contents of the eight-bit address field are placed in the eight low-order bits of the L Register, bits 8 and 11 of the command are placed in bits 8 and 9 of the L Register respectively. Bits 10 through 12 of the L Register and the optional control memory bank select register remain unchanged. The location of the next command is the address specified by the new contents of the L Register. These jump commands provide for jumping within a four-page block of 1024 words whose starting address is zero modulo 1024. The assembler program selects the proper command code from the address which must be in the 1024-word block containing the command. The command executes in 400 nanoseconds.

LU LOAD U



The contents of the eight-bit literal field are placed in the U Register. The condition flags and LINK are not affected. Due to look-a-head access method of the control memory, the new contents of the U Register are not available for command modification during the machine cycle immediately following the Load U command.

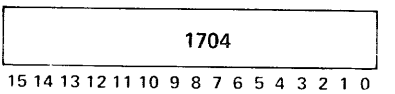
LS LOAD SEVEN CONTROL



The individual bits of the literal field control independent functions. Any number of bits in the literal field may be one's. A number of the control functions are given mnemonics and are described later in this section. The function of each bit is as follows:

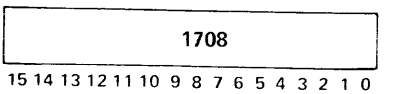
Bit Position	Function
0	(unassigned and unavailable)
1	(unassigned and unavailable)
2	Disable External Interrupts
3	Enable External Interrupts
4	Disable Real Time Clock
5	Enable Real Time Clock
6	(unassigned and unavailable)
7	Halt Processor

DEI DISABLE EXTERNAL INTERRUPTS



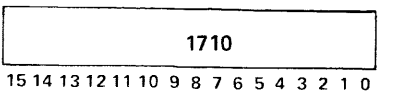
This command causes the external interrupt system to be disabled. Interrupts are not lost when the interrupt system is disabled, but cannot be recognized by the processor.

EEI ENABLE EXTERNAL INTERRUPTS



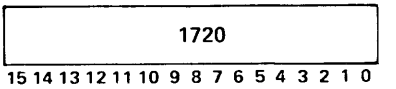
The external interrupt system is enabled allowing the processor to recognize external interrupts.

DRT DISABLE REAL-TIME CLOCK



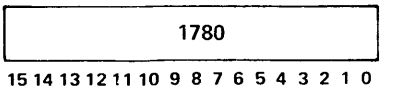
The real-time clock and interrupt are disabled.

ERT ENABLE REAL-TIME CLOCK



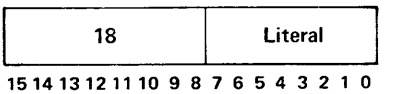
The real-time clock and interrupt are enabled. The first interrupt will occur after a full interrupt interval.

HLT HALT



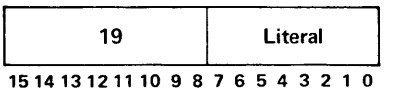
The processor is halted stopping all microcommand execution. However, the Direct Memory Access channel, if activated, will continue its operation until completion.

LE LOAD EIGHT CONTROL



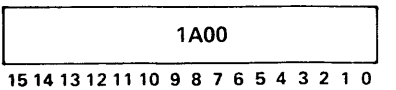
This command is not implemented in the standard MICRO 1600. It can be defined for special user applications employing techniques similar to those used in Load Zero Control Group and Load Seven Control Group.

RLT RETURN AND LOAD T



This command combines the functions of Return (RTN), and Load T (LT) and executes in 600 nanoseconds.

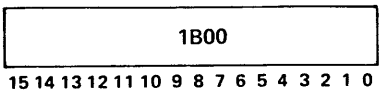
MLC MODIFY LOWER COMMAND



This command causes the contents of the Output Data Buffer to be ANDed with 8 low-order bits of the next command accessed from the control memory. If the IC Register is set to 0, the contents of the Output

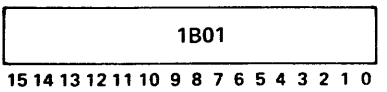
Data Buffer will be the same as the T Register. If the IC Register is not set to 0, the Output Data Buffer will contain the value that was in the T Register at the time the IC Register was set non-zero.

ILS INHIBIT L SAVE



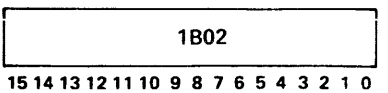
This command inhibits the saving of the L Register in the L Save Register or the Bank Select Register and the L Register in the current level of the L Save stack until execution of a Return command.

ISP INCREMENT STACK POINTER (optional)



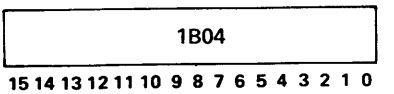
The contents of the four-bit stack pointer register are incremented by one to select the next level in the stack for saving the contents of the L Register during the next Jump Extended. If the modified contents of the register are greater than 15, bit 6 in file 0 will be set; otherwise it will be reset.

DSP DECREMENT STACK POINTER (optional)



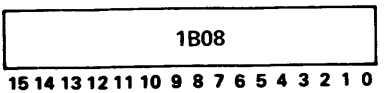
The contents of the four-bit stack pointer register are decremented by one to select the previous level in the stack for a return operation. If the modified contents of the register are less than zero, bit 6 in file 0 will be set; otherwise it will be reset.

CSP CLEAR STACK POINTER (optional)



The contents of the four bit stack pointer register are set to zero.

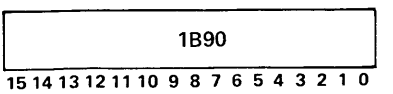
BSL BANK SELECT (optional)



The bank select register is set to zero selecting the first 4096-word bank of control memory. The actual selection of the new bank takes place when the next Jump Extended command is executed. As shown below, variations of this command are used to select up to eight individual banks.

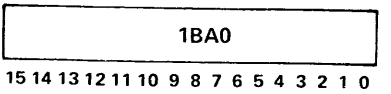
BSL 0	1B08	Select bank 0	(0-4095)
BSL 1	1B 09	Select bank 1	(4096-8191)
BSL 2	1B 0A	Select bank 2	(8192-12,287)
BSL 3	1B 0B	Select bank 3	(12,288-16,383)
BSL 4	1B 0C	Select bank 4	(16,384-20,479)
BSL 5	1B 0D	Select bank 5	(20,480-24,575)
BSL 6	1B 0E	Select bank 6	(24,576-28,671)
BSL 7	1B 0F	Select bank 7	(28,672-32,767)

SSP SELECT STACK POINTER (optional)



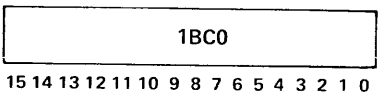
This command causes loading and reading of the four-bit Stack Pointer Register. Execution of this command with the IC Register cleared causes the contents of the four low-order bits of the T Register to be placed in the stack pointer. Execution of this command following a stack input command places the contents of the stack pointer on the four low-order bits of the input bus.

SSU SELECT STACK UPPER (optional)



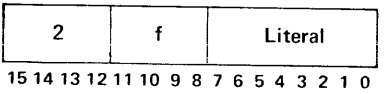
This command causes loading and reading of the upper seven bits of the selected L Save Stack register. Execution of this command with the IC Register cleared causes the contents of the T Register to be placed in the upper seven bits of the current L Save Stack level. Execution of the command following a stack input command places the contents of the upper seven bits of the current L Save Stack level on the input bus.

SSL SELECT STACK LOWER (optional)



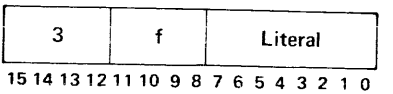
This command causes loading and reading of the lower eight bits of the selected L Save Stack register. Execution of this command with the IC Register cleared causes the contents of the T Register to be placed in the lower eight bits of the current L Save Stack level. Execution of the command following a stack input command places the contents of the lower eight bits of the current L Save Stack Level on the input bus.

LF LOAD FILE REGISTER



The contents of the eight-bit literal field are placed in the file register designated by f. Since file register 0 is not used for general storage, it is not to be loaded by this command. The condition flags and LINK are not affected.

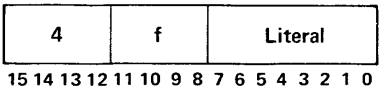
AF ADD TO FILE REGISTER



The contents of the eight-bit literal field are added to the contents of the file register designated by f. Since file register 0 is not used for general storage, it is not altered by this command. Two's complement subtraction may be performed by placing the two's complement of the operand in the literal field. The condition flags and LINK are not affected.

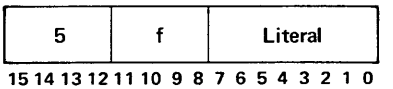
The following three test commands provide for arithmetic comparison and logical testing of bits. When the compare or test condition is met the next command in sequence is treated as a no operation, and there is a delay of 200 nanoseconds before executing the next command.

TZ TEST IF ZERO



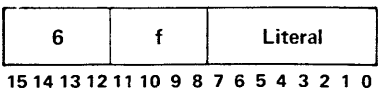
If, for all the one bits of the literal field, the corresponding bits of the file register designated by f are zero, the next command is not executed. This command performs the logical product of the literal and the contents of the file register and tests for zero result. The condition flags, LINK and designated file register are not affected.

TN TEST IF NOT ZERO



If, for any one bit in the literal field, the corresponding bit of the file register designated by f is also a one, the next command is not executed. This command performs a logical product of the literal and the contents of file register and tests for not zero. The condition flags, LINK and designated file register are not affected.

CP COMPARE

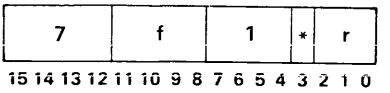


If the sum of the contents of the literal field and the file register designated by f is greater than 255, the next command is not executed. The condition flags and designated file register are not affected. The LINK stores the carry out of the adder. This means that if the skip is not taken, the content of LINK will be a 0, or if the skip is taken, the content of LINK will be set to a 1.

All forms of command seven unconditionally update the arithmetic condition codes in file 0, but do not affect the LINK. A destination of seven is undefined for these commands. In other words, the U Register may not be used to modify the command. All permissible variations of the basic command seven are explained. Unlisted c field values, (bits 7-4), are not assigned assembler mnemonics and will not be executed by the simulator.

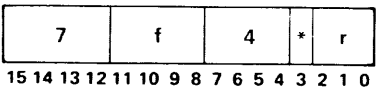
The following four commands provide special data flow operations.

ESS **ENTER SENSE SWITCHES**



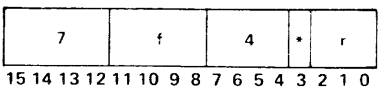
The status of the four console sense switches, with four low-order one bits appended, are placed in the file register designated by f, if * is zero, and in the register designated by r. The status of a switch is a one, when the switch is set.

SRF **SHIFT RIGHT FOUR**



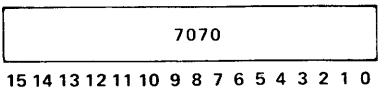
The four high-order bits of the file register designated by f are placed in the four low-order bits of that file, if * is zero, and in four low-order bits of the register designated by r. The four high order bits of the result are set to ones.

EIS **ENTER INTERNAL STATUS**



The eight internal status bits are placed in the file register designated by f, if * is zero, and in the register designated by r. The internal interrupt flag in file 0 is reset by this command, along with the console interrupt, real time clock, and power fail/restart status bits. Console step is reset upon release of the console switch and spare bits are controlled according to their individual implementation in hardware.

ECS ENTER CONSOLE SWITCHES

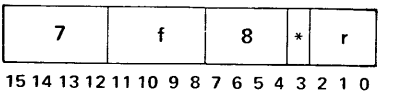


The contents of the eight low-order console command switches are ANDed with the eight low-order bits of the next command. The value of a switch is a one, when the switch is set. If the switch is either not set, or, as in the case of a basic panel, not there, its value is a zero. File register 0 and destination register 0 must be selected because data movement is not permitted.

The command could be used to implement eight additional sense switches. This is done by following the Enter Console Switches command with a Load Register or Load File command that has a literal value of all ones.

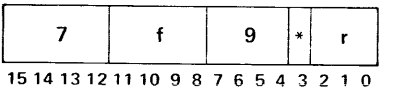
The next eight command descriptions explain control of the input/output buss for standard Microdata peripheral controllers. If a system is to contain only special controllers, these signals could be assigned any desired function subject to the following rule. Whenever the value in the I/O Control Register (IC), is set to four, five, six, or seven, the input buss is enabled in the CPU. This means that the input buss will be substituted for the T Register by any command that selects the T Register or its complement.

CIO CLEAR I/O



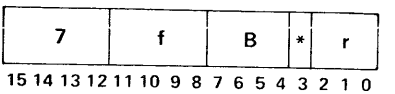
A value of 0 is placed in the I/O Control Register (IC), which removes all control signals from the I/O buss. This places the buss in the no activity mode. All standard Microdata peripheral controllers require the K Register to return to the zero state after each non-zero setting. When the current contents of the IC Register is 0, 1, 2, or 3, the file register designated by f is moved to the register designated by r. When the current contents of the IC Register is 4, 5, 6, or 7, the contents of the input buss are ANDed with the file register designated by f. The result is placed in the file register, if * is zero, and in the register designated by r.

COX CONTROL OUTPUT



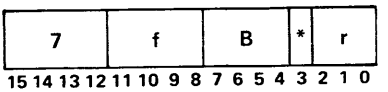
A value of 1 is placed in the IC Register which enables the control output signal until removed by a Clear I/O command. The contents of the file register designated by f are moved to the register designated by r.

DOX DATA OUTPUT



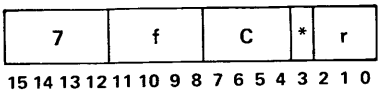
A value of 2 is placed in the IC Register which enables the data output signal until removed by a Clear I/O command. The contents of the file register designated by f are moved to the register designated by r.

SOX SPARE OUTPUT



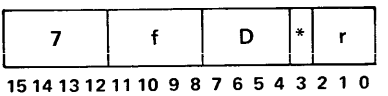
A value of 3 is placed in the IC Register which enables the spare output signal until removed by a Clear I/O command. This command also removes the MARKing current from the serial I/O channel causing a SPACEing condition. The contents of the file register designated by f are moved to the register designated by r.

CAK CONCURRENT ACKNOWLEDGE



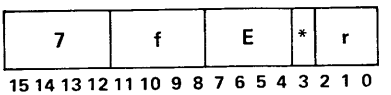
A value of 4 is placed in the IC Register which enables the concurrent acknowledge signal until removed by a Clear I/O command. Upon removal of this signal, the requesting controller will reset the concurrent request flag bit in file 0. The contents of the file register designated by f are moved to the register designated by r.

IAK INTERRUPT ACKNOWLEDGE



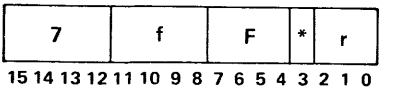
A value of 5 is placed in the IC Register which enables the interrupt acknowledge signal until removed by a Clear I/O command. Upon removal of this signal, the requesting controller will reset the external interrupt request flag bit in file 0. The contents of the file register designated by f are moved to the register designated by r.

DIX DATA INPUT



A value of 6 is placed in the IC Register which enables the data input signal until removed by a Clear I/O command. The contents of the file register designated by f are moved to the register designated by r.

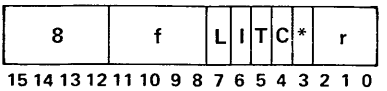
SIX STACK INPUT



A value of 7 is placed in the IC Register which enables the stack input signal until removed by a Clear I/O command. If the L Save stack is not in the machine, this signal may be used as a spare input.

The following commands operate on a specified file register. In addition some other operation such as input or memory control may take place. Some commands in this section are adaptations of others. For example, the Increment command is a form of the Add command. These adaptations are provided with their own mnemonics because of their frequent use and to simplify programming.

ADD **ADD**



The sum of the contents of the file register designated by f and the selected operand is formed. The sum is placed in the file register if * is zero and in the register designated by r. The state of the carry out of the high-order bit position of the adder is placed in LINK. The five Modifier bits perform the following control:

L – Link Control: The content of LINK is added to the sum. When this flag is a one the zero condition flag can be reset but not set. This allows for propagation of the zero test over multiple byte operations, the first of which would have this bit off.

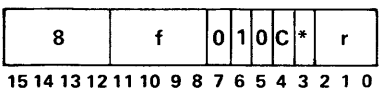
I – Increment: One is added to the sum.

T – Select T: The contents of the T Register or the input bus is selected as the B-bus operand. If this bit is off the operand is zero.

C – Condition Flag Update: The condition flags are updated according to the result of the current operation.

* – File Write Inhibit: The result is not placed in the file register.

INC **INCREMENT**

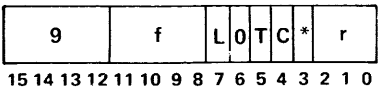


This command is a form of the Add command. The contents of the file register designated by f are incremented and the result is placed in the file register if * is zero and in the register designated by r. The state of the carry out of the high-order bit position of the adder is placed in LINK. The two modifier bits perform the following control:

C – Condition Flag Update: The condition flags are updated according to the result of the current operation.

* – File Write Inhibit: The result is not placed in the file register.

SBT SUBTRACT (Two's complement)



The two's complement difference of the contents of the file register designated by f and the selected operand is formed. The difference is placed in the file register if * is zero and in the register designated by r. The state of the carry out of the high-order bit position of the adder is placed in LINK. The four modifier bits perform the following control:

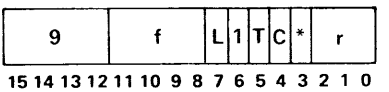
L – Link Control: The content of LINK is added to the one's complement difference. When this flag is a one the zero condition flag can be reset but not set. This allows for propagation of the zero test over multiple byte operations, the first of which would have this bit off.

T – Select T: The contents of the T Register or the Input Bus is selected as the B Bus operand. If this bit is off the operand is zero.

C – Condition Flag Update: The condition flags are updated according to the result of the current operation.

* – File Write Inhibit: The result is not placed in the file register.

SBO SUBTRACT (One's complement)



The one's complement difference of the contents of the file register designated by f and the selected operand is formed. The difference is placed in the file register if * is zero and in the register designated by r. The state of the carry out of the high-order bit position of the adder is placed in LINK. The four modifier bits perform the following control:

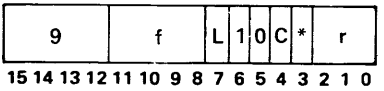
L – Link Control: The content of LINK is added to the difference. When this flag is one the zero condition flag can be reset but not set. This allows for propagation of the zero test over multiple byte operations, the first of which would have this bit off.

T – Select T: The contents of the T Register or the Input Bus is selected as the B Bus operand. If this bit is off the operand is zero.

C – Condition Flag Update: The condition flags are updated according to the result of the current operation.

* – File Write Inhibit: The result is not placed in the File register.

DEC DECREMENT

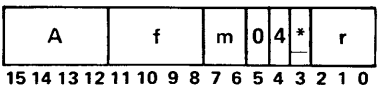


The contents of the file register designated by f is decremented by one and the result is placed in the file register if * is zero and in the register designated by r. The state of the carry out of the high-order bit position of the adder is placed in LINK. The two modifier bits perform the following control:

C – Condition Flag Update: The condition flags are updated according to the result of the current operation.

* – File Write Inhibit: The result is not placed in the file register.

RMF READ MEMORY, FULL CYCLE

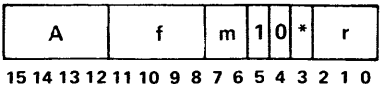


The contents of the file register designated by f in unaltered, incremented or decremented form as determined by m, is placed in the file register if * is zero and in the register designated by r. The condition flags and LINK are not affected. Subsequently a full cycle memory read is initiated at the location specified by the contents of the M and N Registers. Command execution is delayed if the memory is busy when the command is accessed. The accessed data is placed in the T Register 400 nanoseconds after the command is executed and the addressed memory location is left unaltered.

The contents of the file register are modified as follows:

m		
No flag	00 –	The contents of the selected file register are transferred unaltered to the specified destination register.
D	01 –	Decremented: The contents of the file register minus 1 are routed as specified unless the M register is specified to receive the result. When the M register is selected the contents of the file register, minus 1, plus the content of LINK are routed.
L	10 –	Add Link: The content of LINK is added to the contents of the file register and the sum is routed as specified.
I	11 –	Increment: The contents of the file register is incremented by 1 and the result is routed as specified.

RMH READ MEMORY, HALF CYCLE

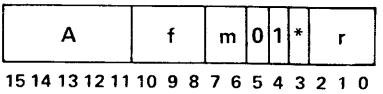


The contents of the file register designated by f in unaltered, incremented or decremented form as determined by m, is placed in the file register if * is zero and in the register designated by r. The condition flags and LINK are not affected. Subsequently a half cycle memory read is initiated at the location specified by the contents of the M and N Registers. Command execution is delayed if the memory is busy when the command is accessed. The accessed data is placed in the T Register 400 nanoseconds after the command is executed and the memory location is left in an all one's condition.

The contents of the file register are modified as follows:

m	
No flag	00 – The contents of the selected file register are transferred unaltered to the specified destination register.
D	01 – Decremented: The contents of the file register minus 1 are routed as specified unless the M Register is specified to receive the result. When the M Register is selected the contents of the file register, minus 1, plus the content of LINK are routed.
L	10 – Add Link: The content of LINK is added to the contents of the file register and the sum is routed as specified.
I	11 – Increment: The contents of the file register is incremented by one and the result is routed as specified.

WMF WRITE MEMORY, FULL CYCLE



The contents of the file register designated by f in unaltered, incremented or decremented form as determined by m is placed in the file register if * is zero and in the register designated by r. The condition flags and LINK are not affected. Subsequently a full cycle memory write operation is initiated at the location specified by the contents of the M and N Registers. Command execution is delayed if the memory is busy when the command is accessed.

The data to be written must be in the T Register at the time the command is executed, or must be entered into the T Register with the next command. The T Register may be used for other uses with the second command after the WMF.

The contents of the file register are modified as follows:

m

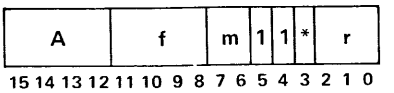
No flag 00 – The contents of the selected file register are transferred unaltered to the specified destination register.

D 01 – Decremented: The contents of the file register minus 1 are routed as specified unless the M Register is specified to receive the result. When the M Register is selected the contents of the file register, minus 1, plus the content of LINK are routed.

L 10 – Add Link: The content of LINK is added to the contents of the file register and the sum is routed as specified.

I 11 – Increment: The contents of the file register is incremented by one and the result is routed as specified.

WMH WRITE MEMORY, HALF CYCLE

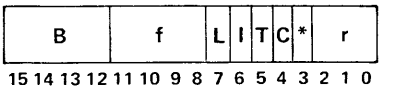


The contents of the file register designated by f in unaltered, incremented or decremented form as determined by m is placed in the file register if * is zero and in the register designated by r. The condition flags and LINK are not affected. Subsequently a half cycle memory write operation is initiated at the location specified by the contents of the M and N Registers. Command execution is delayed if the memory is busy when the command is accessed. The data to be written must be in the T Register when the command is executed. The contents of the addressed memory location must be all one bits for a proper write to take place, because there will be an ANDing of the original contents of the memory location and the contents of the T Register. The T Register may be used for other purposes with the first command after WMH.

The contents of the file register are modified as follows:

m		
No flag	00 –	The contents of the selected file register are transferred unaltered to the specified destination register.
D	01 –	Decremented: The contents of the file register minus 1 are routed as specified unless the M Register is specified to receive the result. When the M Register is selected the contents of the file register, minus 1, plus the content of LINK are routed.
L	10 –	Add Link: The content of LINK is added to the contents of the file register and the sum is routed as specified.
I	11 –	Increment: The contents of the file register are incremented by one and the result is routed as specified.

CPY COPY



The selected operand is placed in the file register designated by f if * is zero and in the register designated by r. The LINK is not affected. The 5 modifier bits perform the following control:

L – Link Control: The content of LINK is added to the operand. When this bit is a one the zero condition flag can be reset but not set. This allows for propagation of the zero test over multiple byte operations, the first of which would have this bit off.

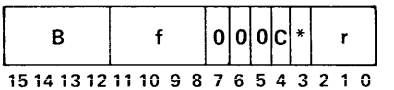
I – Increment: One is added to the operand.

T – Select T: The contents of the T Register or the input bus is selected as the B Bus operand. If this bit is zero the operand is zero.

C – Condition Flag Update: The condition flags are updated according to the result of the current operation.

* – File Write Inhibit: The result is not placed in the file register.

ZOF ZERO FILE

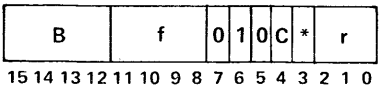


A value of zero is placed in the file register designated by f if * is zero and in the register designated by r. The LINK is not affected. The two modifier bits perform the following control:

C – Condition Flag Update: The condition flags are updated according to the zero data.

* – File Write Inhibit: The zero is not placed in the file register.

POF PLUS ONE TO FILE

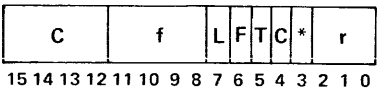


A value of plus one is placed in the file register designated by f if * is zero and in the register designated by r. The LINK is not affected. The two modifier bits prefer the following control:

C – Condition Flag Update: The condition flags are cleared.

* – File Write Inhibit: The plus one is not placed in the file register.

LOR LOGICAL OR



The logical inclusive-OR of the contents of the file register designated by f and the selected operand is placed in the file register if * is zero and the register designated by r. The LINK is not affected. The five modifier bits perform the following control:

L – Link Control: When this bit is a one the zero condition flag can be reset but not set. This allows for propagation of the zero test over multiple byte operations, the first of which would have this bit off.

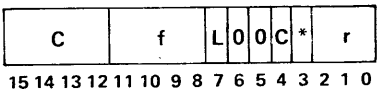
F – Select T Complement: The one's complement of the contents of the T Register or input bus are selected as the B Bus operand. If the T Register is also selected the effective operand contains all one's.

T – Select T: The contents of the T Register or the Input Bus is selected as the B Bus operand. If both the T and F bits are off the selected operand is zero.

C – Condition Flag Update: The condition flags are updated according to the result of the current operation.

* – File Write Inhibit: The result is not placed in the file register.

MOV MOVE



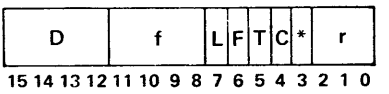
The contents of the file register designated by f are moved to the register designated by r. The LINK is not affected. The three modifier bits perform the following control:

L – Link Control: When this bit is a one the zero condition flag can be reset but not set. This allows for propagation of the zero test over multiply byte operations, the first of which would have this bit off.

C – Condition Flag Update: The condition flags are updated according to the contents of the file register.

* – File Write Inhibit: The result is not placed in the file register.

XOR EXCLUSIVE-OR



The exclusive-OR of the contents of the file register designated by f and the selected operand is placed in the file register if * is zero and in the register designated by r. The LINK is not affected. The five modifier bits perform the following control:

L – Link Control: When this bit is a one the zero condition flag can be reset but not set. This allows for propagation of the zero test over multiple byte operations, the first of which would have this bit off.

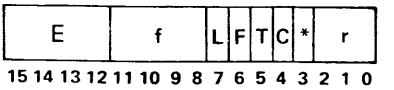
F – Select T Complement: The one's complement of the contents of the T Register or Input Bus are selected as the B Bus operand. If the T Register is also selected the effective operand contains all one's.

T – Select T: The contents of the T Register or the Input Bus is selected as the B Bus operand. If both the T and F bits are off the selected operand is zero.

C – Condition Flag Update: The condition flags are updated according to the result of the current operation.

* – File Write Inhibit: The result is not placed in the file register.

AND **AND**



The logical product (AND) of the contents of the file register designated by f and the selected operand is placed in the file register if * is zero and in the register designated by r. The LINK is not affected. The five modifier bits perform the following control:

L – Link Control: When this bit is a one the zero condition flag can be reset but not set. This allows for propagation of the zero test over multiple byte operations, the first of which would have this bit off.

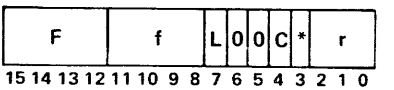
F – Select T Complement: The one's complement of the contents of the T Register or Input Bus are selected as the B Bus operand. If the T Register is also selected the effective operand contains all one's.

T – Select T: The contents of the T Register or the Input Bus is selected as the B Bus operand. If both the T and F bits are off the selected operand is zero.

C – Condition Flag Update: The condition flags are updated according to the result of the current operation.

* – File Write Inhibit: The result is not placed in the file register.

SFL **SHIFT LEFT**



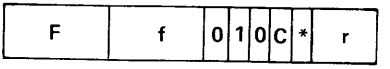
The contents of the file register designated by f are shifted one bit position to the left and placed in the file register if * is zero and the register designated by r. The high-order bit which is shifted out is placed in LINK. A zero or the content of LINK is shifted into the vacated bit position as determined by the L modifier bit. The three modifier bits perform the following controls:

L – Link Control: LINK content is shifted into low-order bit of result. When this bit is a one the zero condition flag can be reset but not set. This allows for propagation of zero test over multiple byte operations, the first of which would have this bit off.

C – Condition Flag Update: The condition flags are updated according to the result of the current operation.

* – File Write Inhibit: The shifted register contents are not placed in the file register.

SLI SHIFT LEFT AND INSERT

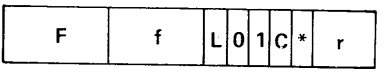


The contents of the file register designated by f are shifted 1 bit position to the left and placed in the file register if * is zero and the register designated by r. The high-order bit which is shifted out is placed in LINK. A one is inserted in the vacated low-order bit position. The two modifier bits perform the following control:

C – Condition Flag Update: The condition flags are updated according to the result of the current operation.

* – File Write Inhibit: The shifted register contents are not placed in the file register.

SFR SHIFT RIGHT



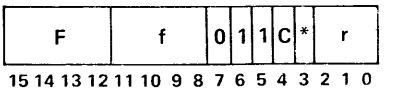
The contents of the file register designated by f are shifted one bit position to the right and placed in the file register if * is zero and the register designated by r. A zero or the content of LINK is shifted into the vacated bit position as determined by the L modifier bit. The low-order bit which is shifted out is placed in LINK. The three modifier bits perform the following control:

L – Link Control: LINK content is shifted into low-order bit of result. When this bit is a one the zero condition flag can be reset but not set. This allows for propagation of zero test over multiple byte operations, the first of which would have this bit off.

C – Condition Flag Update: The condition flags are updated according to the result of the current operation.

* – File Write Inhibit: The shifted register contents are not placed in the file register.

SRI SHIFT RIGHT AND INSERT



The contents of the file register designated by f are shifted 1 bit position to the right and placed in the file register if * is zero and the register designated by r. A one is inserted into the vacated high-order bit position. The low-order bit which is shifted out is placed in LINK. The two modifier bits perform the following control:

C – Condition Flag Update: The condition flags are updated according to the result of the current operation.

* – File Write Inhibit: The shifted register contents are not placed in the file register.

The two Execute commands are special commands which cause the eight high-order bits of the U Register to be ORed with the eight high-order bits of control memory output. The ORing is performed before the command is gated into the R register. The actual command executed is a combination of the bits in the U Register and those read out of control memory. The Execute command is designated by zero in the four high-order bits of the command in control memory. An effective command with zeros in these bits is a Jump Extended. The 12 remaining bits of the Execute command can be coded as needed. The same effect as the Execute command can be obtained by coding destination register 7 on operate commands.

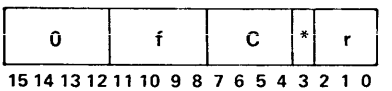
The Execute command is used for command modifications generally to save on the number of commands needed in a program. Three uses of this feature are:

- Indexing of file registers in a loop.
- Selection of alternate file register depending on program variables.
- Performing different functions such as load, add, subtract etc. in a common string of coding.

An example is shown below. The X'81' contents of the U Register merge with the execute command in control memory to form an Add to file register 5.

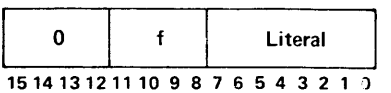
U Register	'81'
Control Memory	'0420'
Effective Command	'8520'

EOT EXECUTE, OPERATE TYPE



The eight-bit contents of the U Register are ORed with the eight high-order bits of the command to form an effective command which is then executed. The f, C, *, and r fields are used as described for the desired effective command. In coding the c-field bits any modifier may be used.

ELT EXECUTE, LITERAL TYPE



The eight-bit contents of the U Register are ORed with the eight high-order bits of the command to form an effective command which is then executed. The f and Literal fields are used as described for the desired effective command.

CHAPTER 4

CONTROL PANEL OPERATION

The MICRO 1600 system console control panel (Figure 2) provides for control of the running of the processor, display of register contents and execution of commands. The panel is very useful in debugging micro-programs and can also be used for macro level machine control and software debugging. The control panel can be replaced by a minimum basic console control panel which eliminates the display and manual command execution.

All console panels are pluggable and fully interchangeable without modification of the computer.

An optional parallel Teletype controller, physically contained within the control consoles, may be specified.

SYSTEM CONSOLE

The system console provides control plus a selectable display of all hardware registers in the machine including the files. It is designed for maintenance operations and for installations where system development and firmware checkout is being performed.

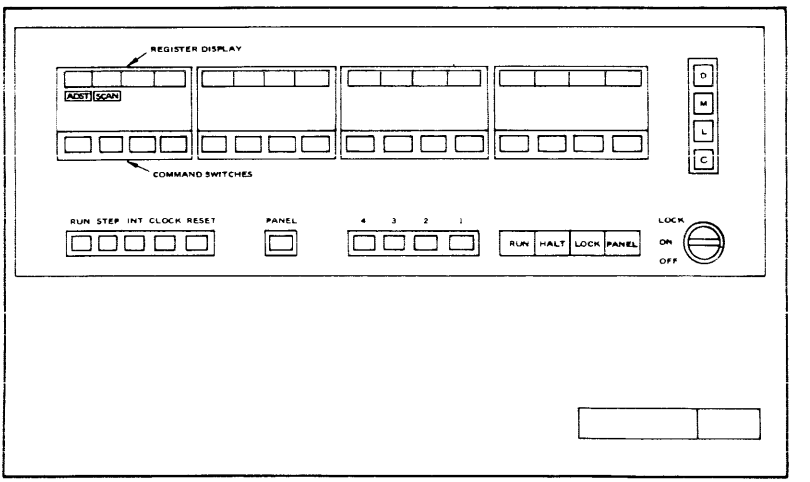


Figure 2. MICRO 1600 System Console

DISPLAYS

Data Display

The 16-bit data indicators (16 lamps on console) display the 8-bit A Bus, Memory Address, 16-bit Control Memory Output, or 12-bit Control Memory Address as selected by the Display Selector switches.

Run

The RUN indicator is on when the processor is running.

Halt

The HALT indicator is on when the power is applied and the processor is not running.

Lock

The LOCK indicator is on when the panel is disabled.

Panel

The PANEL indicator is on when the command switches are enabled and substituting for the control memory.

Scan

The SCAN indicator is on when in the SCAN mode. This takes place with the PANEL switch off and command switch 14 is on.

Address Stop

The ADST indicator is on when the processor has stopped because of address breakpoint.

SWITCHES

Display Selector

The four interlocked switches located in the upper right hand corner select one of the four displays as follows:

D – Data: This 8-bit display is the processor's A Bus. The data on the A bus when the processor is halted and in the panel enable mode depends on the setting of the command switches.

M – Memory Address: This 16-bit display is of the memory address lines. This is normally the contents of the M and N registers.

L – Control Memory Address: This 12 or 15-bit display is the contents of the L Register.

C – Control Memory: This 16-bit display is of the output of the control memory. When the processor is halted the R Register contains the same data.

Command Switches

These 16 locking switches are substituted for the control memory when the PANEL switch is in the down position. When the processor is halted

the switch setting is constantly clocked into the R Register and depressing the CLOCK switch causes the command set in the switches to be executed. The command may also be executed repeatedly by depressing the RUN switch. These switches are used to gate registers onto the A Bus for display and for entering data into registers.

Panel Switch

This locking switch selects the source of commands. When in the normal up position the control memory is used and when in the down position the 16 command switches on the panel are substituted for the control memory.

Sense Switches

The four locking sense switches are available on the control panel. These switches may be read by an Enter Sense Switch command.

Run

This momentary contact switch places the processor in the run mode causing it to execute microcommands.

Step

This momentary contact switch places the processor in the run mode and as long as the switch is depressed causes an internal interrupt. The halt internal interrupt is bit 7 of the internal status. This switch is normally microprogrammed to cause a processor halt. Since the processor is forced to run when the switch is depressed, the machine can be microprogrammed to cause a single macro instruction to be executed.

Interrupt

This momentary contact switch places the processor in the run mode and causes an internal interrupt. The console interrupt is bit 0 of the internal status. This switch is normally microprogrammed to cause a console interrupt.

Clock

This momentary contact switch causes the processor to execute a single microcommand. If the processor is running at the time the switch is depressed, the processor will come to a forced halt following the current microcommand execution.

Reset

This momentary contact switch halts the processor and clears the L register, I/O control register and other control flip-flops. The reset is made available to I/O devices. Since the current microcommand execution will not be completed, the computer should not be stopped by this switch.

OFF-ON-LOCK

A three-position key lock switch applies power and disables the panel. The key can be removed in any position. In the OFF position AC power is

turned off. In the ON position power is applied and the panel is active. In the LOCK position power remains on, but the panel switches are not active except for the sense switches.

Control Memory Scan

When the PANEL switch is off and command switch 14 is depressed the processor is in a control memory scan mode and the SCAN indicator is turned on. This mode allows advancing the L Register without executing a command by depressing the CLOCK switch. In this manner the operator can sequentially step through control memory addresses and view its contents on the data display indicators.

Address Stop

The processor can be stopped at a particular control memory address. The address is set on the command switches 14-0 and switch 15 is turned on. When the contents of the L Register are the same as the address set into the command switches the processor is halted with the L Register containing the stop address and the ADST indicator on.

Address Sync

A sync jack is mounted on the rear of the front panel for maintenance purposes. A positive pulse of 200 nanosecond duration is obtained when the contents of the L Register are the same as the address set into command switches 14-0.

REGISTER DISPLAY AND ENTRY

Display

The processor registers can be displayed directly by selecting the proper display selector or indirectly by use of commands set into the command switches to cause the register to be gated to the A Bus where it can be displayed by selecting 'D'.

The R, U, MD and OD Registers cannot be displayed, but the R Register will hold the same information as on the R Bus when the processor is halted. The M, N and L Registers can be displayed by selecting them with the display selector.

The file registers, T Register and LINK can be displayed indirectly by setting the commands shown below into the command switches and selecting the data display (A Bus).

Register	Command Setting
Selected File Register X	CX00
T Register	B020
LINK (AL)	B080
LINK (ML)	B082

Enter

Information can be entered into a register by executing a command from the panel. This requires turning on the PANEL switch, setting the command into the command switches and pressing the CLOCK switch. In addition control functions such as interrupt enable or file select can be performed by executing the appropriate command. The commands for placing the literal 'ZZ' in a register are shown below:

Register	Command	
T	11ZZ	
M	12ZZ	
N	13ZZ	
U	16ZZ	
File Register X	2XZZ	
L	0ZZZ	(U Register must be cleared)

CHAPTER 5

MICRO ASSEMBLER PROGRAM

The Micro Assembly Program, MAP1600, is a two-pass symbolic program which assembles MICRO 1600 microprograms on MICRO 820, 821, 1600/20 or 1600/21 computers with 8K bytes of core memory. The basic assembler is designed for use on ASR33 teletype with paper tape reader and punch. Other versions of the program permit use of a high-speed paper tape reader and punch, a card reader, or a line printer.

Output from the assembler consists of an assembly listing and a binary program tape. This program tape is used as the input to the Micro 1600 simulator, SIM1600, for static environmental checkout and, if necessary, as input into the electrically alterable read only memory system, AROS1600, for system checkout in a dynamic environment. When a final, debugged, version of the microprogram is assembled, the program tape is the input to a utility program, ICM1600, that generates the memory maps necessary for manufacturing of either bipolar integrated circuits or field programmable read-only memories.

The MAP1600 assembly language includes the following features:

Address Arithmetic – Decimal and hexadecimal numbers, symbolic addresses, and arithmetic expressions.

Listing Control – The format of the listing is automatically controlled and comment statements may be included.

Diagnostics – Diagnostics for source statement errors are included in the output listing.

SOURCE LANGUAGE

The source language is a sequence of symbolic commands called statements, which are punched on paper tape or cards. Each statement may consist of from one to four entries: a name field, an operation field, an operand field, and a comments field. The maximum length of a statement is 72 characters.

Source program paper tapes that are prepared off-line on a teletype may be in a free form format. This means they may have one or more spaces between fields. All paper tape statements must be terminated by a carriage return, line feed, and two rub-out characters. If the first character of a statement is a left arrow ←, it is treated as an end of tape indicator and the assembler will halt to permit another tape to be inserted before continuing. This allows large programs to be separated into several smaller tapes to make editing easier. Source program tapes also may be prepared using the Tape Editor; the tapes will be in a compressed format that removes strings of blank characters.

STATEMENT FORMAT

Name Field

The name field entry is a symbol composed of from one to six characters starting with character position 1 and terminating with the first blank. Only the first three characters are retained, therefore, they must be unique. The first character of a symbol is alphabetic or a period; subsequent characters may be alphabetic, numeric or a period. A name entry is usually optional and the type of command determines the legal content of the name field. The symbol takes on the current value of the assembler's location counter unless assigned another value by an assembler instruction. When an asterisk (*) appears in character position 1, the remainder of the line is considered as comment and is not processed by the assembler except to place it on the listing.

Operation Field

The operation field entry consists of a two or three-character mnemonic operation code specifing the machine command or assembler instruction. Class one commands may use a special symbol (*) suffixed to the mnemonic which indicates that updating of the source file is to be inhibited. All other commands are followed by blanks. The field begins with the first non blank character following the name field in paper tape or with ccolumn 8 in cards, and is a minimum of 4 characters in length. Two or three character commands are considered to have a blank as the third and fourth character of the mnemonic.

Operand Field

The operand field entries identify and describe registers of data to be acted upon by a command. One or more operands may be written, depending on the needs of the command. No blanks may appear within the operand field. In the classes of commands requiring source file operands, a comma is required to separate the source file operand from any following operand entries. The operand field may start anywhere between the first and third characters following the operation field. When punched in cards, column 14 is the normal column. It is terminated by the first blank.

C-Field Designators

C-Field designators are only allowed in Class One commands. These C-Field designators must be preceded by a source file operand and/or a comma separator, and must proceed any destination register designator. Only specific C-Field designators are allowed and the operation field regulates their use. C-Field designators may be separated by commas for clarity, if desired.

Following are the legal C-Field designators:

Designator	Definition	Bit Value
C	Modify Condition Codes	0001
T	Select T Register	0010
F	Select T Complement	0100
I	Add One/Increment	0100
	for Read/Write Memory	1100
D	Decrement	0100
L	Link Control/Add Link	1000

Destination Register Designators

The destination register designators are only allowed in Class One Commands. If specified, it may follow either the C-Field designators, or the source file operand if no C-Field designators are specified. The destination register is enclosed by parenthesis.

Following is a list of the legal destination register designators enclosed by parenthesis:

Designator	Register
(T)	T register
(M)	M register
(N)	N register
(L)	L register (even address pages)
(K)	L register (odd address pages)
(U)	U register
(S)	Or U with command (op codes 8 to F only)

Inhibit Source File Update

This indicator is only allowed by Class One commands. The indicator is an asterisk (*) following the three character operation field. When punched on cards, this indicator is located in column 11.

Comments Field

Comments describing the information about the program may be inserted between the end of the operand field and column 72. All characters, including spaces, may be used in writing a comment. If the listing is printed on a teletype, only the first 53 characters of the source line are printed.

OPERAND FIELD EXPRESSIONS

Source File and Literal Expressions

Expressions in the operand field are made up of one or more terms which are connected by + and - arithmetic operators. No parenthetical expressions are allowed. Each term of the expression represents a value. Values may be assigned by the assembler program (symbols), or they may be inherent in the term itself (constants). The range of values depends on the operand and the command.

Symbols

A symbol is composed of one to six characters, but only the first three are recognized, and therefore must be unique. The first character must be alphabetic or a period; subsequent characters may be numeric, alphabetic, or a period. Imbedded blanks are not allowed and the assembler stops scanning the symbol with the first character which is not alphanumeric or a period. All symbols, except the special symbol (*) used in an operand field, must be defined by a single appearance in the name field of statement within the program.

Special Symbol

The special symbol (*) represents the momentary value of the assembler's location counter. It may be used as any other symbol in an expression but must never appear in the name field.

Constants

The values of the constant terms are not assigned by the assembler program but are inherent in the terms. There are two types of constant terms: decimal and hexadecimal.

a. Decimal Constant

   A decimal constant is an unsigned decimal number. The value must be less than 65,536.

b. Hexadecimal Constants

   A hexadecimal constant is an unsigned hexadecimal number of up to four characters written as a sequence of hexadecimal digits. The digits are enclosed in single quotation marks and preceded by the letter X. Each hexadecimal digit represents a four-bit binary number. The characters A through F are used to identify the hexadecimal integers 10 through 15.

MACHINE COMMANDS

Machine commands occupy one word (16 bits) of read-only memory. These commands are divided into five classes or formats. All classes allow a name, operation, and comments field, while the operand field contents is defined by class.

The five classes are defined as follows:

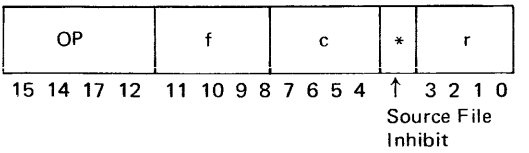
CLASS ONE

The Operation Field may contain the Source File update inhibit designator (*). The operand field consists of:

1. Source file Expression (f)
2. C-Field designators if allowed by the particular operation code. (c)
3. Destination register designator. (r)

The Operand field must contain a Source file expression first and it must be separated from the remaining operand field by a comma. Either or both

the C-Field designators and Destination register designator are then specified if desired.

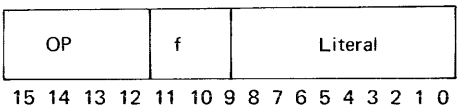


CLASS TWO

The operand field consists of:

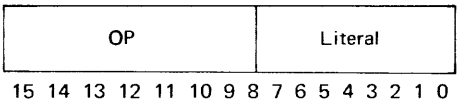
1. Source file Expression (f)
2. Literal expression

The source file expression must be followed by a comma if a literal expression follows. If no literal expression is specified it is assumed to be zero. Only an 8-bit literal value is expressed in the object code.



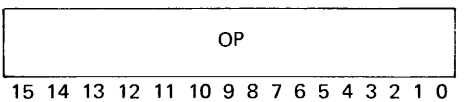
CLASS THREE

The operand field consists of a literal expression. Only an 8-bit literal value is expressed in the object code.



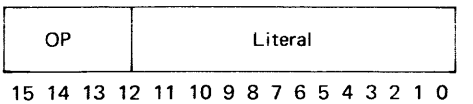
CLASS FOUR

The generic class of commands occupy all 16 bits of the microcommand format.



CLASS FIVE

The operand field is a literal expression of which the low-order 12-bit value is expressed in the object code.



The only Class Five Op Code mnemonic is JE.

ASSEMBLER INSTRUCTIONS

Five assembler instructions are included for control of the assembly process and the output listing.

ORG – Set Location Counter

The ORG assembler instruction alters the setting of the location counter. The name field entry, if any, will be assigned the value of the program counter after it is altered. The operand field of ORG must contain an expression whose value will be placed in the location counter. All symbols in the expression must have been previously defined when the instruction is first encountered. The next comma which places object code in the program is forced to begin a new object record.

EQU – Equate Symbol

The EQU assembler instruction is used to define a symbol by assigning to it the value of the operand field expression. Any symbols appearing in the expression must have been previously defined when the instruction is first encountered. A name field entry must be present.

DC – Define Constant

The DC assembler instruction is used to generate any 16-bit value in order to create a command that is not symbolically recognized by the assembler. The operand field may contain any expression.

EJE – Page Eject

The EJE assembler instruction is only recognized in Pass 2. When encountered, it ejects the assembler listing to the next page and prints the page heading. Two or more consecutive EJE statements yield only one top of form operation.

END – End Assembly

The END assembler instruction terminates the assembly of a program and must be the last statement in a source program. Any operand field entry is ignored.

ASSEMBLY LISTING

A listing generated during the second pass of the assembly supplies a side-by-side representation of source statements, generated object code, control memory address of the command, and diagnostic messages. The format locates the various items at fixed positions across the printed page to produce a columnar organization. Print positions for listing are shown below.

FORMAT

Printer Columns	Contents
1 - 4	Memory address of assembled command.
6 - 15	Hexadecimal digits of one word of object code.
17 - 18	Diagnostic message identifiers.
20 - 25	Name field of source statement.
27 - 29	Operation code of source statement.
30	Source File update inhibit designator.
33 - 72	Operand field and comment section.

Source paper tapes that are punched in the free form or compressed format will be listed in above fixed form format.

ERROR FLAGS

Diagnostic messages are indicated by single character identifiers appearing in columns 17 or 18. If more than two errors are detected for a single source statement, only the last two encountered will be listed. The diagnostic symbols are as follows:

A – Address Error

This error occurs when an address expression in the operand field is incorrectly written or the value is out of range for one of the operands. An error flag will occur for each operand in error or out of range.

C – C-Field Error

This error occurs when an illegal C-Field designator is encountered with class one commands.

D – Destination Register Error

This error occurs when an illegal Destination register designator is encountered with class one commands.

F – Name Table Full

This error occurs when the name table is full. The name or operand in this statement was not entered into the table.

M – Multidefined Symbol Error

This error occurs when the symbol in the name field has been previously defined by appearing in the name field of another instruction. During pass 1, this error references the statements with name field errors, while operand field occurrences of the multidefined names are indicated on pass 2.

N – Name Field Error

This error flag occurs when the symbol in the name field starts with a character other than alphabetic or period or contains a non-alphanumeric or non-period character.

O – Operation Mnemonic Error

This error occurs when the assembler does not recognize the contents of the operation field. A 16-bit zero value is assembled to allow patching.

R – Range Error

This error occurs when the expression in the operand field of a JP operation is beyond the current 1K block of read-only memory.

S – Source File Error

This error occurs when the value of a Source file expression exceeds 15.

U – Undefined Symbol Error

This error occurs when the symbol encountered in an expression of the operand field is not defined by an appearance in the name field.

PROGRAM TAPE FORMAT

The binary paper tape format generated by the assembler is shown in Figure 3. This format allows for variable length records of up to 64_{10} 16-bit microcommands. Each record contains a count of the number of data bytes in the record, (2 bytes per command), the 15-bit address where loading is to start, and a checksum byte. The last byte of each record is the checksum. It is the summation of the byte count, load address, and data bytes formed on a signed eight bit basis with any overflow added into the least significant bit of the sum.

The END assembler instruction causes an end of tape record to be attached. This record contains a byte count, load address, and checksum of all zeros.

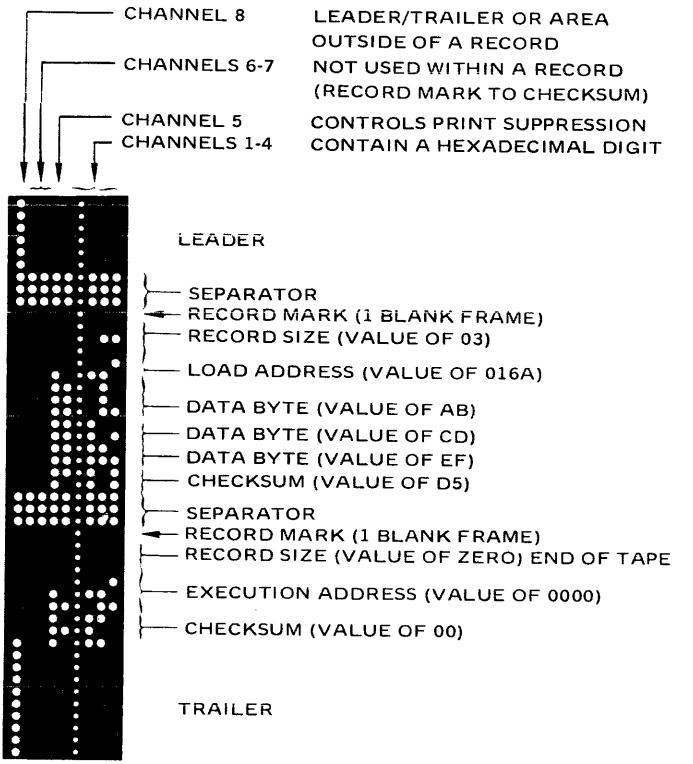


Figure 3. Binary Paper Tape Format

SAMPLE LISTING

The two sample listings show the format of a listing and provides examples of how to write each command type, literals, constants, and assembler instructions. Nine types of error conditions are also illustrated.

Upon completion of Pass 1, all undefined symbols will be printed. For the programmers information, a list of all unused symbols will also be printed.

PASS 1

```
PAGE 001

S           LF      16,100          FILE NUMBER .GT. 15
O   MM      WHO     AM,I            UNDEFINED MNEMONIC
N   1       LF      6,2             ILLEGAL NAME
M   MLT     TN      0,2             MULTIPLE DEFINED NAME
C           ADD     5,F             ILLEGAL FOR ADD
C           INC     6,T             ILLEGAL FOR INCREMENT
D           MOV     7,(X)           ILLEGAL DESTINATION REG
A           LF      3               MISSING LITERAL
A           LF      4,              MISSING LITERAL
A           LF      NF,             UND. FILE, MISSING LIT.
UNDEFINED NAMES ....
OUT
NG
NF
UNUSED NAMES .....
MM
```

PASS 2

```
PAGE 001

                    *  EXAMPLE OF ALL ASSEMBLER ERROR
                    *  DIAGNOSTICS EXCEPT NAME TABLE FULL
0000 2064       S           LF      16,100          FILE NUMBER .GT. 15
0001 7000       U   MLT     JE      OUT             UNDEFINED SYMBOL
0002 0000       O   MM      WHO     AM,I            UNDEFINED MNEMONIC
0003 2602           1       LF      6,2             ILLEGAL NAME
0004 3701       M           AF      7,MLT           MULTIPLE DEFINED NAME
0005 5002           MLT     TN      0,2             MULTIPLE DEFINED NAME
0006 5520       C           ADD     5,F             ILLEGAL FOR ADD
0007 8640       C           INC     6,T             ILLEGAL FOR INCREMENT
0008 C700       D           MOV     7,(X)           ILLEGAL DESTINATION REG
0009 1400       R           JP      X'400'          OUTSIDE CURRENT 1K WORDS
000A 2300       A           LF      3               MISSING LITERAL
000B 2400       A           LF      4,              MISSING LITERAL
000C 2500       U           LF      5,NG            UNDEFINED LITERAL
000D 2000       UA          LF      NF,             UND. FILE, MISSING LIT.
000E 0000                   END
                    *  EXAMPLE OF HOW TO CODE VARIOUS MICROCOMMANDS
                    *  CLASS 1 FORMAT
0000 7026                   EOT     0,T,(U)         COULD BE A COPY T TO U
0001 7129                   SRF*    1,(T)           SHIFT RIGHT 4 TO T
0002 70A0                   DOX     0               DATA OUTPUT I/O SIGNAL
0003 8261                   ADD     2,IT,(T)        FILE 2+1+T TO FILE 2, T
0004 8350                   INC     3,C             FILE 3+1 WITH COND CODE
```

```
0005 9420              SBT   4,T          FILE 4=T TO FILE 4
0006 9543              DEC   5,(N)        FILE 5=1 TO FILE 5, N
0007 A642              RMF   6,D,(M)      FILE 6=1+LINK, FULL READ
0008 A722              RMH   7,(M)        FILE 7 TO M, HALF READ
0009 A853              WMF   8,D,(N)      FILE 8=1 TO N FULL WRITE
000A A931              WMH   9,(T)        FILE 9 TO T, HALF WRITE
000B BAA0              CPY   10,LT        T+LINK TO FILE 10
000C BB01              ZOF   11,(T)       ZERO TO FILE 11 AND T
000D BC43      POF     POF   12,(N)       PLUS 1 TO FILE 12 AND N
000E CD40              LOR   13,F         OR FILE 13 WITH T COMP
000F CE06              MOV   14,(U)       MOVE FILE 14 TO U
0010 DF36              XOR*  15,T,C       IF FILE 15=T, RESULT 0
0011 E120              AND   1,T          FILE 1 ANDED WITH T
0012 F280              SFL   2,L          SHIFT LEFT, INSERT LINK
0013 F340              SLI   3            SHIFT LEFT, INSERT A 1
0014 F420              SFR   4            SHIFT RIGHT, INSERT A 0
0015 F570              SRI   5,C          LSB SAVED IN OVERFLOW

               *  CLASS 2 FORMAT
0016 0618              ELT   6,24         COULD BE LOAD FILE 6
0017 270D              LF    7,POF        ADDRESS OF POF COMMAND
0018 3806              AF    8,6          ADD 6 TO FILE 8
0019 4001              TZ    0,X'01'      SKIP IF OVERFLOW COND
001A 5A01              TN    TEN,1        SKIP IF FILE 10 IS ODD
001B 6B10              CP    11,-240      SKIP IF FILE 11 ,GT, 239

               *  CLASS 3 FORMAT
001C 1101              LT    1            1 TO T
001D 1202              LM    2            2 TO M
001E 1303              LN    3            3 TO N, CLEAR M
001F 142C              JP    .XY          JUMP WITHIN 1024 WORDS
0020 16AB              LU    X'AB'        HEX 'AB' TO U
0021 1728              LS    X'28'        ENABLE RTC AND EXT, INT,
0022 19A0              RLT   X'A0'        HEX 'A0' TO T AND RETURN
0023 1B0A              SBL   2            SELECT 3RD BANK OF 4096

               *  CLASS 4 FORMAT
0024 1000              NOP                USED FOR 200NS DELAY
0025 1020      RTN     RTN                RETURN FROM LAST JE SAVE
0026 1040              SPF                SELECT PRIMARY FILES
0027 1050              RSP                RETURN, SELECT PRIMARY
0028 1060              SSF                SELECT SECONDARY FILES
0029 10A0              RSS                RETURN, SELECT SECONDARY
002A 1700              HLT                HALTS CPU, BUT NOT DMA
002B 1A02              MLC                OD ANDED ON NEXT COMMAND
002C 1B01      .XY     ISP                MOVE STACK TO NEXT LEVEL
002D 7272              ECS                PANEL SWITCHES ANDED

               *  CLASS 5 FORMAT
002E 2702              JE    ORG          JUMP AND SAVE FOR RETURN

               *  ASSEMBLER INSTRUCTIONS
2702           ORG     ORG   X'700'
000A           TEN     EQU   10           SYMBOLIC NAME FOR A FILE
2702 0000              DC    0            FILLER FOR UNUSED ROM
2703 0000              END
```

MICROCOMMANDS (CLASS ORDER)

Class 1 – Mnemonic/File Inhibit – File Name, C-Field Design (Destination Reg. Design)

COMMAND	MNEMONIC	OPERAND FIELD
EXECUTE, OPERATE TYPE	EOT*	f,LCTIFD(r)
ENTER SENSE SWITCHES	ESS*	f,(r)
SHIFT FILE RIGHT 4	SRF*	f,(r)
ENTER INTERNAL STATUS	EIS*	f,(r)
CLEAR I/O MODE	CIO*	f,(r)
CONTROL OUTPUT	COX*	f,(r)
DATA OUTPUT	DOX*	f,(r)
SPARE OUTPUT	SOX*	f,(r)
CONCURRENT ACKNOWLEDGE	CAK*	f,(r)
INTERRUPT ACKNOWLEDGE	IAK*	f,(r)
DATA INPUT	DIX*	f,(r)
STACK/SPARE INPUT	SIX*	f,(r)
ADD FILE	ADD*	f,LITC(r)
INCREMENT FILE	INC*	f,C(r)
SUBTRACT FILE, TWOS COMPLEMENT	SBT*	f,LTC(r)
SUBTRACT FILE, ONES COMPLEMENT	SBO*	f,LTC(r)
DECREMENT FILE	DEC*	f,C(r)
READ MEMORY, FULL CYCLE	RMF*	f,LID(r)
READ MEMORY, HALF CYCLE	RMH*	f,LID(r)
WRITE MEMORY, FULL CYCLE	WMF*	f,LID(r)
WRITE MEMORY, HALF CYCLE	WMH*	f,LID(r)
COPY	CPY*	f,LITC(r)
ZERO FILE/REG.	ZOF*	f,C(r)
+1 TO FILE/REG.	POF*	f,C(r)
LOGICAL OR WITH FILE	LOR*	f,LFTC(r)
MOVE FILE	MOV*	f,LC(r)
EXCLUSIVE OR WITH FILE	XOR*	f,LFTC(r)
AND WITH FILE	AND*	f,LFTC(r)
SHIFT FILE LEFT	SFL*	f,LC(r)
SHIFT FILE LEFT AND INSERT	SLI*	f,C(r)
SHIFT FILE RIGHT	SFR*	f,LC(r)
SHIFT FILE RIGHT AND INSERT	SRI*	f,C(r)

Class 2 – Mnemonic – File Name, Literal

COMMAND	MNEMONIC	OPERAND FIELD
EXECUTE, LITERAL TYPE	ELT	f,n
LOAD FILE WITH LITERAL	LF	f,n
ADD FILE WITH LITERAL	AF	f,n
TEST IF ZERO	TZ	f,n
TEST NOT ZERO	TN	f,n
COMPARE FILE	CP	f,n

Class 3 – Mnemonic – Literal

COMMAND	MNEMONIC	8 BIT OPERAND
LOAD ZERO CONTROL	LZ	n
LOAD T	LT	n
LOAD M	LM	n
LOAD N	LN	n
JUMP IN 1K	JP	n
LOAD U	LU	n
LOAD SEVEN CONTROL	LS	n
LOAD EIGHT CONTROL	LE	n
RETURN, LOAD T	RLT	n
BANK SELECT	BSL	n

Class 4 – Mnemonic Only – No Operand Field

COMMAND	MNEMONIC
NO OPERATION	NOP
ENABLE COMM. RATES	ECR
DISABLE COMM. RATES	DCR
INPUT COMM. RATES	ICR
RETURN	RTN
SELECT PRIMARY FILE	SPF
RETURN, SELECT PRIMARY FILE	RSP
SELECT SECONDARY FILE	SSF
RETURN, SELECT SECONDARY FILE	RSS
DISABLE EXTERNAL INTERRUPTS	DEI
ENABLE EXTERNAL INTERRUPTS	EEI

COMMAND	MNEMONIC
DISABLE R.T. CLOCK	DRT
ENABLE R.T. CLOCK	ERT
HALT	HLT
MODIFY LOWER COMMAND	MLC
INHIBIT L SAVE	ILS
INCREMENT STACK POINTER	ISP
DECREMENT STACK POINTER	DSP
CLEAR STACK POINTER	CSP
SELECT STACK POINTER	SSP
SELECT STACK UPPER	SSU
SELECT STACK LOWER	SSL
ENTER CONSOLE SWITCHES	ECS

Class 5 – Mnemonic – Literal

COMMAND	MNEMONIC	OPERAND FIELD (12 BIT LITERAL)
JUMP EXTENDED	JE	n

Assembler Instructions – Mnemonic – Operand Field

COMMAND	MNEMONIC	OPERAND FIELD
ORIGIN	ORG	n
DEFINE CONSTANT	DC	n
** EQUATE	EQU	n
EJECT	EJE	
END	END	

** The EQU statement must contain a name field, all other statements may contain a name field.

*	=	File update inhibit
f	=	File Expression
(r)	=	Destination Register Designator
n	=	Literal Expression
L,C,T,F,I,D	=	C-Field Designators

CHAPTER 6

INPUT/OUTPUT

GENERAL DESCRIPTION

The CPU provides an extremely fast elementary input/output capability. The data paths and control functions are simple elements that are sequenced from the control memory with flexible disciplines. The fact that the control memory is very fast, 200 ns/step, means that microprograms (firmware) in the control memory can implement facilities with a high degree of versatility in timing, data paths and I/O capabilities such as priority interrupts, fully buffered data channels, macroprogrammable transfers, and special purpose communication multiplexer channels. This basic I/O element called the "Byte I/O Bus" is described in the following paragraphs. In addition, the direct memory occurs (DMA) and serial data interface are described.

BYTE I/O BUS

The byte I/O facility allows for data transfers over a party-line I/O bus under microprogram control. This I/O facility consists of a byte input bus, a byte output bus, and a three-bit I/O control register.

The contents of the I/O control register define an I/O bus mode. The I/O control register outputs may be decoded to form individual control signals defining the type of transfer being performed on the byte I/O bus and the state of the serial interface output. Of the eight possible states of the I/O control register, one represents no activity on the bus, three are output modes, and four are input modes. One of the output modes removes the MARKing current from the serial interface to output a SPACE to the serial data interface.

The I/O control register is loaded by the control command, an operate-type command with an operation code of 7. When the c field of this instruction equals hexadecimal 8-F, the operations are associated with external input/output, and the three low order bits of c are placed in the I/O control register. The control functions of this instruction are shown in Table 2.

Table 2. Byte I/O Control Modes

Control Command

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

			Hex	Mode	Control Activity
	0 0 0 0		0		No Operation
	0 0 0 1		1		Enter Sense Switches
	0 0 1 0		2		Shift "f" Right Four Places
	0 1 0 0		4		Enter Internal Status
	0 1 1 1		7		Enter Console Switches (0-7)
NO ACTIVITY OUTPUT FUNCTIONS	1 0 0 0		8	0	Clear I/O Mode
	1 0 0 1		9	1	SPARE (*)
	1 0 1 0		A	2	SPARE (*)
	1 0 1 1	I/O CONTROL	B	3	Space Serial Interface
	1 1 0 0		C	4	SPARE (*)
INPUT FUNCTIONS	1 1 0 1		D	5	SPARE (*)
	1 1 1 0		E	6	SPARE (*)
	1 1 1 1		F	7	SPARE

*These functions are used in the MICRO 1600/10 and 1600/20 I/O systems.

To summarize the I/O control modes set by the control command:

Mode	Control Activity	Comments
0	Clear I/O Mode:	The I/O control register is cleared. Data from the designated file or Input bus can be transferred to the designated file register and register (r).
1-7	Set I/O Mode:	The I/O Control register is loaded with the three low order bits of c placing it in one of seven I/O bus or serial interface modes. These modes are described above. Data from the designated file or Input bus can be transferred to a designated file register and register (r).

NOTE: Once an I/O control register mode has been SET, an I/O clear mode must be executed to change the I/O control register mode of operation.

The three output modes and four input modes are distinguished by the high order bit of the I/O control register, which is referred to as the input flag. During execution of some of the operate-type commands, (including the control command) the source of data is the input bus if this input flag is a 1-bit, and the T register otherwise.

Bus Lines

The byte I/O bus consists of

- input data lines
- input control lines
- output data lines
- output control lines

The electrical implementation of the input and output bus lines is shown in Figure 4.

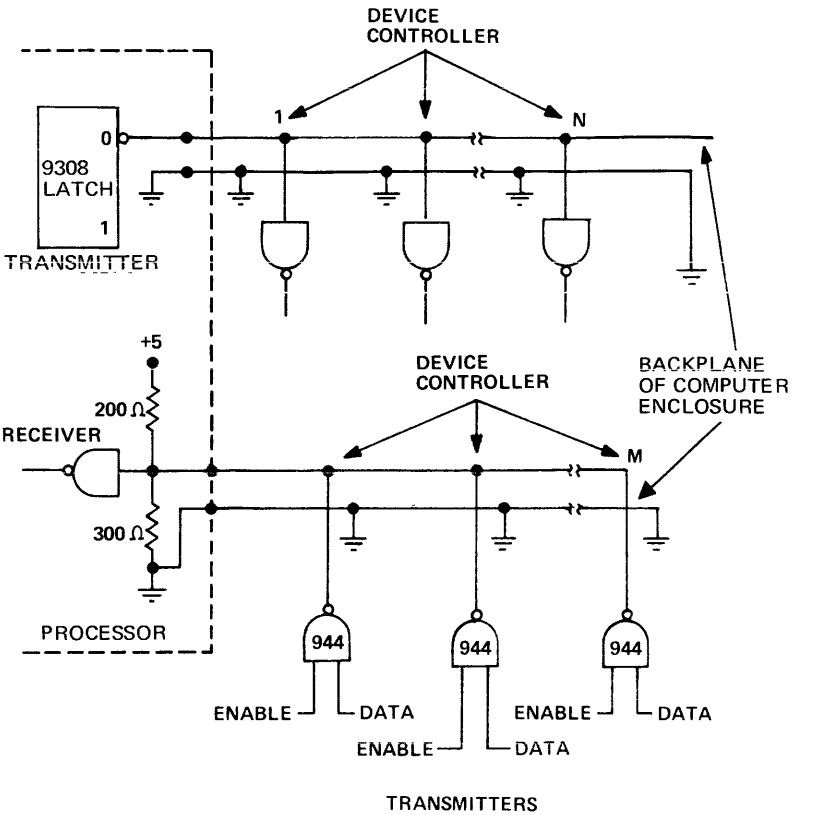


Figure 4. Bus Lines

Input Lines

The input data lines are inputs to the B bus gating. The input control lines are inputs to bits of file register 0. The input lines are ground TRUE signals which are properly terminated at the processor. If the bus is carried out of the basic enclosure it also must be terminated at the remote end. Each peripheral device gates information onto the bus by means of open collector type 944 DTL drive circuits. Up to 15 drivers may be connected to each line.

The logic level on the twisted pairs are:

One – 0 Volts
Zero – +3 Volts

There are three input control lines. They are used in standard 1600/10 and 1600/20 systems for the following purposes:

Typical Input Control Line Definitions
(1600/10 and 1600/20)

Control Lines In	Use in the System
External Interrupt (EINT/):	A peripheral device makes this line low to request an interrupt of the macroprogram. The microprogram must respond with an I/O acknowledge (mode 5) signal. This line is bit 7 of the file register 0 where a 1-bit indicates an external interrupt request.
I/O Reply (ERPY/):	A peripheral device makes this line low in response to an I/O operation when closed-loop operation is required. This line is bit 5 of the file register 0.
I/O Request (ECIO/):	A peripheral device makes this line low in order to request a concurrent data transfer. The microprogram must respond with an I/O acknowledge (mode 5) signal. This line is bit 3 of the file register 0.

The file register 0 bits are defined as follows:

File Register 0 Flags

Bit	Flag
0	– Overflow result Condition
1	– Negative Result Condition
2	– Zero Result Condition
3	– Concurrent I/O Request Line* or (SPARE)
4	– Internal Interrupt
5	– I/O Reply Line* or (SPARE)
6	– Serial Interface
7	– External Interrupt Line* or (SPARE)

*If a standard 1600/10 or 1600/20 CPU interface is not used, these flags may be used as SPARE bits.

Output Lines

The output data lines originate with the FALSE output of the Output Data register. The output control lines originate with the I/O control register. If all peripheral devices on the bus are local to the enclosure, and the bus does not leave the enclosure, then the bus is standard logic levels and no DTL drivers and terminations are used. It may be necessary to repower the signals, if the bus leaves the enclosure. An I/O control board is required in this case to provide output drivers and to decode the control register. The cable length can be up to 30 feet in length, must use twisted pairs and must be terminated at the remote end. Up to 15 receivers can be accommodated. The levels on the twisted pairs are:

One – 0 Volts
Zero – +3 Volts

The I/O control register is decoded within standard 1600/10 and 1600/20 device interface controllers to provide seven control terms:

Typical Byte I/O Control Modes
(1600/10 and 1600/20)

Mode	Control Activity	Term
0	None*	None
1	Control Output	COXX/
2	Data Output	DOXX/
3	Space Serial Interface	SP1X/
4	Interrupt Acknowledge	IACK/
5	Concurrent Acknowledge	CACK/
6	Data Input	DIXX/
7	Spare*	SP2X/

*These terms are used by the L-Save Stack option.

Typical Byte I/O Interface

To illustrate byte I/O programming, a typical interface has been selected which has minimum functions for transferring bytes in and out of the computer. A more complex device, such as a tape controller, or card reader, using the byte I/O function would contain logic similar to this for transferring control, status, and data between the controller and the MICRO 1600.

The byte I/O interface described contains the following basic functions.

- Line receivers and drivers
- Device address decoder
- Function latch and decoder
- Connection latch
- Input multiplexer

- Input selection gates
- Output latches
- Control decoder

These items are shown in block diagram form in Figure 5.

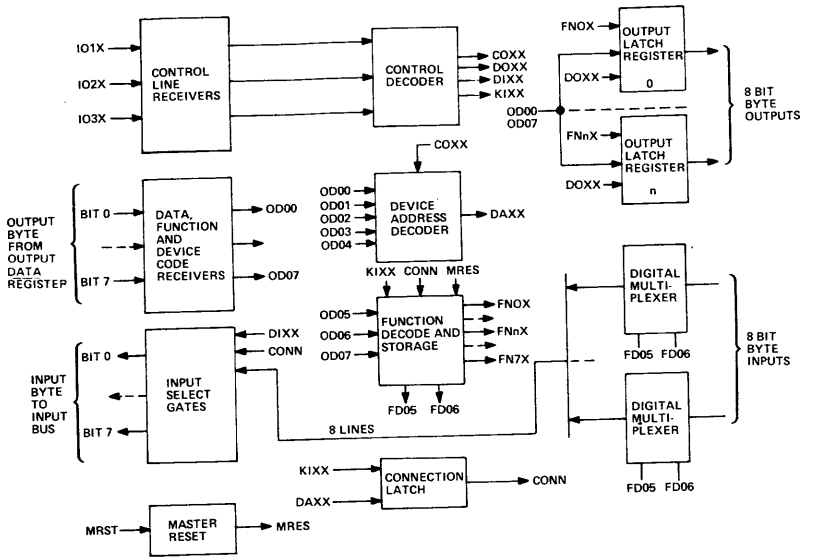


Figure 5. I/O Interface Block Diagram

For summary purposes the logic terms used in the I/O interface example (which are standard for MICRO 1600 interfaces) are defined in Table 3.

Table 3. Definition of Terms in I/O Interface Block Diagram

COXX	FUNCTION AND DEVICE CODE OUTPUT CONTROL PULSE
DOXX	DATA OUTPUT CONTROL PULSE
DIXX	DATA INPUT CONTROL PULSE
KIXX	INTERFACE CLOCK PULSE FORMED BY ORing COXX, DOXX, AND DIXX
DAXX	DETECTED DEVICE ADDRESS ENABLED BY COXX

Table 3. Definition of Terms in I/O Interface Block Diagram (Cont)

OD00-OD07	OUTPUT DATA LINES (COMPLEMENT OUTPUTS OF 1600 OUTPUT DATA FROM MICRO REGISTER)
FN0X-FN7X	LATCHED AND DECODED FUNCTIONS ENABLED BY CONN
FD05-FD06	LATCHED BUT UNDECODED FUNCTION BITS
CONN	CONNECT LATCH INDICATING THAT THE I/O BOARD HAS RECEIVED ITS DEVICE CODE WITH COXX
MRES	MASTER RESET FROM MICRO 1600
101X-103X	3 BITS FROM OUTPUT CONTROL REGISTER
DIG MUX	DIGITAL MULTIPLEXER

Description of Functional Block Diagram (Figure 5)

The control decoder receives the IOnX lines from the control line receivers and first decodes them into COXX, DOXX, and DIXX. These three are ORed to produce KIXX which is used to set and reset function and connect latches.

The device address decoder becomes active whenever the board's address appears on the OD00-OD04 lines. DAXX is active only when COXX is active. Otherwise DAXX would become active every time the device address appeared on the output data lines.

The function latches set or reset every time there is a KIXX pulse. The output functions FN0X, etc., are not enabled unless CONN is active. The functions are used to enable the output latch.

The connection latch is set when the board detects its device address and COXX is active. It is reset on the trailing edge of the next DIXX or DOXX pulse.

The connection latch enables the functions and the input selection gate.

The input selection gates place the input data onto the input bus during DIXX whenever the CONN latch has been set indicating that this board has been addressed.

The output latches are updated to the values on the OD0X lines during DOXX whenever the corresponding function code FNNX is active.

EXAMPLES OF I/O MICROPROGRAMMING

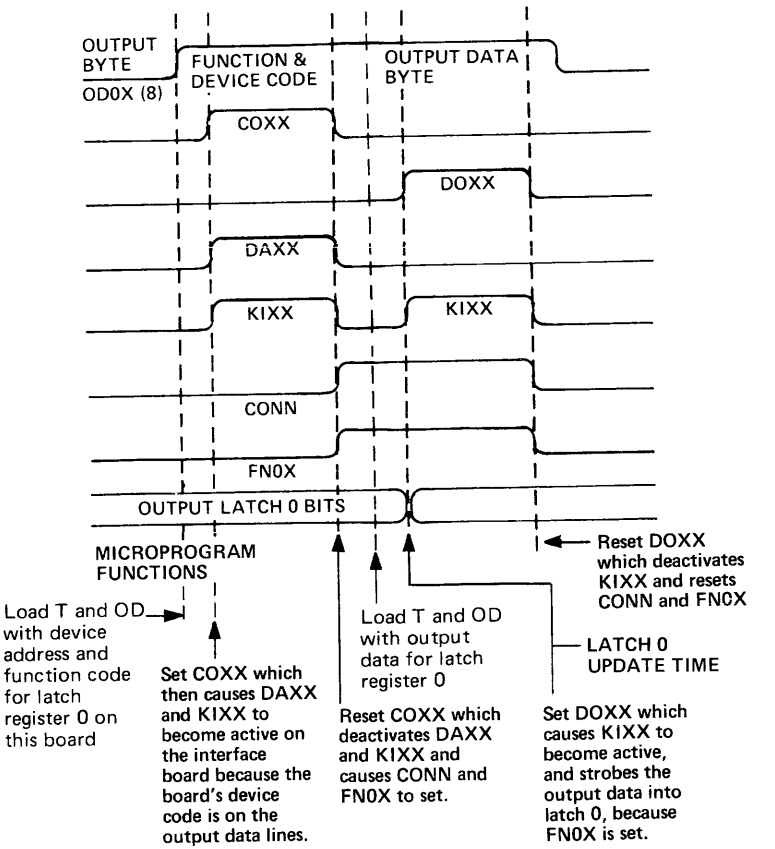
For the following examples assume that the device code is binary 00001. In the standard 1600/10 and 1600/20 the function code sent to a device

interface controller is combined with the device interface's address to form a two-character (hex) device and function code. This results in the following device and function codes:

Function Code		Device and Function Code	
Binary	Hex	Binary	Hex
000	0	000 0 0001	01
001	1	001 0 0001	21
010	2	010 0 0001	41
011	3	011 0 0001	61

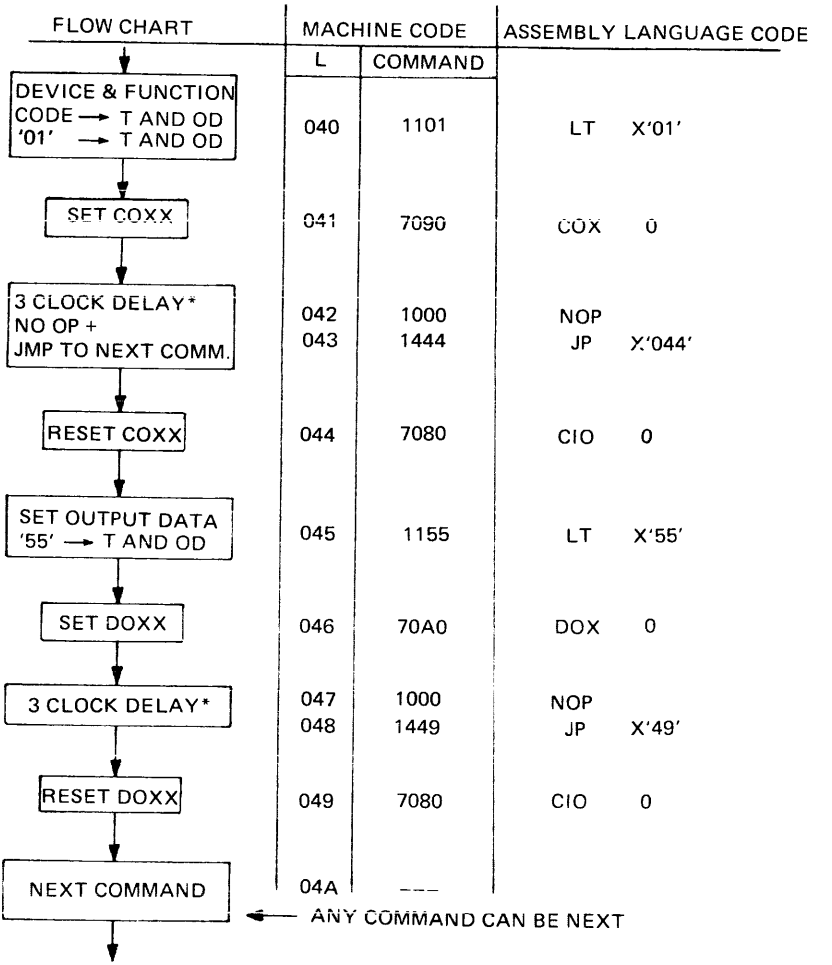
Example 1. For the first Input/Output example the timing of events and the microprogram routine are described for outputting a byte from the MICRO 1600 to latch 0 in the interface board with device code 01.

Timing Diagram:



Microprogram:

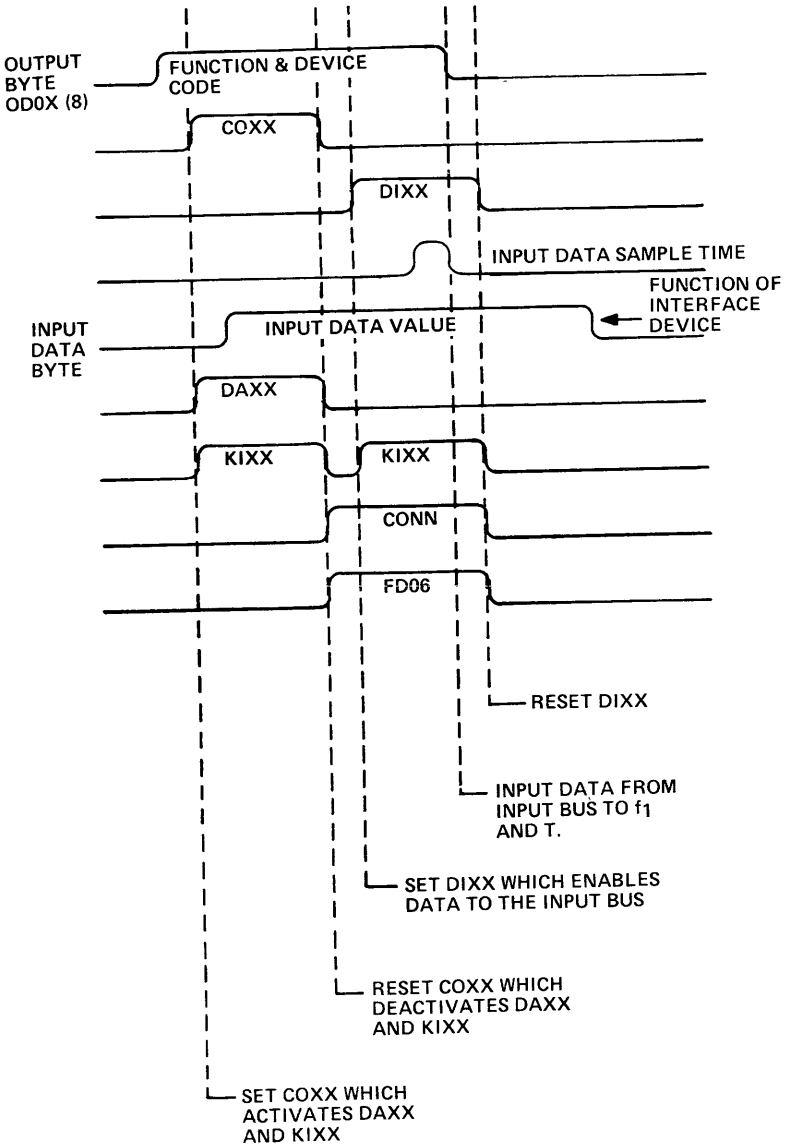
Example 1.



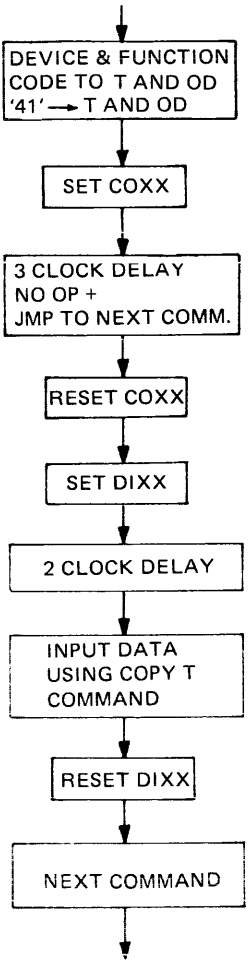
*This is the standard delay in the MICRO 1600/10 or 1600/20 to generate an 800 ns COXX and DOXX. It could be shorter if the interface is in the main computer chassis. Housekeeping can be done on delay clocks.

Example 2. For the second input/output example, the timing of events and the microprogram are described for inputting a byte from input byte 2 of device 01 to file register 1 and T.

Timing Diagram:



Microprogram:



FLOW CHART	MACHINE CODE		ASSEMBLY LANGUAGE CODE	
	L	COMMAND		
DEVICE & FUNCTION CODE TO T AND OD '41' → T AND OD	060	1141	LT	X'41'
SET COXX	061	7090	COX	0
3 CLOCK DELAY NO OP + JMP TO NEXT COMM.	062	1000	NOP	
	063	1464	JP	X'64'
RESET COXX	064	7080	CIO	0
SET DIXX	065	70E0	DIX	0
2 CLOCK DELAY	066	1467	JP	X'67'
INPUT DATA USING COPY T COMMAND	067	B121	CPY	1,T,(T)
RESET DIXX	068	7080	CIO	0
NEXT COMMAND	069	ANY COMMAND CAN BE NEXT		

Example 3. Special Input Function

To achieve minimum input time and still achieve one clock delay after setting DIXX use the following:

DIX	0	Set DIXX
LF	1, X'FF'	Set file 1 = all ones and generate 1 clock delay
CIO	1	Reset DIXX and simultaneously 'and' the input bus with (file 1)

Serial Interface

The processor contains a serial interface capable of communicating with a full duplex teletype. The input from the teletype appears as bit 6 of file register 0 where a 1-bit indicates that the teletype is transmitting a SPACE. The output to the teletype normally transmits a 20 milliampere MARKing current which can be keyed off to send a SPACE signal by placing the I/O control register in mode 3. Character assembly and disassembly, including all timing and synchronization, are performed by microprogramming.

The serial interface is standard. A teletype or CRT wired for 4-wire full duplex 20 milliampere operation may be directly connected to the cable provided with the machine. Other types of serial I/O devices also may use this condition.

Direct Memory Access

The direct memory access (DMA) interface allows for direct connection to the memory address, data and control busses. The DMA may be installed in any slot within the machine enclosure. This board may contain a channel to which a number of peripheral devices are connected, or a device controller which has direct memory access capability. Generally the DMA system will be customized for special applications.

The maximum data transfer rate is 1 million bytes per second. The DMA I/O takes precedence over the processor for memory operations. The DMA must supply its own address control.

Computer Backplane Pin Definitions

The backplane pin connections for the Byte I/O bus, serial data interface and DMA are shown in Figure 6. Connectors J1 and J2 are used for the CPU. Connectors J3-J12 are available for use by memory modules, ROM modules, and device interface controllers. (In an extended backplane option, connectors J3-J17 are available for these purposes.)

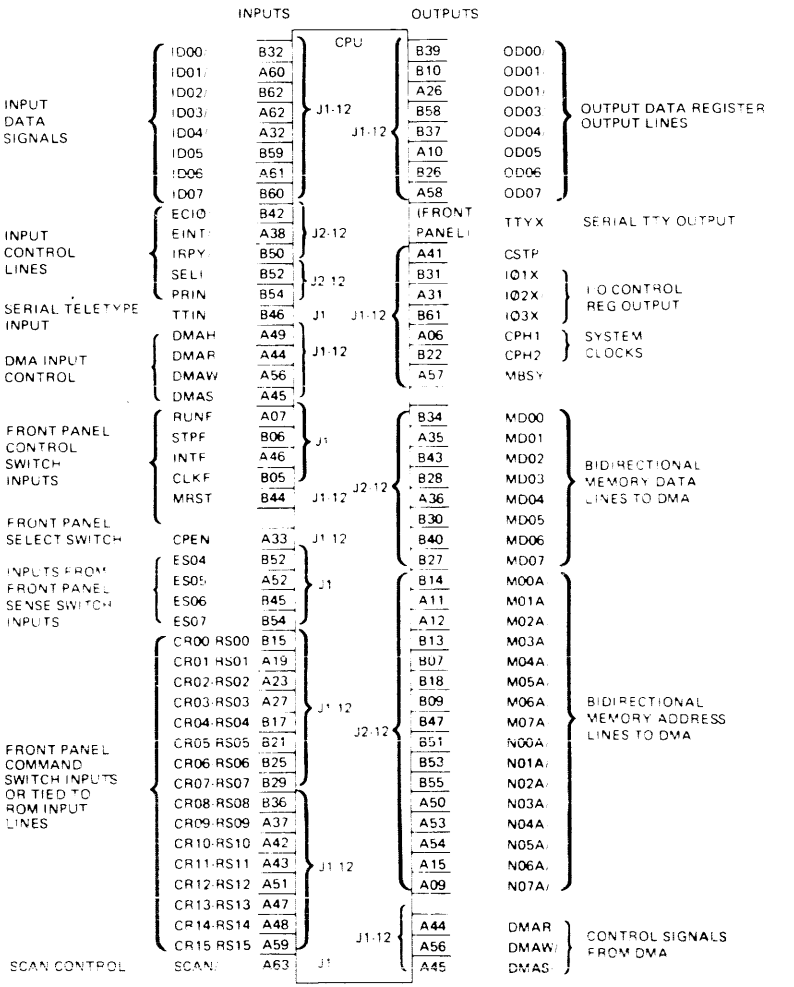


Figure 6. CPU Input/Output Signals

CHAPTER 7

PHYSICAL CHARACTERISTICS AND SYSTEM POWER

MECHANICAL CONFIGURATIONS

Enclosures

A variety of enclosures is available to cover a range of requirements for both OEM and end users.

The basic OEM package provides a simple card cage which may be rack-mounted. Optional rails are available for 24, 26 or 28-inch enclosures. The user may install rear-mounted fans or may use vertical convection cooling as desired.

A rack-mounted package is available which provides wrap-around enclosures with controlled air flow. This package can be specified to go into 24, 26 or 28-inch enclosures.

A distinctive, attractive table-top enclosure is available when appearance is a consideration. When the back panel is removed from this version, power is automatically turned off. In addition, a standard 63-inch-high by 24-inch-deep custom-styled enclosure is available.

Rack-mounted and table-top enclosures are shown in Figures 7, 8, 9 and 10.

Universal Backplanes

The printed-circuit backplanes provide simple point-to-point strapping of all connector pins.

Two versions are available. The first contains 12 connector slots and an integral power supply.

To make larger system configurations possible, a second version contains 16 connector slots and a remote power supply.

Specifications and characteristics of various packaging options are shown in Table 4.

All MICRO 1600 printed circuit boards are equipped with handles which make possible convenient card extraction, insertion and lockdown. All cards are fully accessible from the rear and may be easily extended for trouble-shooting operations. Convenient flat cables are used for connection to I/O devices, with cable routing and strain relief facilities built into the rack-mounted and table-top enclosures.

SYSTEM POWER

Power supplies of the MICRO 1600 are available both integrally or as a remote system and furnish power to all basic and optional component subsystems plugged into printed circuit connectors in the computer enclosures.

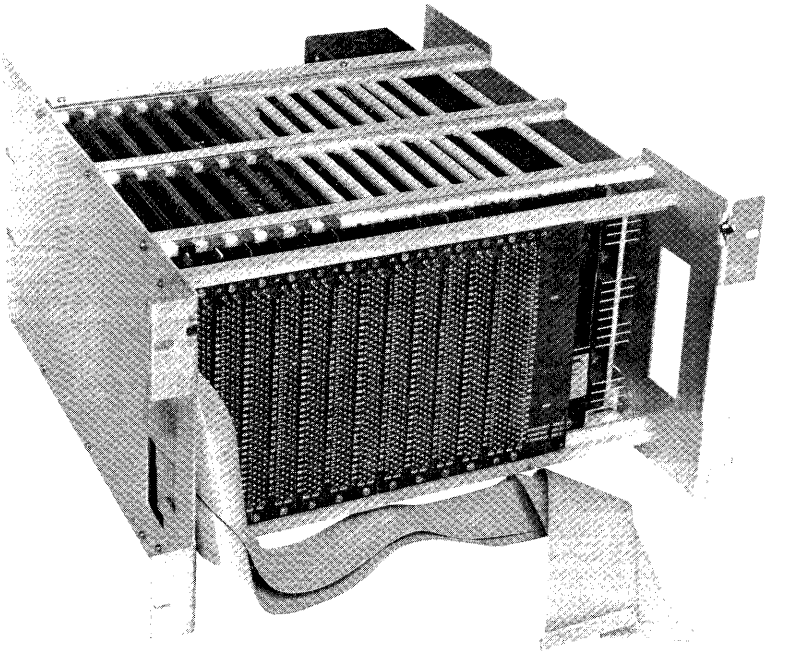


Figure 7. OEM Card Cage

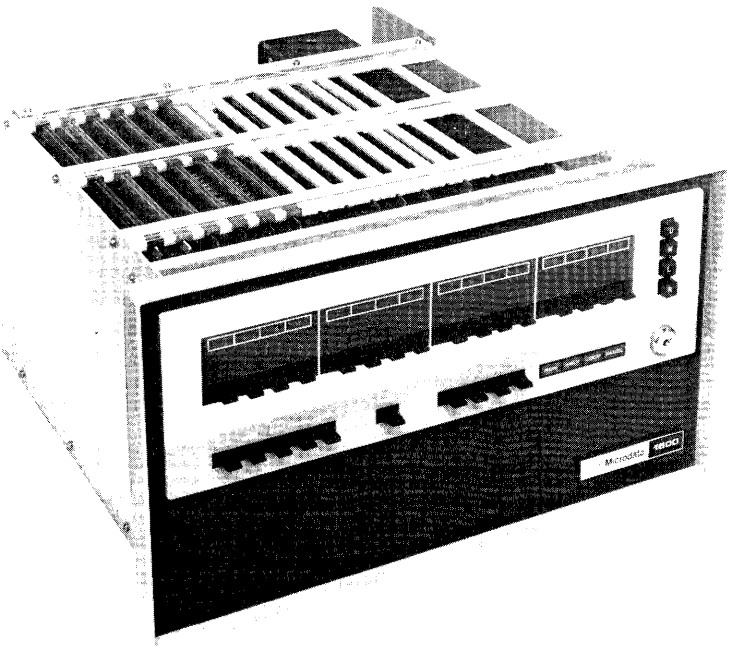


Figure 8. OEM Card Cage with Control Panel

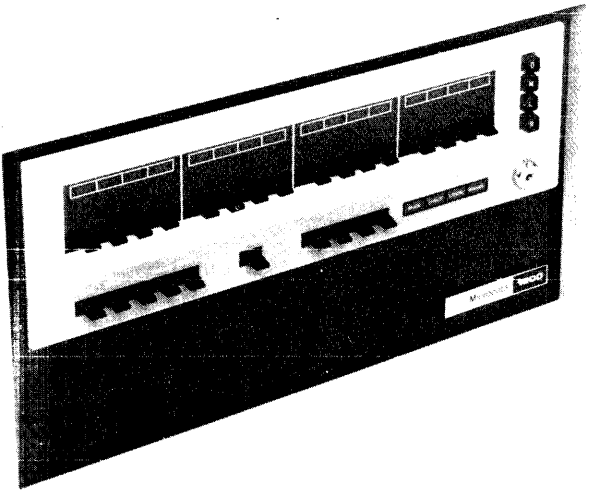


Figure 9. Rack-Mounted Enclosure With Cooling

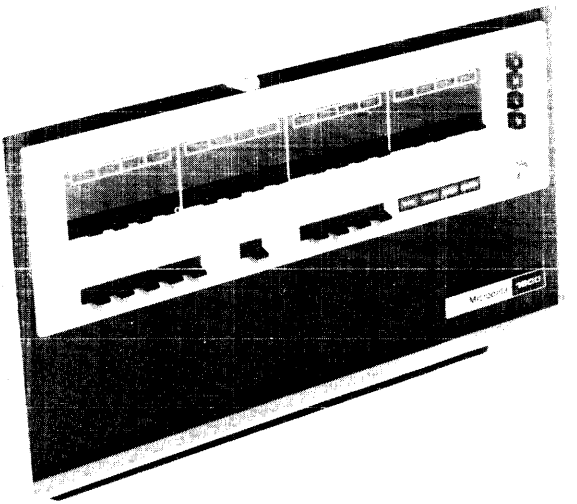


Figure 10. Table Top Enclosure With Internal Power Supply and Cooling. (UL Approval Pending)

Table 4. MICRO 1600 System Specifications

Specifications	Characteristics
PHYSICAL/ ENVIRONMENTAL	
Packaging	Three packages available: 1. OEM Package – basic rack mountable card cage containing all mechanical and electrical facilities except for wrap-around and side panels. 2. Systems Package – Adds wrap-around panels to OEM package for controlled air flow via plenum chambers (side and rear panel intake/exhaust) within the computer cabinet. 3. Table-Top Package – Adds side panels and tilt angle supports to system package for attractive, convenient desk-top installations.
Dimensions	Standard rack mount, 10.5 inch panel height, 20 inch depth. The enclosure provides 12 or 16 connectors which may be used to mount CPU, core and control memory, DMA, and I/O modules. Two slots are dedicated to the CPU.
Power Voltages	115/230VAC, ±10%, 47-63 Hz, 350 watts
Environment	0-50^{o}C (ambient); 10 to 90% relative humidity without condensation.
CIRCUITS	
Type	Integrated LSI, MSI, and SSI circuit design throughout; TTL internal and DTL input/output drivers.
Internal Logic Levels	FALSE = 0v; TRUE = +5v (nominal)
I/O Logic Levels	FALSE = +3v; TRUE = 0v (nominal)

Power is distributed in the system via etched conductors on the printed circuit backplane in the integral system. A power distribution printed circuit module for remote power supply operation also is available.

The power supply is designed with ample spare capacity to power most system configurations which use MSI and LSI circuit modules. User-designed modules may also draw power from the supply if current ratings are not exceeded.

The integral power supply physically mounts in the MICRO 1600 enclosures along the lefthand side (from rear) by sliding into standard card guide assembly (see Figures 11 and 12). A direct plug-in to the backplane via a standard printed circuit connector supplies dc voltages directly to the circuit boards via etched conductors.

The remote power supply is operated by means of an extension cable. It requires no cooling over its rated range of operating temperature.

Summary specifications for both power supplies are given in Table 5. Detailed specifications are available in Microdata Specification No. CS 20001003.

Power Configuration

Peak current consumption for the subsystem components are given in Table 6. A typical system configuration of the MICRO 1600/20 requires the following +5V current:

Subsystem	Current
CPU	5.0 amp
System Console	1.2 amp
768-word bipolar ROM	1.5 amp
4096/8192-byte core memory	0.9 amp
Total	8.6 amp

These current requirements use only three of the available 12 integral power supply card slots. The total +5V capacity is 20 amps, leaving 13.5 amps available for expansion. This is an average of 1.5 amps per connector, which is sufficient to handle a reasonable mix of interface boards and other modules.

For example, assume a fully expanded configuration such as the following:

Subsystem	Card Slots	+5V Current
Basic system	3	8.6
Additional 8192-byte core	1	0.5
Mag. Tape controller	2	1.7
Byte I/O controller	1	1.5
Card reader controller	1	0.8
I/O bus and parallel TTY	1	1.3
DMA	2	1.6
Modem interface	1	1.2
Total	12	17.2

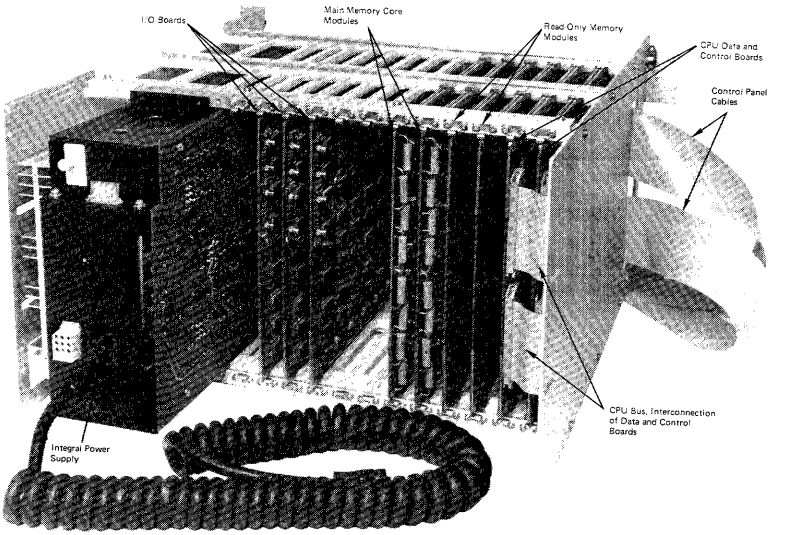


Figure 11. Rear View of Typical MICRO 1600 Card Cage Configuration Showing Integral Power Supply

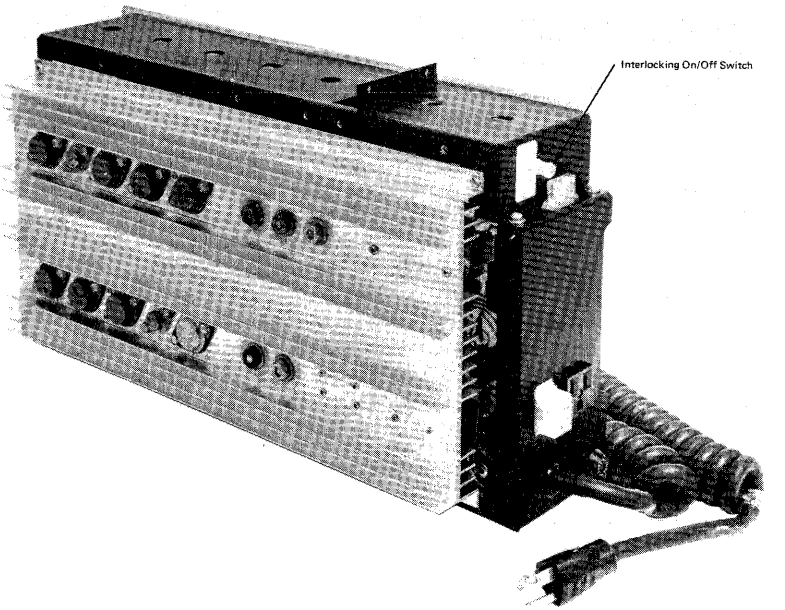


Figure 12. Integral Power Supply of the MICRO 1600

In an extreme case, the system power supply may be unable to handle total current equipments of all modules which could be installed in the available space. In such cases, the user should consult Microdata for the best alternative. Particular care should be used where very large amounts of control memory are to be used since the current is relatively high for these devices, and additional system cooling and increased power may be needed.

Table 5. Power Supply Specifications

Specification	Characteristics
Type	Series regulator dc power supply, 3 regulated outputs, designed for integral or remote operation with Micro 1600 computer.
AC Input	115/230 VRMS ±10%, 47-63 Hz, single phase; 105 VAC tap also provided.
DC Outputs (regulated)	1. +12VDC, 1.5A, ±5%, adjustable 2. +5VDC, 20A, +0.5V adjustable 3. −16.75VDC, 3.5A, ±2% adjustable
Overload/ Overvoltage	1. Overvoltage: 7VDC limit on +5VDC output 2. Overload: current limiting with automatic recovery
Power Fail Detect	Power fail indicator drops 2ms before loss of dc regulation; automatic return after ac returns.
Line Filtering	Input line filter for ac line transient protection integral to supply.
Components	Hermetically sealed semiconductors and ceramic integrated circuits used throughout.
Size	4.8" X 8.6" X 16.3"
Remote (Rack Mounted)	8-3/4" X 19" X 10.5"
Weight	25lbs. (approximate)
Temperature Range	0-50°C

Table 6. MICRO 1600 Power Configuration

Subsystem	+5v	+12v	-16.75v
CPU, Data Board, Control Board	5.0	0	0
System Control Panel	1.2	1.2	0
Bipolar Read-Only Memory (BROM), 256 Words	0.5	0	0
Programmable Read-Only Memory (PROM), 256 Words	0.5	0	0
Alterable Read-Only Memory (AROM), 1024 Words*	7.5	0	0
First Unit, 4096 or 8192-Byte Core Memory Module	0.9	0	1.7
Additional 4096 or 8192-Byte Core Memory Modules	0.5	0	0.1
4 Channel TTY, 20 ma Out	1.5	0	0
4 Channel TTY, 60 ma Out	2.6	0	0
32X32 I/O Expander	2.4	0	0
Mag Tape Controller	1.7	0	0
I/O Bus and Parallel TTY	1.3	0.1	0.1
Priority Interrupt	1.1	0	0
Byte I/O Controller **	1.5	0	0
Card Reader Controller	0.8	0	0
8-Way Async Modem	0.7	0.6	0.4
DMA Selector Channel	1.6	0	0

* Use of separate power distribution for AROM is optional.
** Used for paper tape reader/punch, line printers, cassette I/O.

APPENDIX A. MICROCOMMAND REFERENCE TABLE (NUMERICAL ORDER)

OBJECT BASE	COMMAND	MNEMONIC	CLASS	PAGE
0000	EXECUTE, LITERAL TYPE	ELT	2	43
0000	EXECUTE, OPERATE TYPE	EOT	1	43
0000	JUMP EXTENDED	JE	5	15
1000	LOAD ZERO CONTROL	LZ	3	15
1000	NO OPERATION	NOP	4	16
1001	ENABLE COMM. RATES	ECR	4	16
1002	DISABLE COMM. RATES	DCR	4	17
1004	INPUT COMM. RATES	ICR	4	17
1020	RETURN	RTN	4	17
1040	SELECT PRIMARY FILE	SPF	4	17
1060	RETURN, SELECT PRIMARY FILE	RSP	4	18
1080	SELECT SECONDARY FILE	SSF	4	18
10A0	RETURN, SELECT SECONDARY FILE	RSS	4	18
1100	LOAD T	LT	3	18
1200	LOAD M	LM	3	18
1300	LOAD N	LN	3	19
1400	JUMP IN 1K	JP	3	19
1600	LOAD U	LU	3	19
1700	LOAD SEVEN CONTROL	LS	3	20
1704	DISABLE EXTERNAL INTERRUPTS	DEI	4	20
1708	ENABLE EXTERNAL INTERRUPTS	EEI	4	20
1710	DISABLE R.T. CLOCK	DRT	4	20
1720	ENABLE R.T. CLOCK	ERT	4	21
1780	HALT	HLT	4	21
1800	LOAD EIGHT CONTROL	LE	3	21
1900	RETURN, LOAD T	RLT	3	21
1A00	MODIFY LOWER COMMAND	MLC	4	21
1B00	INHIBIT L SAVE	ILS	4	22
1B01	INCREMENT STACK POINTER	ISP	4	22
1B02	DECREMENT STACK POINTER	DSP	4	22
1B04	CLEAR STACK POINTER	CSP	4	23
1B08	BANK SELECT	BSL	3	23
1B90	SELECT STACK POINTER	SSP	4	23
1BA0	SELECT STACK UPPER	SSU	4	24
1BC0	SELECT STACK LOWER	SSL	4	24

OBJECT BASE	COMMAND	MNEMONIC	CLASS	PAGE
2000	LOAD FILE WITH LITERAL	LF	2	24
3000	ADD FILE WITH LITERAL	AF	2	24
4000	TEST IF ZERO	TZ	2	25
5000	TEST NOT ZERO	TN	2	25
6000	COMPARE FILE	CP	2	25
7010	ENTER SENSE SWITCHES	ESS*	1	26
7020	SHIFT FILE RIGHT 4	SRF*	1	26
7040	ENTER INTERNAL STATUS	EIS*	1	26
7070	ENTER CONSOLE SWITCHES	ECS	4	27
7080	CLEAR I/O MODE	CIO*	1	28
7090	CONTROL OUTPUT	COX*	1	28
70A0	DATA OUTPUT	DOX*	1	28
70B0	SPARE OUTPUT	SOX*	1	29
70C0	CONCURRENT ACKNOWLEDGE	CAK*	1	29
70D0	INTERRUPT ACKNOWLEDGE	IAK*	1	29
70E0	DATA INPUT	DIX*	1	29
70F0	STACK/SPARE INPUT	SIX*	1	30
8000	ADD FILE	ADD*	1	31
8040	INCREMENT FILE	INC*	1	31
9000	SUBTRACT FILE, TWOS COMPLEMENT	SBT*	1	32
9040	SUBTRACT FILE, ONES COMPLEMENT	SBO*	1	32
9040	DECREMENT FILE	DEC*	1	33
A000	READ MEMORY, FULL CYCLE	RMF*	1	33
A020	READ MEMORY, HALF CYCLE	RMH*	1	34
A010	WRITE MEMORY, FULL CYCLE	WMF*	1	35
A030	WRITE MEMORY, HALF CYCLE	WMH*	1	36
B000	COPY	CPY*	1	37
B000	ZERO FILE/REG.	ZOF*	1	37
B040	+1 TO FILE/REG.	POF*	1	38
C000	LOGICAL OR WITH FILE	LOR*	1	38
C000	MOVE FILE	MOV*	1	39
D000	EXCLUSIVE OR WITH FILE	XOR*	1	39
E000	AND WITH FILE	AND*	1	40
F000	SHIFT FILE LEFT	SFL*	1	40
F040	SHIFT FILE LEFT AND INSERT	SLI*	1	41
F020	SHIFT FILE RIGHT	SFR*	1	41
F060	SHIFT FILE RIGHT AND INSERT	SRI*	1	42

Microdata

Microdata Corporation
644 East Young Street
Santa Ana, California 92705
(714) 540-6730 TWX: 910-595-1764

Printed in U.S.A. 10m 11/71