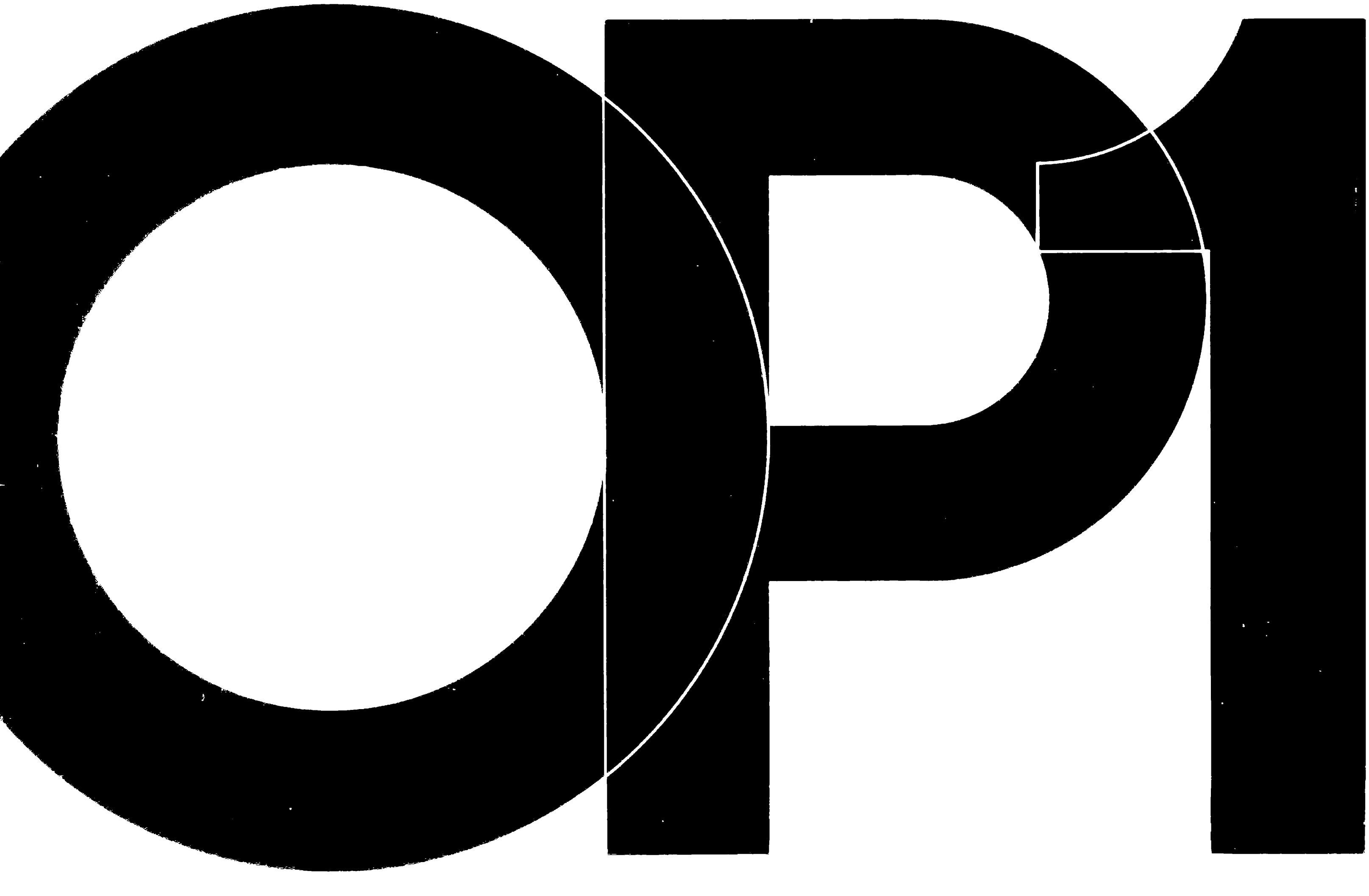
Ontel OP-1/R Reference Manual



OP-1/R

OP-1/R

REFERENCE MANUAL

ONTEL CORPORATION
250 Crossways Park Drive
Woodbury, N.Y. 11797

June 12, 1978

REFERENCE MANUAL

TABLE OF CONTENTS

APPENDIX

SECTION 1R

INTRODUCTION

The Ontel OP-1/R is a user programmable intelligent terminal system designed to function as an on-line system to a host OP-1 system or host computer. Programs can be loaded from a remote computer or stored in an integral Read Only Memory.

The system contains two microprocessors combined with random access read/write memory to create a low cost multiprocessor system with exceptionally high speed performance display capabilities.

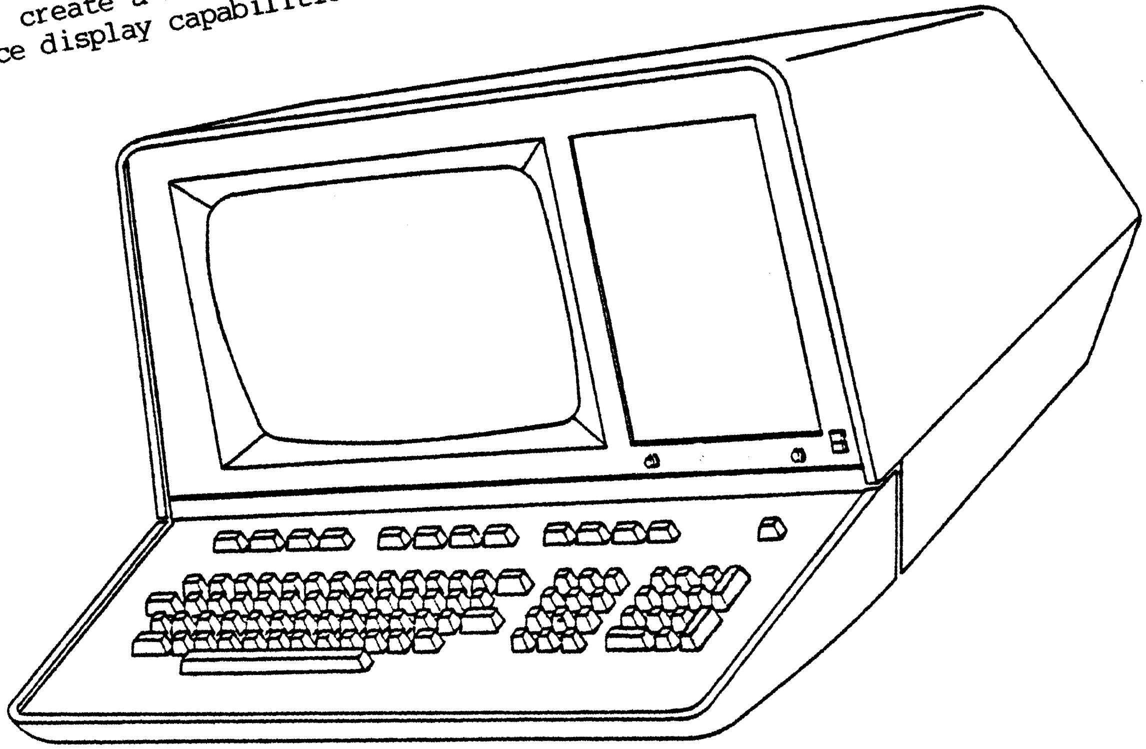


Figure 1R-1. OP-1/R Configuration

The Central Processor Unit can be programmed by the user for any application. The input/output disciplines are program controlled and enable the system to operate with various host computers.

The Display Microprocessor provides a movable window in memory and performs fast roll/scroll and erase operations, as well as display functions such as blinking and video reversal.

The system features include random access memory available in various configurations; program controlled asynchronous communications up to 19200 bits-per-second; a 14 inch non-glare CRT; and a complete programmable keyboard, arranged in four functional sections that generates unique codes readable by the CPU.

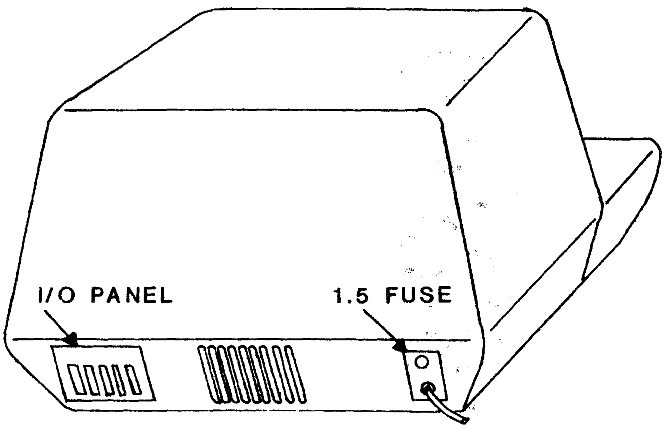


Figure 1R-2. OP-1/R Rear View

There are four versions of the OP-1/R. The OP-1/R Standard, OP-1/R II, OP-1/RW and the OP-1/RS. See Figure 1R-3 for summary.

All Boards include:

a) 8085 CPU
b) 4K, 8K, 16K, or 32K DYNAMIC RAM
c) Up to four 2708, 2716 or 2732 ROM, PROM or EROM
d) Parallel I/O
e) Two Fixed Data Switches
f) Support for Alternate I/O Adapter (Optional second board).

FEATURES

OP-1/R STANDARD (5100-1101)

- 24 Line Display
- Async Communications - RS-232, Current Loop or 2 Wire Direct. Programmable up to 19.2K BAUD.
- Will not support IOM and Device Controller.

OP-1/R II (5100-1108)

- Same as Standard but will support IOM and Device Controller.

OP-1/RW (5100-1109)

- Async Communications - RS-232, 2 Wire Direct. Programmable up to 19.2K BAUD.
- Word Processing Display.
- Support IOM and Device Controller.

OP-1/RS (5100-1110)

- Synchronous Communications - RS-232. Programmable up to 19.2K BAUD.
- 24 Line Display
- Support IOM and Device Controller.

IOM AND CONTROLLERS

The following controllers can be used on the OP-1/R. Refer to OP-1 Reference Manual.

Word Move Controller I	5000-1170
Word Move Controller II	-11101
Synchronous Communications I	-1104
Synchronous Communications II	-1193
Bisynchronous Communications I	-1148
Bisynchronous Communication II	-1192
Asynchronous Communications	-1134
Diskette Controller	-1135
MPDC II Controller	-11110
Mini Diskette Controller	-11118
SDLC	-11106

These PC Boards are identical to OP-1 Device Controllers except for the additional connection of signal $\overline{\text{SEL}}$ to I/O Pin 4.

FEATURE	REF. MANUAL SECTION	OP-1/R STD 5100-2001	OP-1/R II 5100-2008	OP-1/RW 5100-2009	OP-1/RS 5100-2010
INTERRUPTS	4R	0-3	0-7	0-7	0-7
MEMORY RAM 4K,8K,16K, or 32K	R1	X	X	X	X
MEMORY ROM (4) 1K,2K or 4K	R1	X	X	X	X
DISPLAY 24 Line	9R	X	X		X
Word Processing	11R			X	
COMMUNICATIONS Asynchronous	7R	X	X	X	
Synchronous	12R				X
RS-232		X	X	X	X
Current Loop		X	X		
2 Wire Direct		X	X	X	
FIXED DATA SWITCHES	5R	X	X	X	X
PARALLEL I/O	8R	X	X	X	X
OPTION BOARDS Alt.I/O Adapter	10R	X	X	X	X
IOM	13R		X	X	X

Figure 1R-3

SECTION 2R

SYSTEM ARCHITECTURE

The OP-1/R system is designed for high speed communications and interactive display capability. The system memory is shared by two or three microprocessors:

1. Central Processor Unit
2. Display Microprocessor
3. Optional Input/Output Microprocessor

Figure 2R-1 illustrates the system architecture.

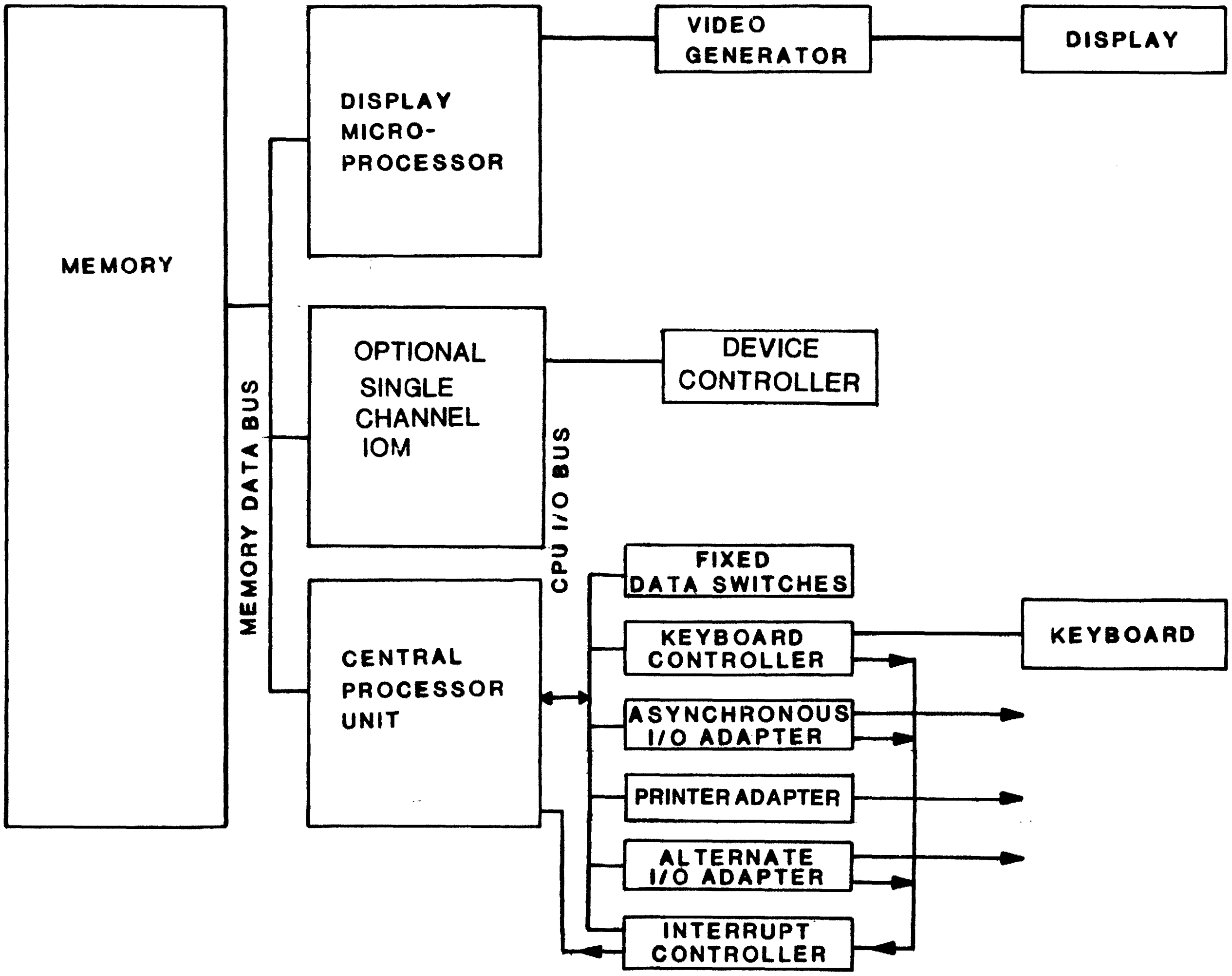


Figure 2R-1. OP-1/R System Block Diagram

MAJOR SYSTEM ELEMENTS

MEMORY

Random access 8-bit byte memory is used. Various combinations of Read/Write or Read Only Memory are available. Bootstrap memory is available for systems equipped with only Read/Write main memory.

The memory can be allocated to any use at the programmer's option for program storage, display or I/O buffers.

CENTRAL PROCESSOR UNIT

The Central Processor Unit (CPU) performs the control, arithmetic and logic functions of the OP-1/R. An 8-bit parallel microprocessor with interrupt capability is used as the Central Processor Unit. A bi-directional data bus is used for communications between the CPU, memory and I/O devices.

The instruction repertoire includes Arithmetic/Logic Instructions, Load, Increment/Decrement, Rotate, Jump, Call and Return instructions.

DISPLAY MICROPROCESSOR

The Display Microprocessor converts the OP-1/R memory into a continous display page. The display screen is a movable window in the page. Any section of the memory can be assigned as a display buffer. Over 400 lines with 80 characters each can be implemented in a 32K byte memory system.

A detailed description of each device controller and device is supplied in the appropriate section.

INPUT/OUTPUT MICROPROCESSOR

Refer to Section 13.

SECTION 3R

CENTRAL PROCESSOR UNIT AND I/O BUS

This section describes the CPU and the Instruction Repertoire. The basic devices directly connected to the CPU I/O bus: Fixed Data Switches, Keyboard and Asynchronous I/O Adapter, Printer Adapter and Alternate I/O Adapter are described in later sections.

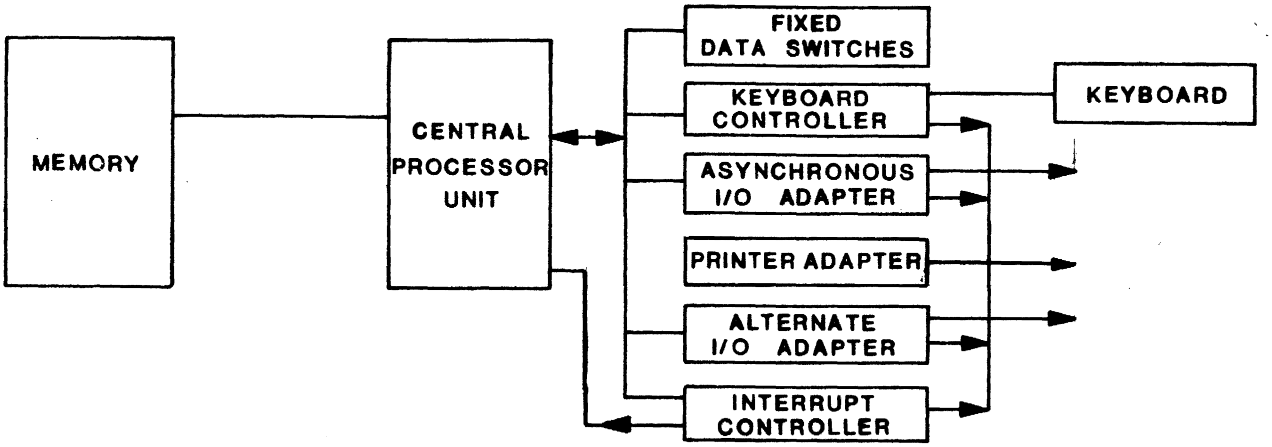


Figure 3R-1. CPU and I/O Bus

CENTRAL PROCESSOR UNIT

The CPU consists of an Arithmetic/Logic Unit, five condition flags, seven general purpose 8-bit registers, and a pushdown stack pointer and a program counter, each 16 bits long. The CPU is capable of directly addressing up to 64K bytes of main memory.

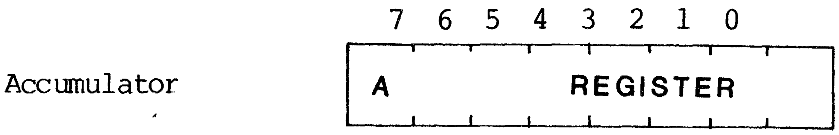
ARITHMETIC/LOGIC UNIT

The Arithmetic/Logic Unit is an 8-bit parallel binary computation device that performs addition, subtraction and logical operations.

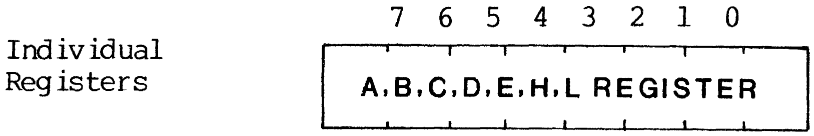
All individual register arithmetic and logical operations are carried out between the A Register (Accumulator) and any one of the seven general purpose registers or between the A Register and memory. Register pair addition operations are carried out between the H and L registers and any one of the four register pairs.

GENERAL PURPOSE REGISTERS

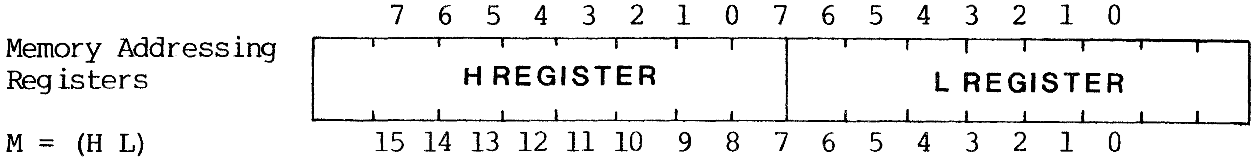
Seven general purpose registers are used for temporary data storage internal to the CPU:



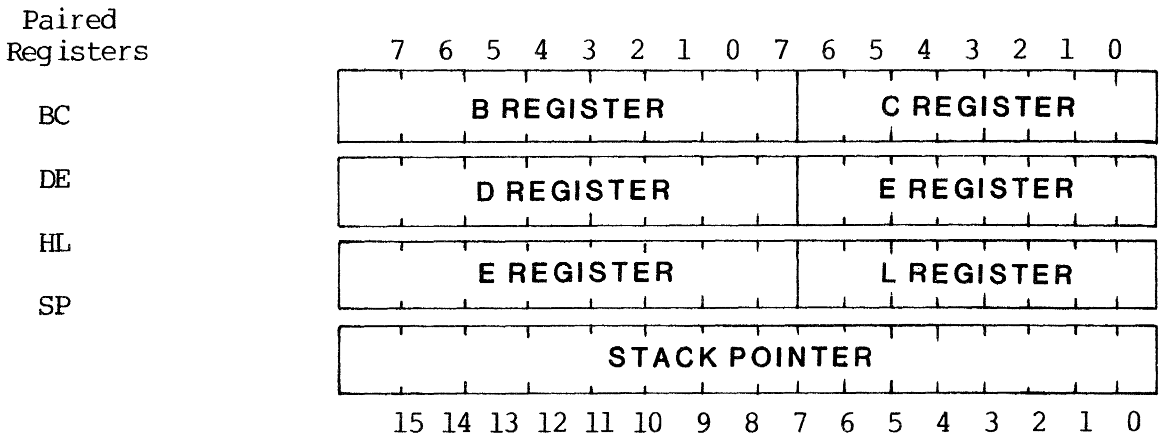
The A Register (Accumulator) receives the result of individual register arithmetic, logical and rotate operations. The A Register is also used as the Input/Output Register for data and control information exchanged between the CPU and the I/O Devices.



The A, B, C, D, E, H, and L Registers can be used in conjunction with the Register for individual register arithmetic and logical operations. All registers are independent and can be incremented, decremented or loaded from another register or from memory.



The H and L Registers, besides being used individually, are also used to provide memory addressing capability. The L Register contains the eight lower order address bits and H Register the eight higher order address bits of the memory location referenced. The contents of memory pointed to by the H and L registers are denoted by the letter M.



The individual registers can be concatenated in pairs to form a 16-bit register pair. The pair can be used to address memory or can be added to the HL pair. The standard names for the pairs are shown above.

THE STACK

A stack is an area of memory allocated for subroutine or interrupt linkage or for temporary storage. Various data bytes may be "pushed" onto the stack in sequential order and later "popped" or retrieved from the stack in reverse order. To keep track of the last byte pushed to the stack, a stack pointer is provided. The stack pointer (SP) is a 16-bit register which always stores the address of the 1st byte in the stack. As illustrated in Figure 3R-3, a stack starts at its initial location and expands linearly toward lower addresses as items are pushed to the stack. It is the programmer's responsibility to initiate the stack pointer register and reserve enough room for stacking purposes so that pushing data to the stack never destroys other data stored in memory. Any portion of the memory can be allocated for stack purposes.

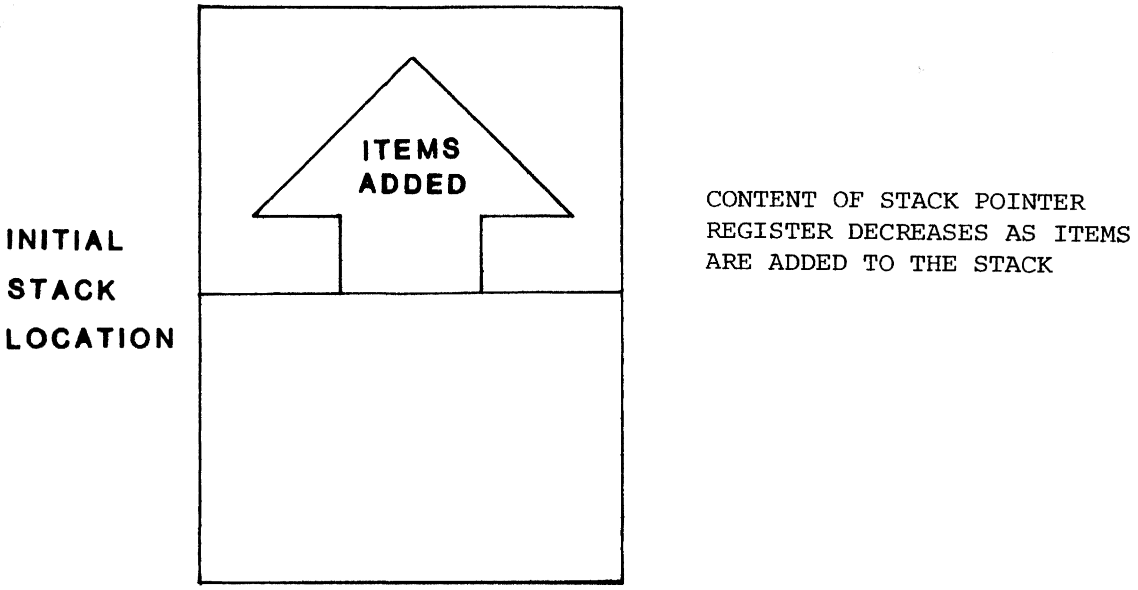


Figure 3R-3. The Stack

BOOTSTRAP MEMORY

When power is turned on, or the PROG key is depressed in conjunction with the SHIFT and CTRL keys, the CPU will execute the program starting at location 8000 of the memory. The OP-1/R can be equipped with a bootstrap memory that can be factory programmed to load a program from a communication line. Alternately, the OP-1/R can be equipped with a fixed program stored in a Read Only Memory (ROM).

THE OP-1R INSTRUCTION SET

The OP-1R instruction set includes six different types of instructions:

- **Data Transfer Group** - move data between registers or between memory and registers.
- **Arithmetic Group** - add, subtract, increment or decrement data in registers or in memory.
- **Logical Group** - AND, OR, EXCLUSIVE-OR, compare, rotate or complement data in registers or in memory.
- **Branch Group** - conditional and unconditional jump instructions, subroutine call instructions and return instructions.
- **Stack and Machine Control Group** - intructions for maintaining the stack and internal control flags.
- **Input/Output Group** - instructions to select, input from or output to external devices.

Instruction and Data Formats:

Memory for the OP-1R is organized into 8-bit quantities, called Bytes. Each byte has a unique 16-bit binary address corresponding to its sequential position in memory.

The OP-1R can directly address up to 65,536 bytes of memory, which may consist of both read-only memory (ROM) elements and random-access memory (RAM) elements (read/write memory).

Data in the OP-1R is stored in the form of 8-bit binary integers:

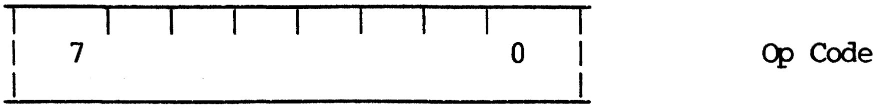
```
          DATA WORD (byte)

 ________________________________
 |   |   |   |   |   |   |   |   |
 |  7   6   5   4   3   2   1   0  |
 |_________________________________|
  MSB                           LSB
```

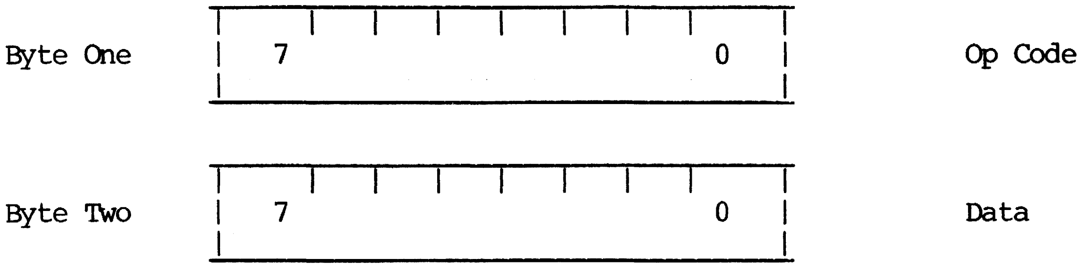
When a register or data word byte contains a binary number, it is necessary to establish the order in which the bits of the number are written. In the OP-1R, BIT 0 is referred to as the Least Significant Bit (LSB), and BIT 7 (of an 8 bit number) is referred to as the Most Significant Bit (MSB).

The OP-1R program instructions may be one, two or three bytes in length. Multiple byte instructions must be stored in successive memory locations; the address of the first byte is always used as the address of the instruction. The exact instruction format will depend on the particular operation to be executed.

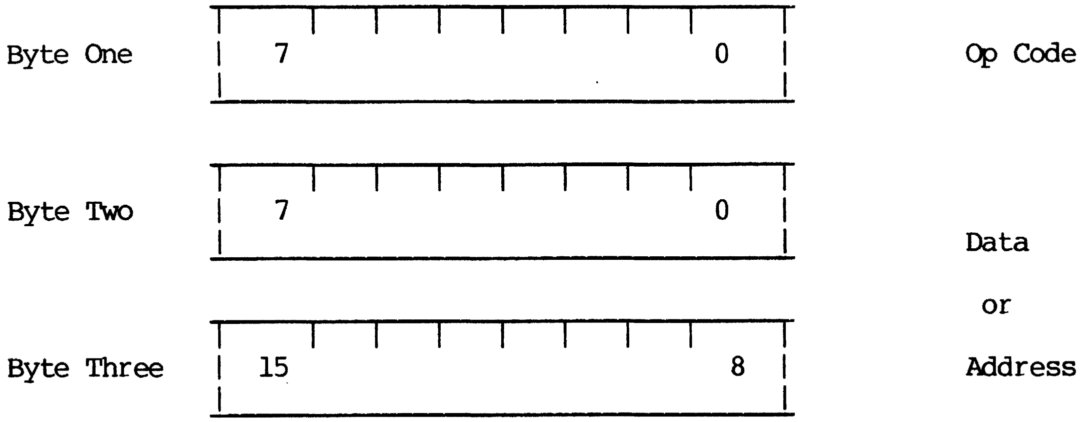
SINGLE BYTE INSTRUCTIONS



TWO-BYTE INSTRUCTIONS



THREE-BYTE INSTRUCTIONS



Addressing Modes

Often the data that is to be operated on is stored in memory. When multi-byte numeric data is used, the data, like instructions, is stored in successive memory locations, with the least significant byte first, followed by increasingly significant bytes. The OP-1R has four different modes for addressing data stored in memory or in registers:

- * Direct - Bytes 2 and 3 of the instruction contain the exact memory address of the data item (the low-order bits of the address are in byte 2, the high order bits in byte 3).
- * Register - The instruction specifies the register or register pair in which the data is located.
- * Register Indirect - The instruction specifies a register pair which contains the memory address where the data is located (the high order bits of the address are in the first register of the pair, the low order bits in the second.)
- * Immediate - The one or two bytes of data operated on by the instruction immediately follow the instruction in memory.

Unless directed by an interrupt or branch instruction, the execution of instructions proceeds through consecutively increasing memory locations. A branch instruction can specify the address of the next instruction to be executed in one of two ways:

- * Direct - The branch instruction contains the address of the next instruction to be executed. (Except for the 'RST' instruction, byte 2 contains the low order address and byte 3 contains the high order address.)
- * Register Indirect - The branch instruction indicates a register pair which contains the address of the next instruction to be executed. (The high-order bits of the address are in the first register of the pair, the low order bits in the second.)

The RST instruction is a special one-byte call instruction (usually used during interrupt sequences). RST includes a three bit field; program control is transferred to the instruction whose address is eight times the contents of this three bit field.

Condition Flags

There are five condition flags associated with the execution of instructions on the OP-1R. They are Zero, Sign, Parity, Carry, and Auxiliary Carry, and are each represented by a 1 bit register in the CPU. A flag is set by forcing the bit to 1; reset by forcing the bit to 0.

Zero: If the result of an instruction execution has the value 0, this flag is set; otherwise it is reset.

Sign: If the most significant bit of the result of an instruction execution has the value 1, this flag is set; otherwise it is reset.

Parity: If the modulo 2 sum of the bits of the result of an instruction execution is 0, (i.e., if the result has even parity), this flag is set; otherwise it is reset (i.e., the result has odd parity).

Carry: If the instruction execution resulted in a carry (from addition), or a borrow (from subtraction or a comparison) out of the high order bit, this flag is set; otherwise it is reset.

Auxiliary Carry: If the instruction execution caused a carry out of bit 3 and into bit 4 of the resulting value, the auxiliary carry is set; otherwise it is reset. This flag is affected by single precision additions, subtractions, increments, decrements, comparisons, and logical operations, but is principally used with additions and increments preceding a DAA (Decimal Adjust Accumulator) instruction.

Symbols and Abbreviations

The following symbols and abbreviations are used in the subsequent description of the OP-1R instructions:

SYMBOLS	**MEANING**
accumulator	Register A
addr	16-bit address quantity
addr byte	low or high order byte of address, as indicated
data	8-bit data quantity

Symbols and Abbreviations - (CONTINUED)

SYMBOLS **MEANING**

data 16 16 bit data quantity

byte 2 The second byte of the instruction

byte 3 The third byte of the instruction

r,rd,rs One of the registers A,B,C,D,E,H,L

DDD,SSS The bit pattern designating one of the registers A,B,C,D,E,H,L (DDD=destination, SSS=source):

DDD or SSS	REGISTER NAME
111	A
000	B
001	C
010	D
011	E
100	H
101	L

rp One of the register pairs:

B represents the B,C pair with B as the high order register and C as the low order register;

D represents the D,E pair with D as the high order register and E as the low order register;

H represents the H,L pair with H as the high order register and L as the low order register;

SP represents the 16 bit stack pointer register.

RP The opcode bits corresponding to a register pair, as follows:

B - 00
D - 01
H - 10
SP - 11

SYMBOLS	MEANING
rh	The first (high order) register of a designated register pair.
rl	The second (low order) register of a designated register pair.
PC	16 bit program counter register (PCH and PCL are used to refer to the high order and low order 8 bits respectively).
SP	16 bit stack pointer register (SPH and SPL are used to refer to the high order and low order 8 bits respectively).
r_m	Bit m of the register r (bits are number 7 through 0 from left to right).
Z,S,P,CY,AC	The condition flags: Z - Zero, S - Sign, P - Parity, CY - Carry, AC - Auxiliary Carry
( )	The contents of the memory location or registers enclosed in the parentheses.
<—	"Is transferred to"
AND	Logical AND
XOR	Exclusive OR
OR	Inclusive OR
*	Multiplication
+	Addition
-	Two's complement subtraction
<—>	"Is exchanged with"
~	The one's complement (e.g., ~(A))
n	The restart number 0 through 7
NNN	The binary representation 000 though 111 for restart number 0 through 7 respectively.
W	(extra) Wait state for memory access

Description Format:

The following pages provide a detailed description of the instruction set of the OP-1R. Each instruction is described in the following manner.

1. The OP-1 assembler format, consisting of the instruction mnemonic and operand fields, is printed in **BOLDFACE** on the left side of the first line.

2. The name of the instruction is enclosed in parentheses on the right side of the first line.

3. The next line(s) contains a symbolic description of the operation of the instruction.

4. This is followed by a narrative description of the operation of the instruction.

5. The following line(s) contain the binary fields and patterns that comprise the machine instruction.

6. The last four lines contain incidental information about the execution of the instruction. The number of machine cycles and states required to execute the instruction are listed first. If the instruction has two possible execution times, as in a conditional Jump, both times will be listed, separately by a slash. Next, any significant data addressing modes (see Page 3R4) are listed. The last line lists any of the five Flags that are affected by the execution of the instruction.

Approximate Timing

The number of microseconds per instruction can be approximated as follows: # microseconds = (# CPU states)*(.32)

DATA TRANSFER GROUP:

This group of instructions transfers data to and from registers and memory. Condition flags are not affected by any instruction in this group.

MOV rd, rs (Move Register)

(rd) <— (rs)

The contents of register rs is copied to register rd.

0	1	D	D	D	S	S	S

Mem. Cycles:	1
CPU States:	5 (4+1 WAIT (W))
Addressing:	register
Flags:	none

MOV r, M (Move from memory)

(rl) <— ((H) (L))

The contents of the memory location, whose address is in registers H and L, is copied to register r.

0	1	D	D	D	1	1	0

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	reg. indirect
Flags:	none

MOV M,r (Move to memory)

((H) (L)) <— (r)

The content of register r is copied to the memory location whose address is in registers H and L.

0	1	1	1	0	S	S	S

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	reg. indirect
Flags:	none

MVI r, data (Move Immediate)

(r) <— (byte 2)

The content of byte 2 of the instruction is copied to register r.

<table>
<tr><td>0</td><td>0</td><td>D</td><td>D</td><td>D</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">data</td></tr>
</table>

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	immediate
Flags:	none

MVI M, data (Move to memory immediate)

((H) (L)) <— (byte 2)

The content of byte 2 of the instruction is copied to the memory location whose address is in the register pair HL.

<table>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">data byte</td></tr>
</table>

Mem. Cycles:	3
CPU States:	13 (10+3W)
Addressing:	immed./reg. indirect
Flags:	none

LXI rp, data 16 (Load register pair immediate)

(rh) <— (byte 3),

(rl) <— (byte 2)

Byte 3 of the instruction is copied into the high order register (rh) of the register pair rp. Byte 2 of the instruction is copied into the low order register (rl) of the register pair rp.

<table>
<tr><td>0</td><td>0</td><td>R</td><td>P</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">data byte (-> rl)</td></tr>
<tr><td colspan="8">data byte (-> rh)</td></tr>
</table>

Mem. Cycles:	3
CPU States:	13 (10+3W)
Addressing:	immediate
Flags:	none

LDA addr (Load Accumulator direct)

(A) <— ((byte 3)(byte 2))

The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is copied to register A.

```
| 0  0  1  1  1  0  1  0 |
|  low-order addr byte   |
|  high-order addr byte  |
```

Mem. Cycles:	4
CPU States:	17 (13+4W)
Addressing:	direct
Flags:	none

STA addr (Store Accumulator direct)

((byte 3)(byte 2)) <— (A)

The content of the accumulator is copied to the memory location whose address is specified in byte 2 and 3 of the instruction.

```
| 0  0  1  1  0  0  1  0 |
|  low-order addr byte   |
|  high-order addr byte  |
```

Mem. Cycles:	4
CPU States:	17 (13+4W)
Addressing:	direct
Flags:	none

LHLD addr (Load H and L direct)

(L) <— ((byte 3)(byte 2))

(H) <— ((byte 3)(byte 2) + 1)

The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is copied to register L. The content of the next (higher) memory location is copied to register H.

```
| 0  0  1  0  1  0  1  0 |
|  low-order addr byte   |
|  high-order addr byte  |
```

Mem. Cycles:	5
CPU States:	21 (16+5W)
Addressing:	direct
Flags:	none

SHLD addr (Store H and L direct)

((byte 3)(byte 2)) <-- (L)
((byte 3)(byte 2) + 1) <-- (H)

The content of register L is copied to the memory location whose address is specified in byte 2 and byte 3. The content of register H is copied to the next (higher) memory location.

<table>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">low-order addr byte</td></tr>
<tr><td colspan="8">high-order addr byte</td></tr>
</table>

Mem. Cycles:	5
CPU States:	21 (16+5W)
Addressing:	direct
Flags:	none

LDAX rp (Load Accumulator indirect)

(A) <-- ((rp))

The content of the memory location, whose address is in the register pair rp, is copied to register A. Note: only register pairs rp=B (registers B and C) or rp =D (registers D and E) may be specified.

0	0	R	P	1	0	1	0

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	reg. indirect
Flags:	none

STAX rp (Store Accumulator indirect)

((rp)) <-- (A)

The content of register A is copied to the memory location whose address is in the register pair rp. Note: only register pairs rp=B (registers B and C) or rp=D (registers D and E) may be specified.

0	0	R	P	0	0	1	0

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	reg. indirect
Flags:	none

XCHG [Exchange (HL) with (DE]

(H) <--> (D)
(L) <—> (E)
The contents of register pair HL is exchanged with the contents of register pair DE.

1	1	1	0	1	0	1	1

Mem. Cycles:	1
CPU States:	5 (4+1W)
Addressing:	register
Flags:	none

Arithmetic Group:

This group of instructions performs arithmetic operations on data in registers and memory.

Unless indicated otherwise, all instructions in this group affect the Zero, Sign, Parity, Carry, and Auxiliary Carry flags according to the standard rules.

All subtraction operations are performed via two's complement arithmetic and set the carry flag to one to indicate a borrow, and clear it to indicate no borrow.

ADD r (Add Register)

(A) <— (A) + (r)
The content of register r is added to the content of the accumulator. The result is placed in the accumulator.

1	0	0	0	0	S	S	S

Mem. Cycles:	1
CPU States:	5 (4+1W)
Addressing:	register
Flags:	Z,S,P,CY,AC

ADD M (Add memory)

(A) <-- (A) + ((H) (L))

The content of the memory location whose address is contained in the HL register pair is added to the content of the accumulator. The result is placed in the accumulator.

1	0	0	0	0	1	1	0

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	reg. direct
Flags:	Z,S,P,CY,AC

ADI data (Add immediate)

(A) <-- (A) + (byte 2)

The content of the second byte of the instruction is added to the content of the accumulator. The result is placed in the accumulator.

1	1	0	0	0	1	1	0
data byte							

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	immediate
Flags:	Z,S,P,CY,AC

ADC r (Add Register with carry)

(A) <-- (A) + (r) + (CY)

The content of register r and the content of the carry bit are added to the content of the accumulator. The result is placed in the accumulator.

1	0	0	0	1	S	S	S

Mem. Cycles:	1
CPU States:	5 (4+1W)
Addressing:	register
Flags:	Z,S,P,CY,AC

ADC M (Add memory with carry)

(A) <-- (A) + ((H) (L)) + (CY)

The content of the memory location whose address is contained in the HL register pair and the content of the CY flag are added to the accumulator. The result is placed in the accumulator.

1	0	0	0	1	1	1	0

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	reg. indirect
Flags:	Z,S,P,CY,AC

ACI data (Add immediate with carry)

(A) <-- (A) + (byte 2) + (CY)

The content of the second byte of the instruction and the content of the CY flag are added to the contents of the accumulator. The result is placed in the accumulator.

<table>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">data byte</td></tr>
</table>

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	immmediate
Flags:	Z,S,P,CY,AC

SUB r (Subtract Register)

(A) <-- (A) - (r)

The content of register r is subtracted from the content of the accumulator. The result is placed in the accumulator.

1	0	0	1	0	S	S	S

Mem. Cycles:	1
CPU States:	5 (4+1W)
Addressing:	register
Flags:	Z,S,P,CY,AC

SUB M (Subtract memory)

(A) <— (A) - ((H) (L))

The content of the byte whose address is in register pair HL is subtracted from the accumulator. The result is placed in the accumulator.

1	0	0	1	0	1	1	0

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	reg. indirect
Flags:	Z,S,P,CY,AC

SUI data (Subtract immediate)

(A) <— (A) - (byte 2)

The content of the second byte of the instruction is subtracted from the content of the accumulator. The result is placed in the accumulator.

1	1	0	1	0	1	1	0
data byte							

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	immediate
Flags:	Z,S,P,CY,AC

SBB r (Subtract Register with borrow)

(A) <— (A) - (r) - (CY)

The content of register r and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

1	0	0	1	1	S	S	S

Mem. Cycles:	1
CPU States:	5 (1+4W)
Addressing:	register
Flags:	Z,S,P,CY,AC

SBB M (Subtract memory with borrow)

(A) <— (A) - ((H) (L)) - (CY)

The content of the memory location whose address is contained in the HL register pair and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

1	0	0	1	1	1	1	0

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	reg. indirect
Flags:	Z,S,P,CY,AC

SBI data (Subtract immediate with borrow)

(A) <— (A) - (byte 2) - (CY)

The contents of the second byte of the instruction and the contents of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

1	1	0	1	1	1	1	0
data byte							

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	immediate
Flags:	Z,S,P,CY,AC

INR r (Increment Register)

(r) <— (r) + 1

The content of register r is incremented by one. All conditions flags except CY are affected.

0	0	D	D	D	1	0	0

Mem. Cycles:	1
CPU States:	5 (4+1W)
Addressing:	register
Flags:	Z,S,P,AC

INR M (Increment memory)

((H) (L)) <— ((H) (L)) + 1

The content of the memory location whose address is contained in the H and L registers is incremented by one. All condition flags except CY are affected.

0	0	1	1	0	1	0	0

Mem. Cycles:	3
CPU States:	13 (10+3W)
Addressing:	reg. indirect
Flags:	Z,S,P,AC

DCR r (Decrement Register)

(r) <— (r) - 1

The content of register r is decremented by one. All condition flags except CY are affected.

0	0	D	D	D	1	0	1

Mem. Cycles:	1
CPU States:	5 (4+1W)
Addressing:	register
Flags:	Z,S,P,AC

DCR M (Decrement memory)

((H) (L)) <— ((H) (L)) - 1

The content of the memory location whose address is contained in the HL register pair is decremented by one. All condition flags except CY are affected.

0	0	1	1	0	1	0	1

Mem. Cycles:	3
CPU States:	13 (10+3W)
Addressing:	reg. indirect
Flags:	Z,S,P,AC

INX rp (Increment register pair)

(rh) (rl) <— (rh) (rl) + 1

The content of the register pair rp is incremented by one. No condition flags are affected.

0	0	R	P	0	0	1	1

Mem. Cycles:	1
CPU States:	7 (6+1W)
Addressing:	register
Flags:	none

DCX rp (Decrement register pair)

(rh) (rl) <— (rh) (rl) - 1

The content of the register pair rp is decremented by one. No conditions flags are affected.

0	0	R	P	1	0	1	1

Mem. Cycles:	1
CPU States:	7 (6+1W)
Addressing:	register
Flags:	none

DAD rp (Add register pair to H and L)

(H) (L) <— (H) (L) + (rh) (rl)

The content of the register pair rp is added to the content of the register pair H and L. The result is placed in the register pair H and L. Only the CY flag is affected. It is set if there is a carry out of the double precision add; otherwise it is reset.

0	0	R	P	1	0	0	1

Mem. Cycles:	1
CPU States:	11 (10+1W)
Addressing:	register
Flags:	CY

DAA (Decimal Adjust Accumulator)

The eight-bit number in the accumulator is adjusted to form two four bit Binary-Coded-Decimal digits by the following process:

1. If the value of the least significant 4 bits of the accumulator is greater than 9 or if the AC flag is set, 6 is added to the accumulator.

2. If the value of the most significant 4 bits of the accumulator is now greater than 9, or if the CY flag is set, 6 is added to the most significant 4 bits of the accumulator.

NOTE: All flags are affected.

0	0	1	0	0	1	1	1

Mem. Cycles:	1
CPU States:	5 (4+1W)
Flags:	Z,S,P,CY,AC

LOGICAL GROUP:

This group of instructions performs logical (Boolean) operations on data in registers and memory and on condition flags.

Unless indicated otherwise, all instructions in this group affect the Zero, Sign, Parity, Auxiliary Carry, and Carry flags according to the standard rules.

ANA r (AND Register)

(A) <— (A) AND (r)

The content of register r is logically anded with the content of the accumulator. The result is placed in the accumulator. The CY flag is cleared.

1	0	1	0	0	S	S	S

Mem. Cycles:	1
CPU States:	5 (4+1W)
Addressing:	register
Flags:	Z,S,P,CY,AC

ANA M (AND memory)

(A) <— (A) AND ((H) (L))

The contents of the memory location whose address is contained in the H and L registers is logically anded with the content of the accumulator. The reslt is placed in the accumulator. The CY flag is cleared.

1	0	1	0	0	1	1	0

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	reg. indirect
Flags:	Z,S,P,CY,AC

ANI data (AND immediate)

(A) <— (A) AND (byte 2)

The content of the second byte of the instruction is logically anded with the contents of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

1	1	1	0	0	1	1	0
data byte							

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	immediate
Flags:	Z,S,P,CY,AC

XRA r (Exclusive OR Register)

(A) <— (A) XOR (r)

The content of register r is exclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

1	0	1	0	1	S	S	S

Mem. Cycles:	1
CPU States:	5 (4+1W)
Addressing:	register
Flags:	Z,S,P,CY,AC

XRA M (Exclusive OR Memory)

(A) <— (A) XOR ((H) (L))

The content of the memory location whose address is contained in the HL register pair is exclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

1	0	1	0	1	1	1	0

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	reg. indirect
Flags:	Z,S,P,CY,AC

XRI data (Exclusive OR immediate)

(A) <— (A) XOR (byte 2)

The content of the second byte of the instruction is exclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

1	1	1	0	1	1	1	0
data byte							

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	immediate
Flags:	Z,S,P,CY,AC

ORA r (OR Register)

(A) <— (A) OR (r)

The content of register r is inclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

1	0	1	1	0	S	S	S

Mem. Cycles:	1
CPU States:	5 (4+1W)
Addressing:	register
Flags:	Z,S,P,CY,AC

ORA M (OR memory)

(A) <— (A) OR ((H)(L))

The content of the memory location whose address is contained in the HL register pair is inclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

1	0	1	1	0	1	1	0

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	reg. indirect
Flags:	Z,S,P,CY,AC

ORI data (OR Immediate)

(A) <— (A) OR (byte 2)

The content of the second byte of the instruction is inclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

<table>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">data byte</td></tr>
</table>

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	immediate
Flags:	Z,S,P,CY,AC

CMP r (Compare Register)

(A) - (r)

The contents of register r are logically subtracted from the accumulator. The contents of the accumulator are unchanged. The condition flags are set as a result of the subtraction. The Z flag is set to 1 if (A) = (r). The CY flag is set to 1 if (A) < (r).

1	0	1	1	1	S	S	S

Mem. Cycles:	1
CPU States:	5 (4+1W)
Addressing:	register
Flags:	Z,S,P,CY,AC

CMP M (Compare memory)

(A) - ((H)(L))

The content of the memory location whose address is contained in the HL register pair is logically subtracted from the accumulator. The contents of the accumulator are unchanged. The condition flags are set as a result of the subtraction. The Z flag is set to 1 if (A) = ((H)(L)). The CY flag is set to 1 if (A) < ((H)(L)).

1	0	1	1	1	1	1	0

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	reg. indirect
Flags:	Z,S,P,CY,AC

CPI data (Compare immediate)

(A) - (byte 2)

The contents of the second byte of the instruction are logically subtracted from the accumulator. The contents of the accumulator are unchanged. The condition flags are set by the result of the subtraction. The Z flag is set to 1 if (A) = (byte 2). the CY flag is set to 1 if (A) < (byte 2).

1	1	1	1	1	1	1	0
data byte							

Mem. Cycles:	2
CPU States:	9 (7+2W)
Addressing:	immediate
Flags:	Z,S,P,CY,AC

RLC (Rotate left)

$(A_{n+1}) \leftarrow (A_n); (A_0) \leftarrow (A_7)$
$(CY) \leftarrow (A_7)$

The content of the accumulator is rotated left one position. The low order bit and the CY flag are both set to the value shifted out of the high order bit position. Only the CY flag is affected.

0	0	0	0	0	1	1	1

Mem. Cycles:	1
CPU States:	5 (4+1W)
Flags:	CY

RRC (Rotate right)

$(A_n) \leftarrow (A_{n+1}); (A_7) \leftarrow (A_0)$
$(CY) \leftarrow (A_0)$

The content of the accumulator is rotated right one position. The high order bit and the CY flag are both set to the value shifted out of the low order bit position. Only the CY flag is affected.

0	0	0	0	1	1	1	1

Mem. Cycles:	1
CPU States:	5 (4+1W)
Flags:	CY

RAL (Rotate left through carry)

$(A_{n+1}) \leftarrow (A_n); (A_0) \leftarrow (CY)$
$(CY) \leftarrow (A_7)$

The content of the accumulator is rotated left one position through the CY flag. The low order bit is set equal to the CY flag and the CY flag is set to the value shifted out of the high order bit. Only the CY flag is affected.

0	0	0	1	0	1	1	1

Mem. Cycles:	1
CPU States:	5 (4+1W)
Flags:	CY

RAR (Rotate right through carry)

(A_{n-1}) <— (A_n); (CY) <— (A_0)

(A_7) <— (CY)

The content of the accumulator is rotated right one position through the CY flag. The high order bit is set to the CY flag and the CY flag is set to the value shifted out of the low order bit. Only the CY flag is affected.

0	0	0	1	1	1	1	1

Mem. Cycles:	1
CPU States:	5 (4+1W)
Flags:	CY

CMA (Complement accumulator)

(A) <—~(A)

The contents of the accumulator are complemented (zero bits become 1, one bits become 0). No flags are affected.

0	0	1	0	1	1	1	1

Mem. Cycles:	1
CPU States:	5(4+1W)
Flags:	none

CMC (Complement carry)

(CY) <—~(CY)

The CY flag is complemented. No other flags are affected.

0	0	1	1	1	1	1	1

Mem. Cycles:	1
CPU States:	5 (4+1W)
Flags:	CY

STC (Set carry)

(CY) <— 1

The CY flag is set to 1. No other flags are affected.

0	0	1	1	0	1	1	1

Mem. Cycles:	1
CPU States:	5 (4+1W)
Flags:	CY

BRANCH GROUP:

This group of instructions alter normal sequential program flow. Condition flags are not affected by any instruction in this group.

The two types of branch instructions are unconditional and conditional. Unconditional transfers simply perform the specified operation on register PC (the program counter). Conditional transfers examine the status of one of the four processor flags to determine if the specified branch is to be executed. The conditions that may be specified are as follows:

CONDITION	**CCC**
NZ - not zero (Z = 0)	000
Z - zero (Z = 1)	001
NC - no carry (CY = 0)	010
C - carry (CY = 1)	011
PO - parity odd (P = 0)	100
PE - parity even (P = 1)	101
P - plus (S = 0)	110
M - minus (S = 1)	111

JMP addr (Jump)

(PC) <— (byte 3) (byte 2)

Control is transferred to the instruction whose address is specified in bytes 3 and 2 of the current instruction.

<table>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td colspan="8">low-order addr</td></tr>
<tr><td colspan="8">high-order addr</td></tr>
</table>

Mem. Cycles:	3
CPU States:	13 (10+3W)
Addressing:	immediate
Flags:	none

Jcondition addr (Conditional jump)

If(CCC),

(PC) <— (byte 3)(byte 2)

If the specified condition is true, control is transferred to the instruction whose address is specified in bytes 3 and 2 of the current instruction; otherwise, control continues sequentially.

<table>
<tr><td>1</td><td>1</td><td>C</td><td>C</td><td>C</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">low-order addr</td></tr>
<tr><td colspan="8">high-order addr</td></tr>
</table>

Mem. Cycles:	2/3
CPU States:	9/13 (7+2W/10+3W)
Addressing:	immediate
Flags:	none

CALL addr (Call)

```
((SP) -1) <— (PCH)
((SP) -2) <— (PCL)
 (SP) <— (SP) -2
 (PC) <— (byte 3)(byte 2)
```

First, the contents of the Program Counter are PUSHed into the Stack. Next, the two address bytes following the CALL opcode replace the Program Counter, effecting a branch to that address.

```
 ___________________________________
|    |   |   |   |   |   |   |   |  |
|  1   1   0   0   1   1   0   1    |
|                                   |
|_______low-order addr byte_________|
|_______high-order addr byte________|
```

Mem. Cycles:	5
CPU States:	23 (18+5W)
Addressing:	immediate/reg. indirect
Flags:	none

Ccondition addr (Condition call)

```
If(CCC),
    ((SP) -1) <— (PCH)
    ((SP) -2) <— (PCL)
     (SP) <— (SP) -2
     (PC) <— (byte 3)(byte 2)
```

If the specified condition is true, the actions specified in the CALL instruction (see above) are performed; otherwise, control continues sequentially.

```
 ___________________________________
|    |   |   |   |   |   |   |   |  |
|  1   1   C   C   C   1   0   0    |
|                                   |
|_______low-order addr byte_________|
|_______high-order addr byte________|
```

Mem. Cycles:	2/5
CPU States:	11/23 (9+2W/18+5W)
Addressing:	immediate/reg. indirect
Flags:	none

RET (Return)

```
(PCL) <— ((SP))
(PCH) <— ((SP) + 1)
 (SP) <— (SP) + 2
```

The Program Counter is POP'd from the Stack.

1	1	0	0	1	0	0	1

Mem. Cycles:	3
CPU States:	13 (10+3W)
Addressing:	reg. indirect
Flags:	none

Rcondition (Conditional return)

```
If(CCC),
    (PCL) <— ((SP))
    (PCH) <— ((SP) + 1)
     (SP) <—  (SP) + 2
```

If the specified condition is true, the Program Counter is POP'd from the Stack; otherwise, control continues sequentially.

1	1	C	C	C	0	0	0

Mem. Cycles:	1/3
CPU States:	7/15 (6+1W/12+3W)
Addressing:	reg. indirect
Flags:	none

RST n (Restart)

```
((SP) - 1) <-- (PCH)
((SP) - 2) <-- (PCL)
 (SP) <-- (SP) -2
 (PC) <-- 8 * (NNN) where NNN binary = n decimal
```

The Program Counter is PUSHed onto the Stack, then set to 8*n.

1	1	N	N	N	1	1	1

Mem. Cycles:	3
CPU States:	15 (12+3W)
Addressing:	reg. indirect
Flags:	none

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

Program Counter After Restart

PCHL (Jump H and L indirect - move H and L to PC)

```
(PCH) <-- (H)
(PCL) <-- (L)
```

The content of register H is moved to the high order eight bits of register PC. The content of register L is moved to the low order eight bits of register PC. This effects a branch to the address contained in HL.

0	0	1	1	1	0	1	0

Mem. Cycles:	1
CPU States:	7 (6+1W)
Addressing:	register
Flags:	none

Stack, I/O, and Machine Control Group:

This group of instructions performs I/O, manipulates the Stack, and alters internal control flags.

Unless otherwise specified, condition flags are not affected by any instructions in this group.

PUSH rp (Push)

((SP) - 1) <— (rh)
((SP) - 2) <— (rl)
(SP) <— (SP) -2

The content of the high order register of register pair rp is moved to the memory location whose address is one less than the content of register SP. The content of the low order register of register pair rp is moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by 2. Register pair rp = SP may not be specified.

1	1	R	P	0	1	0	1

Cycles:	3
States:	15 (12+3W)
Addressing:	reg. indirect
Flags:	none

PUSH PSW (Push processor status word)

((SP) - 2) <— PSW

The Accumulator is PUSHed onto the Stack. A Program Status Word (PSW) byte is created from the condition flags and PUSHed onto the Stack.

1	1	1	1	0	1	0	1

Instruction

S	Z	0	AC	0	P	1	CY

PSW Byte

Mem. Cycles:	3
CPU States:	15 (12+3W)
Addressing:	reg. indirect
Flags:	none

POP rp (Pop)

(rl) <— ((SP))
(rh) <— ((SP) + 1)
(SP) <— (SP) + 2

The content of the memory location, whose address is specified by the content of register SP, is moved to the low order register of register pair rp. The content of the memory location, whose address is one more than the content of register SP, is moved to the high order register of register pair rp. The content of register SP is incremented by 2. Register pair rp = SP may not be specified.

1	1	R	P	0	0	0	1

Mem. Cycles:	3
CPU States:	13 (10+3W)
Addressing:	reg. indirect
Flags:	none

POP PSW (Pop processor status word)

flags <— (SP)
(A) <— ((SP)) + 1)
(SP) <— (SP) + 2

The PSW byte is POP'd from the Stack and the processor flags are copied from this byte. The Accumulator is POP'd from the Stack.

1	1	1	1	0	0	0	1

Instruction

S	Z	0	AC	0	P	1	CY

PSW Byte

Mem. Cycles:	3
CPU States:	13 (10+3W)
Addressing:	reg. indirect
Flags:	Z,S,P,CY,AC

XTHL (Exchange stack top with H and L)

(L) <--> ((SP))
(H) <--> ((SP) + 1)

The content of the L register is exchanged with the content of the memory location whose address is specified by the content of register SP. The content of the H register is exchanged with the content of the memory location whose address is one more than the content of register SP.

1	1	1	0	0	0	1	1

Mem. Cycles:	5
CPU States:	21 (16+5W)
Addressing:	reg. indirect
Flags:	none

SPHL (Copy HL to SP)

(SP) <— (H) (L)

The contents of register pair HL (16 bits) is copied to double register SP.

1	1	1	1	1	0	0	1

Cycles:	1
States:	7 (6+1W)
Addressing:	register
Flags:	none

EI (Enable Interrupts)

The interrupt system is enabled following the execution of the next instruction.

1	1	1	1	1	0	1	1

Mem. Cycles: 1
CPU States: 5 (4+1W)
Flags: none

DI (Disable interrupts)

The interrupt system is disabled immediately following the execution of the DI instruction.

1	1	1	1	0	0	1	1

Mem. Cycles: 1
CPU States: 5 (4+1W)
Flags: none

HLT (Halt)

The processor is stopped. The registers and flags are unaffected.

0	1	1	1	0	1	1	0

Mem. Cycles: 1
CPU States: 6 (5+1W)
Flags: none

NOP (No op)

No operation is performed. The registers and flags are unaffected.

0	0	0	0	0	0	0	0

Mem. Cycles: 1
CPU States: 5 (4+1W)
Flags: none

INPUT/OUTPUT INSTRUCTIONS

All Input/Output instructions on the OP-1R are two byte instructions. The first byte is the operation code (either IN or OUT). The second byte is a code number which indicates a function to be performed. Certain devices also require a data byte in the Accumulator which further specifies the function. Many of the functions operate differently depending upon the devices to which they are directed.

Certain I/O function codes operate independently of any device. These codes are as follows:

OPCODE	OPERAND	MNEMONIC	FUNCTION
IN	02	IIN	Reads the OP-1R Interrupt Status Register into the Accumulator.
IN	03	FIX1	Reads the contents of FIXED DATA SWITCH 1 into the Accumulator.
IN	04	FIX2	Reads the contents of FIXED DATA SWITCH 2 into the Accumulator.
OUT	00	INIT	Stop all current devices and clears and initializes all devices.
OUT	01	SEL	Causes all subsequent Device Specific I/O instructions to be directed to the device whose add ress is in the Accumulator when OUT 01 is issued. (Selects a device.) This device will be referred to later as the <u>CURRENT DEVICE</u>.

OPCODE	OPERAND	MNEMONIC	FUNCTION
OUT	0CH	SMSK	Sets the current interrupt mask to the value in the Accumulator.
OUT	0DH	BEEP	Causes the audible (beep) alarm to be sounded.
OUT	0EH	CLICK	Causes an audible click.

Device Specific I/O Instructions should only be issued after a Device has been selected via an OUT 01 instruction.

OPCODE	OPERAND	MNEMONIC	FUNCTION
IN	00	IFL	Reads a byte of status information from the currently selected device into the accumulator.
IN	01	INP	Reads a byte of data from the currently selected device into the accumulator.
OUT	02	OUT	Outputs a byte of data to the currently selected device from the Accumulator.
OUT	03	DVCL	Issues a Device Clear Signal which stops and resets the currently selected device
OUT	04	OFL	Output a byte of Flags or Control information to the currently selected device from the accumulator.
OUT	05	COM1	Are used to output different classes of command bytes to the currently selected device. These instructions are usually interpreted differently depending on the device selected.
OUT	06	COM2	
OUT	07	COM3	

Note: The following sections refer to I/O instructions by their mnemonic name. The Ontel assembler recognizes I/O commands only if both opcode (in, out) and operand (i.e. 00H or mnemonic IFL, if IFL is equated to 00H) are specified.

Device Address assignments are listed below.

ADDRESS	DEVICE
F0	Asynchronous I/O Adapter
E1	Keyboard
2B	Alternate I/O Adapter
0F	Parallel Input
1B	Parallel Output

3R-42. Device Address Assignments

SECTION 4R

INTERRUPT CONTROLLER

The interrupt controller processes all interrupt requests issued by the individual devices. The OP-1/R is designed for a maximum of four hardware interrupt requests from the main logic board and four more from an option board. The interrupt priorities and selection of device(s) from which to accept interrupts are program controlled. Figure 4R-1 illustrates the OP-1/R interrupt structure.

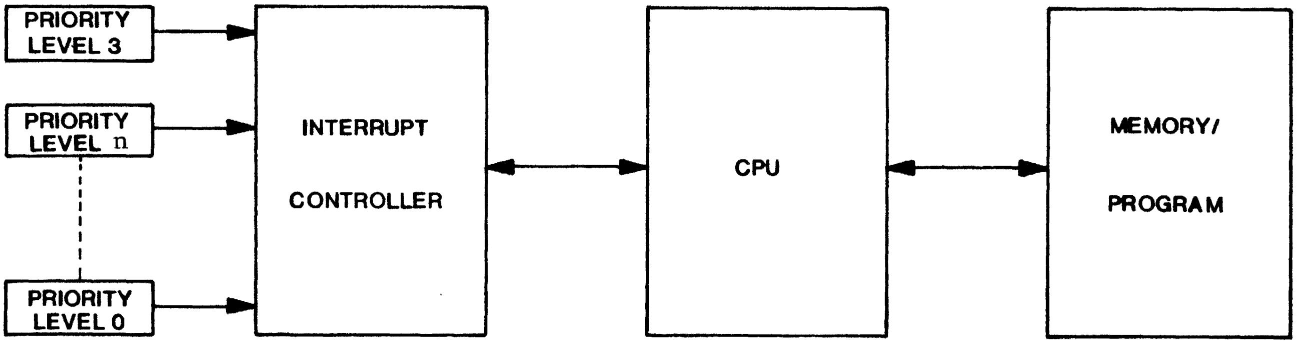
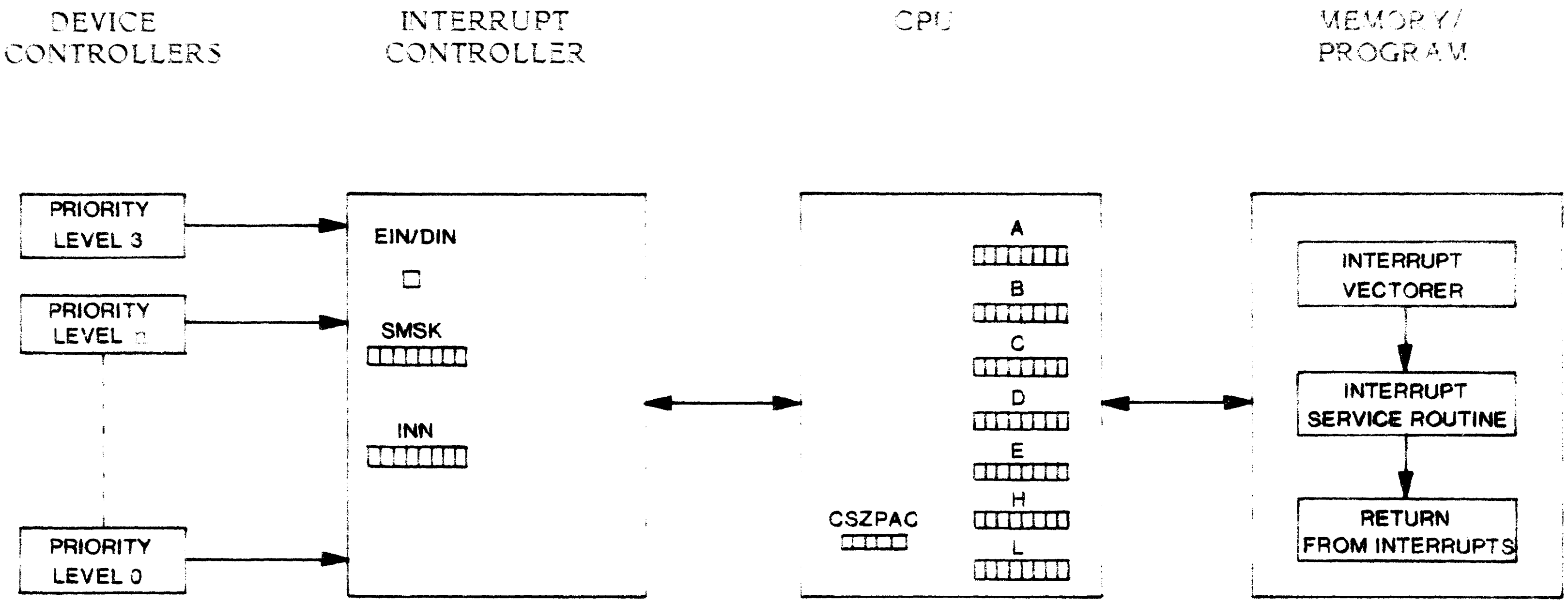


Figure 4R-1. OP-1/R Interrupt Structure

The following conditions have to be met before the interrupt controller will issue an interrupt request to the CPU:

1. Interupt request from a device present.
2. Interrupts enabled.
3. Interrupt mask bit set corresponding to the interrupt request.

After receiving the interrupt request, the CPU will complete its present instruction and then call location 038 (Interrupt Program) in memory using one level of pushdown stack. The existing CPU environment and registers can be saved during interrupt processing in main memory. If two interrupts occur simultaneously, the priorities are serviced in the order determined by the program. The sequence of interrupt operations is illustrated in Figure 4R-2.



Device Controllers

1. General Interrupt Request

Interrupt Controller

1. Check: Interrupts Enabled SMSK Set
2. Set IIN Status Register
3. Issue Interrupt to CPU
4. Disable Interrupts

CPU

1. Complete current operation
2. Call Interrupt Program in Memory

Memory/Program

1. Save All Registers and Condition Flags
2. Service Interrupt with highest priority
3. Restore All Registers and Condition Flags
4. Return to Program in process and Enable Interrupts

Figure 4R-2. Op-1/R Interrupt Sequence of Operations

Table 4R-1 lists standard OP-1/R interruptt assignments:

PRIORITY LEVEL	FUNCTION
4 (highest)	Alternate I/O Adapter
3	Parallel Input/Output
2	Asynchronous-Receive
1	Asynchronous-Transmit
0	Keyboard/Real Time Clock

Table 4R-1. OP-1/R Interrupt Assignment

COMMANDS

ENABLE INTERRUPTS

Command: EI

Command Byte: None

Enables all interrupts. Interrupt enable is set after the first instruction following EI is executed. It is reset by a DI instruction or by the hardware when any interrupt is processed.

DISABLE INTERRUPTS

Command: DI

Command Byte: None

Disables all interrupts.

SET INTERRUPT MASK

Command: OUT SMSK

Command Byte:

Bit 4	Alternate I/O Adapter
Bit 3	Parallel Input/Output
Bit 2	Asynchronous-Receive
Bit 1	Asynchronous-Transmit
Bit 0	Keyboard/Real Time Clock

Selects the device(s) from which to accept interrupts by setting the corresponding bit(s) in the SMSK Register to a "1".

INTERRUPT STATUS

Command: IN IIN

Status Byte:

Bit 4	Alternate I/O Adapter
Bit 3	Parallel Input/Output
Bit 2	Asynchronous-Receive
Bit 1	Asynchronous-Transmit
Bit 0	Keyboard/Real Time Clock

Loads the contents of the Interrupt Status Register into the accumulator. If the corresponding bit(s) of the SMSK register are set, the Interrupt Status Register bit(s) are set to a "1" when an interrupt request is detected by the CPU. Each bit is reset by the appropriate interrupt Service Routine.

Bits 3-7 will be a "1", if options using these bits are not installed.

STORE/RESTORE CPU STATUS

Prior to servicing an interrupt, the status of the CPU can be saved by transferring the condition flags and register contents to the stack by means of the PUSH instruction.

After the interrupt has been serviced, the CPU status can be restored by means of the POP instruction.

SECTION 5R

FIXED DATA SWITCHES

Two eight-bit fixed data switches have been provided for general programming purposes. The switches can be manually set at the time of installation to specify the particular OP-1/R identifying address or any other general function such as mode selection.

The switch configuration is illustrated below:

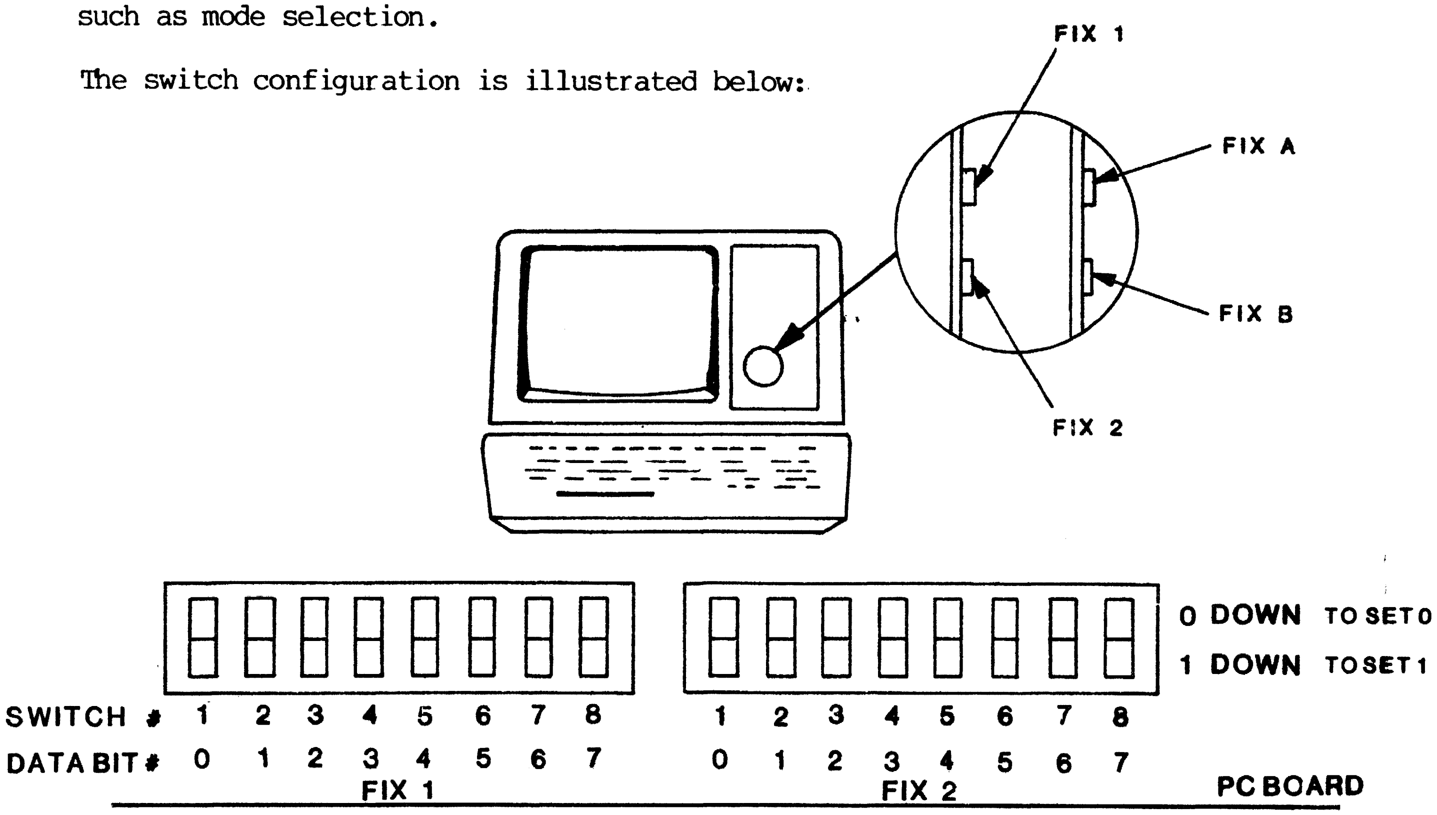

FIX 1
FIX A
FIX B
FIX 2
0 DOWN TO SET 0
1 DOWN TO SET 1
SWITCH # 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8
DATA BIT # 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
FIX 1
FIX 2
PC BOARD

Figure 5R-1. Fixed Data Switches

COMMANDS

The following commands can be executed:

<u>INPUT DATA</u>

Command: IN FIX1

Transfers the contents of the FIX1 switch to the Accumulator. The content of switch Bit n is loaded into Accumulator Bit n-1.

Command: IN FIX2

Transfers the contents of the FIX2 switch to the Accumulator. The content of switch Bit n is loaded into Accumulator Bit n-1.

SECTION 6R

KEYBOARD AND REAL TIME CLOCK

The movable keyboard provides an interface between the operator and the OP-1/R. Every key generates a unique code readable by the CPU. The keyboard is arranged in four functional sections:

1. ASCII Section
2. Control Pad
3. Numeric Pad
4. Function Pad

Figure 6R-1 illustrates the Keyboard arrangement.

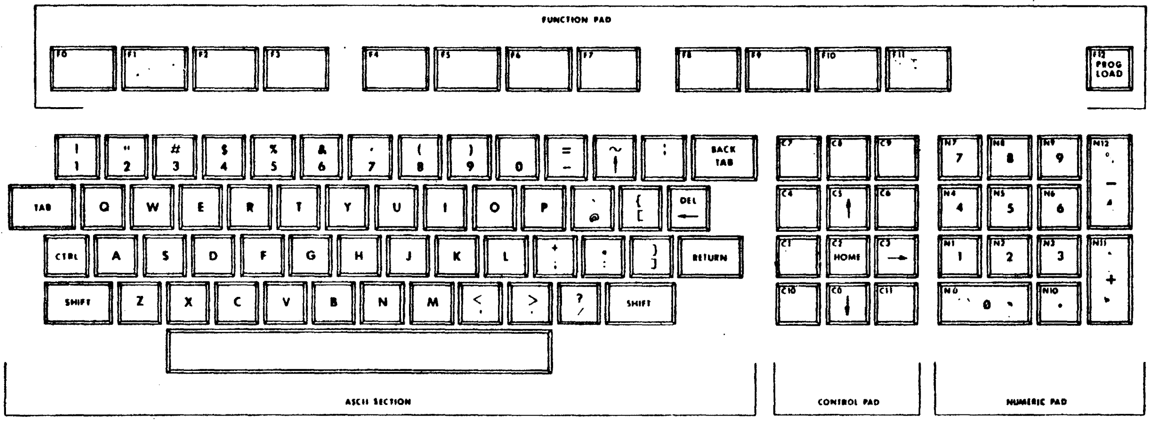


Figure 6R-1. Keyboard Arrangement

The ASCII Section contains the entire upper and lower case alphabet, the number set and standard operator controls such as Shift, Tab and Return. the Control Pad houses 12 control keys including cursor controls. Numerals 0-9, period, plus and minus on the 13-key Numeric Pad are arranged in a calculator format. The Function Pad consists of 12 multi-purpose functions keys, four of which may contain an integral status light and a Program Load (PROG) key.

The keyboard design includes an audible alarm and an automatic repeat feature: every key held down for more than one second will repeat its code at 15 times/second.

The CPU can be interrupted with every key depression. Every key generates up to four unique codes, e.g., the numerals in the ASCII Section generate different codes than the corresponding keys on the Numeric Pad. The codes generated are listed in Tables 6R-1 through 6R-4.

COMMANDS

Commands to the keyboard may be executed if it has been selected by the CPU as the active I/O device.

SELECT

Command: OUT SEL

Command Byte: E1

Selects the keyboard for I/O operation.

STOP

Command: OUT DVCL

Command Byte: None

Clears any character that may have been entered on the keyboard. Resets the interrupt request, Keyboard Character Available flag, Real Time Clock and Timeout Flag.

STATUS

Command: IN IFL

Command Byte:

Bit 7	Keyboard Character Available
Bit 6	Real Time Clock Timeout

Loads the accumulator with keyboard operational status.

INPUT

Command: IN INP

Reads the eight-bit code from the keyboard into the accumulator. Resets the Keyboard Character Available flag and the Keyboard interrupt request.

SET STATUS LIGHTS

Command: OUT OFL

Command Byte:

Bit 3	Turn F3 Light On
Bit 2	Turn F2 Light On
Bit 1	Turn F1 Light On
Bit 0	Turn F0 Light On

Turns on status lights housed in keys F0, F1, F2, F3 located on the Function Pad.

AUDIBLE TONES

Command: OUT BEEP

Command Byte: None

Activates a one second audible tone. The keyboard does not have to be selected to execute a BEEP instruction.

Command: OUT CLICK

Command Byte: None

Activates an audible click. The keyboard does not have to be selected to execute.

REAL TIME CLOCK

Command: OUT COM1

Command Byte: None

Starts a 1 second ± 10% timer. At the end of the timing interval, Status Bit 6 and an Interrupt Request 0 will be set. The keyboard must be selected to execute this command.

Executing another COM1 during the timing interval will reset the timer and initiate a new 1 second timing interval.

Command: OUT COM2

Command Byte: None

Cancels the timing interval by clearing the timer. Status Bit 6 and its Interrupt Request will be reset if set. The keyboard must be selected to execute this command.

INTERRUPT CONTROL

Priority Level No. 0 (lowest priority) Set when a keyboard character is available. Reset by the INP or DVCL command. Identical to IFL status bit 7. Also set when the Real Time Clock times out. Identical to IFL status bit 6. Reset by COM1, COM2 or DVCL.

PROGRAM LOAD KEY (PROG)

Depressing the PROG key in conjunction with the SHIFT and CTRL key causes the CPU to execute the instruction stored in location 8000.

Table 6R-1. Keyboard Codes - ASCII Section

Key Legend	Code Generated				Key Legend	Code Generated			
	U	S	C	SC		U	S	C	SC
SPACE	20	20	A0	A0	0	30	30	B0	B0
! 1	31	21	B1	A1	` @	40	60	00	00
" 2	32	22	B2	A2	A	61	41	01	01
# 3	33	23	B3	A3	B	62	42	02	02
$ 4	34	24	B4	A4	C	63	43	03	03
% 5	35	25	B5	A5	D	64	44	04	04
& 6	36	26	B6	A6	E	65	45	05	05
' 7	37	27	B7	A7	F	66	46	06	06
(8	38	28	B8	A8	G	67	47	07	07
) 9	39	29	B9	A9	H	68	48	08	08
* :	3A	2A	BA	AA	I	69	49	09	09
+ ;	3B	2B	BB	AB	J	6A	4A	0A	0A
< ,	2C	3C	AC	BC	K	6B	4B	0B	0B
= -	2D	3D	AD	BD	L	6C	4C	0C	0C
> .	2E	3E	AE	BE	M	6D	4D	0D	0D
? /	2F	3F	AF	BF	N	6E	4E	0E	0E

U=UNSHIFTED S=SHIFTED C=WITH CTRL KEY SC=SHIFTED WITH CTRL KEY

Table 6R-1. Keyboard Codes - ASCII Section
(continued)

Key Legend	Code Generated				Key Legend	Code Generated			
	U	S	C	SC		U	S	C	SC
O	6F	4F	0F	0F	↑	5E	7E	1E	1E
P	70	50	10	10	DEL ←	5F	7F	1F	1F
Q	71	51	11	11	⇤	08	08	08	08
R	72	52	12	12	TAB	09	09	09	09
S	73	53	13	13	RETURN	0D	0D	0D	0D
T	74	54	14	14					
U	75	55	15	15					
V	76	56	16	16					
W	77	57	17	17					
X	78	58	18	18					
Y	79	59	19	19					
Z	7A	5A	1A	1A					
{ [	5B	7B	1B	1B					
| \	5C	7C	1C	1C					
}]	5D	7D	1D	1D					

U=UNSHIFTED S=SHIFTED C=WITH CTRL KEY SC=SHIFTED WITH CTRL KEY

Table 6R-2. OP-1 Keyboard Codes - Control Pad

Ref. No.	Key Legend	Code Generated				Ref No.	Key Legend	Code Generated			
		U	S	C	SC			U	S	C	SC
C0	↓	80	90	80	90	C6		86	96	86	96
C1	←	81	91	81	91	C7		87	97	87	97
C2	HOME	82	92	82	92	C8		88	98	88	98
C3	→	83	93	83	93	C9		89	99	89	99
C4		84	94	84	94	C10		8A	9A	8A	9A
C5	↑	85	95	85	95	C11		8B	9B	8B	9B
N0	0	C0	D0	C0	D0	N7	7	C7	D7	C7	D7
N1	1	C1	D1	C1	D1	N8	8	C8	D8	C8	D8
N2	2	C2	D2	C2	D2	N9	9	C9	D9	C9	D9
N3	3	C3	D3	C3	D3	N10	.	CB	DB	CB	DB
N4	4	C4	D4	C4	D4	N11	+	CD	DD	CD	DD
N5	5	C5	D5	C5	D5	N12	-	CF	DF	CF	DF
N6	6	C6	D6	C6	D6						

U=UNSHIFTED S=SHIFTED C=WITH CTRL KEY SC=SHIFTED WITH CTRL KEY

Table 6R-4. OP-1 Keyboard Codes - Function Pad

Ref. No.	Key Legend	Code Generated				Ref. No.	Key Legend	Code Generated			
		U	S	C	SC			U	S	C	SC
F0		E0	F0	E0	F0	F7		E7	F7	E7	F7
F1		E1	F1	E1	F1	F8		CE	DE	CE	DE
F2		E2	F2	E2	F2	F9		CA	DA	CA	DA
F3		E3	F3	E3	F3	F10		CC	DC	CC	DC
F4		E4	F4	E4	F4	F11		EE	FE	EE	FE
F5		E5	F5	E5	F5	F12	PROG	EF	FF	ED	(1)
F6		E6	F6	E6	F6						

U=UNSHIFTED S=SHIFTED C=WITH CTRL KEY SC=SHIFTED WITH CTRL KEY

(1) Causes the CPU to execute the instruction stored in memory location 8000.

SECTION 7R

ASYNCHRONOUS I/O ADAPTER

The Asynchronous I/O Adapter connects the CPU I/O bus to an external communications modem. Operation is performed in full duplex in a serial start-stop format. The adapter is designed to operate at 50 to 19200 bits-per-second using an EIA RS232C interface or 2 wire direct interface. A 20 ma current loop interface option is limited to 9600 bits per second. All communications functions are program controlled. The CPU can be interrupted by the adapter after completion of every character received or transmitted.

COMMANDS

Commands to the Asynchronous I/O Adapter may be executed if it has been selected by the CPU as the active I/O device. The adapter will remain selected until a different I/O device selection is made.

SELECT

Command: SEL

Command Byte: F0

Selects the Asynchronous Adapter for I/O operations.

STOP

Command: DVCL

Command Byte: None

Resets the Asynchronous I/O Adapter and aborts any current activity. DVCL should be used prior to issuing any operational command when the status of the Asynchronous Adapter is uncertain. The DVCL command will reset the Character Received and Available flag, the Parity, Overrun or Framing Error flag and the output control signals to the modem (Bits 7 and 5 of IFL and OFL Bits 3, 2, 1 and 0). DVCL also sets the Character Needed for Transmission flag (IFL bit 6).

STATUS

Command: IFL

Status Byte:

Bit 7	Character Received and Available
Bit 6	Character Needed for Transmission
Bit 5	Parity, Overrun or Framing Error. This bit remains High until the next character is received.
Bit 4	Clear to Send signal is On
Bit 3	
Bit 2	Carrier Detect signal is On
Bit 1	Data Set Ready signal is On
Bit 0	

Loads the accumulator with an operational status byte from the Asynchronous I/O Adapter. Bits 0 to 4 follow the modem signals.

OUTPUT

Command: OUT

Transfers a data byte from the accumulator to the Asynchronous I/O Adapter for transmission. It should be issued only after the adapter has indicated that a character is needed for transmission. The Character Needed for Transmission flag will be reset until the character is transmitted and a new character is required.

INPUT

Command: INP

Transfers a received data byte from the Asynchronous I/O Adapter to the accumulator. It should be issued only after the adapter has indicated that a received character is available. INP resets the Character Received and Available flag.

SET MODEM

Command: OFL

Command Byte:

Bit 3	Break Transmitted Data
Bit 2	
Bit 1	Data Terminal Ready
Bit 0	Hold Request to Send signal On

SET COMMUNICATIONS PARAMETERS

Command: COM1

Command Byte:

Bit 7	Two Stop Bits (vs. one)
Bit 6	Eight Data Bits (vs. seven)
Bit 5	No Parity
Bit 4	Even Parity (vs. odd)
Bit 3 Bit 2 Bit 1 Bit 0	Baud rate selection in the following format:

3	2	1	0	BAUD RATE
0	0	0	0	50
0	0	0	1	75
0	0	1	0	110
0	0	1	1	134.5
0	1	0	0	150
0	1	0	1	300
0	1	1	0	600
0	1	1	1	1200
1	0	0	0	1800
1	0	0	1	2000
1	0	1	0	2400
1	0	1	1	3600
1	1	0	0	4800
1	1	0	1	7200
1	1	1	0	9600
1	1	1	1	19200

Establishes communications parameters by transferring a command byte from the accumulator to the Asynchronous I/O Adapter. The parameters will remain set until changed by a subsequent COM1.

INTERRUPT CONTROL

The Asynchronous I/O Adapter provides two interrupt request signals to the CPU:

Priority Level No. 2 (No. 3 is highest) A character is received and available. Identical to IFL status bit 7.

Priority Level No. 1 A Character is needed for transmission. Identical to IFL status bit 6.

PIN	CKT	DESCRIPTION
1	AA	Protective Ground
2	BA	Transmitted Data
3	BB	Received Data
4	CA	Request to Send
5	CB	Clear to Send
6	CC	Data Set Ready
7	AB	Signal Ground
8	CF	Carrier Detector
11		
12		
20	CD	Data Terminal Ready
22		

Connector: Cannon DBC-25S

Table 7R-1-1 Asynchronous I/O Adapter Pin Assignments EIA-RS232 Signals

This interface satisfies the requirements of the standard asynchronous interface between data terminal and data communications equipment defined by EIA Standard RS 232.

PIN	DESCRIPTION
1	Protective Ground
2	Serial Output Neg.
3	Serial Output Pos.
5	Serial Input Neg.
6	Serial Input Pos.
7	Signal Ground

Connector: Cannon DBC-25S

Table 7R-1-2 Asynchronous I/O Adapter Pin Assignments
20 ma Current Loop Signals

This interface uses a 20 ma current source supplied by the host computer. The driver will switch currents up to 20 ma (voltage not to exceed 30 VDC). The opto-isolator receiver requires a minimum of 20 ma with a maximum of 25 ma. The data lines can be strapped to allow current flow to be either a mark or space condition.

PIN	DESCRIPTION
1	Protective Ground
4	Data (Bidirectional)
7	Signal Ground

Connector: Cannon DBC-25S

Table 7R-1-3 Asynchronous I/O Adapter Pin Assignments
2 Wire Direct Signals

This is a half duplex interface utilizing two wires: a data line that switches from -12V (mark) to ground (space) and a ground line.

NOTE: RTS turns off received data.

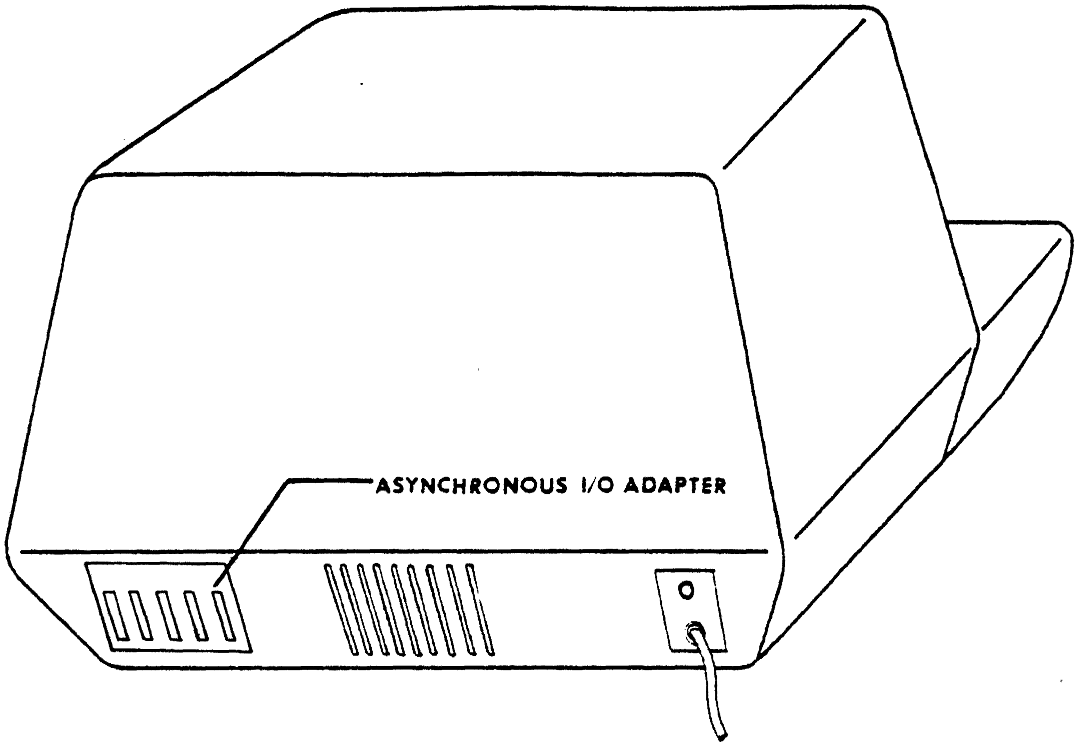


Figure 7R-1. External Connector to Asynchronous I/O Adapter

Figure 7R-1 illustrates the Asynchronous I/O Adapter connector location.

SECTION 8R

PARALLEL OUTPUT PRINTER ADAPTER/PARALLEL INPUT ADAPTER

The Parallel I/O Adapter can be configured as either a Parallel Output Printer Adapter or a Parallel Input Adapter.

PARALLEL OUTPUT/ADAPTER

The Printer Adapter interfaces the CPU I/O bus to a parallel interface printer. All printer functions are program controlled. The CPU can be interrupted by the printer adapter whenever it is ready to receive a character.

COMMANDS

Commands to the Adapter may be executed only if the adapter has been selected by the CPU as the active I/O device. The adapter will remain selected until a different I/O device selection is made.

SELECT

Command:	SEL
Command Byte:	1B

Selects the Adapter for I/O operation.

PRINT

Command: OUT

Transfers a character from the accumulator to the printer via the Adapter. The OUT command also resets the interrupt request and the character request status bit.

STOP

Command:	DVCL
Command Byte:	None

Resets the adapter and aborts any current activity. This command should be used prior to issuing a print command when the status of the controller is uncertain, such as the start of a program.

STATUS

Command: IFL
Status Byte:

Bit 7 NOT BUSY Set when the Adapter is ready to receive a new print or control character.

Bit 6 PRINTER SELECTED Set when the printer has been selected either locally or by CPU command.

Bit 1 PRINTER NOT READY Set when the printer is not ready to receive data, i.e. it is not connected or is out of paper, etc.

Bit 0 PRINTER BUSY Set when the printer is executing a print operation or when the printer is ready but not selected.

Loads the accumulator with an operational status byte from the printer adapter.

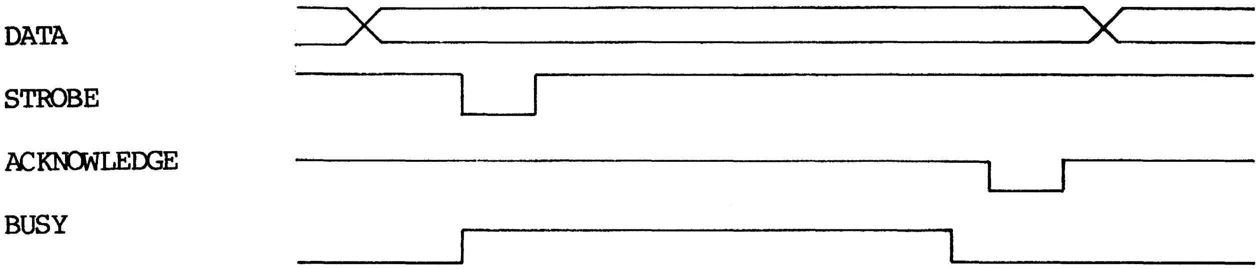
INTERRUPT CONTROL

The Printer Adapter provides an interrupt request whenever it is ready to receive a print or control character. The priority level shall be level 3.

PIN	DESCRIPTION
1	Protective Ground
2	Printer Acknowledge
3	Data Strobe
4	Printer Selected Signal
5	Printer Busy
6	Printer Fault
7	Signal Ground
8	Data Bit 2
9	Data Bit 5
10	Data Bit 6
11	Data Bit 7
15	Data Bit 4
17	Data Bit 3
20	Data Bit 0
21	Printer Prime
22	Data Bit 1

Connector: Cannon DBC-25S

Table 8R-1-1 Printer Adapter Pin Assignments



This is a parallel output interface using TTL compatible signals. Each byte is output together with a 1.5 usec strobe and must be acknowledged before the next byte.

PARALLEL INPUT ADAPTER

The Parallel Input Adapter interfaces the CPU I/O bus to a general purpose parallel input interface. The CPU can be interrupted when input data has been loaded into the adapter.

COMMANDS

Commands to the Parallel Input Adapter may be executed only if the adapter has been selected by the CPU as an active I/O device. The adapter will remain selected until a different I/O device selection is made.

SELECT

Command: SEL
Command Byte: OF

Selects the Parallel Input Adapter for I/O operation.

STOP

Command: DVCL
Command Byte: None

Resets the adapter and aborts any current activity. This command should be used prior to issuing a parallel input command when the status of the controller is uncertain, such as the start of a program. This command should be followed by an INP command to reset the interrupt request.

INPUT

Command: INP
Command Byte: Data - 8 Bits

Transfers a character from the Parallel Input Adapter to the accumulator. The INP Command also resets the interrupt request and data available flag.

STATUS

Command: IFL
Status Byte:

Bit 7	Data Available	Set when data is loaded into the Paralel Input Adapter.

Bit 6: General Purpose input status bits for Programmer's use.

Bit 1: General Purpose input status bits for Programmer's use.

Bit 0: General Purpose input status bits for Programmer's use.

Loads the accumulator with an operational status byte from the parallel input adapter.

INTERRUPT CONTROL

The parallel Input Adaptor provides an interrupt request whenever it has received a data byte. The interrupt request is reset when the CPU reads the data. Priority level is 3.

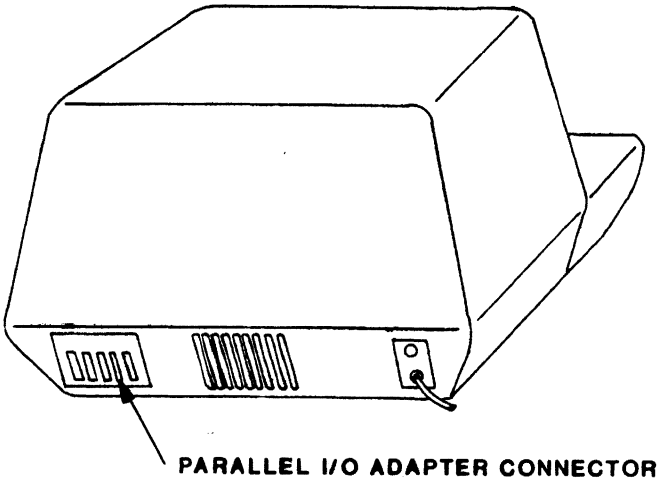
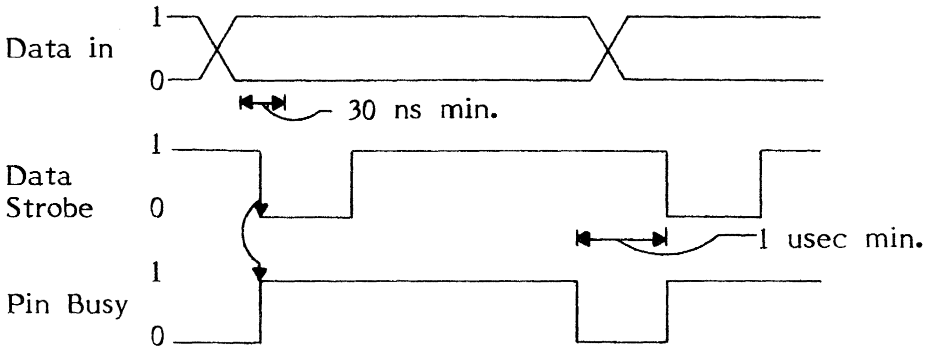


Figure 8R-1

PIN	DESCRIPTION	I/O
1.	Protective Gnd	
2.	Data Strobe (High to Low edge latches in data)	Input
3.	(Illegal Connection)	
4.	General Status Input	Input
5.	General Status Input	Input
6.	General Status Input	Input
7.	GND	GND
8.	Data Bit 2	Input
9.	Data Bit 5	Input
10.	Data Bit 6	Input
11.	Data Bit 7	Input
14.	Parallel In Busy	Output
15.	Data Bit 4	Input
17.	Data Bit 3	Input
20.	Data Bit 0	Input
21.	Clear Pulse (Power on, INIT)	Output
22.	Data Bit 1	Input

Table 8R-1-2 Parallel Input Adapter Pin Assignments



Timing - All Signals TTL

SECTION 9R

DISPLAY MICROPROCESSOR AND CRT

The CRT display unit consists of a Display Microprocessor, video generator and a 14-inch diagonal equipped with a non-glare CRT faceplate. All display functions are program controlled.

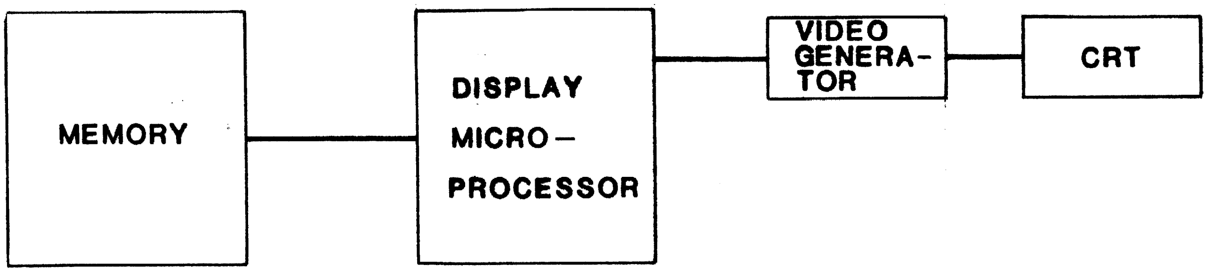


Figure 9R-1. Display Microprocessor and CRT

The Display Microprocessor interfaces the memory to the CRT display. For display purposes, the memory is converted into a continuous 80 character wide display page. Data are transferred from the memory to the video generator to refresh the screen. Access to the memory is obtained during each character line display to obtain the 80 characters to be displayed on the next line.

The total memory can be assigned as a display page. The display screen is a movable window in the display page as illustrated below:

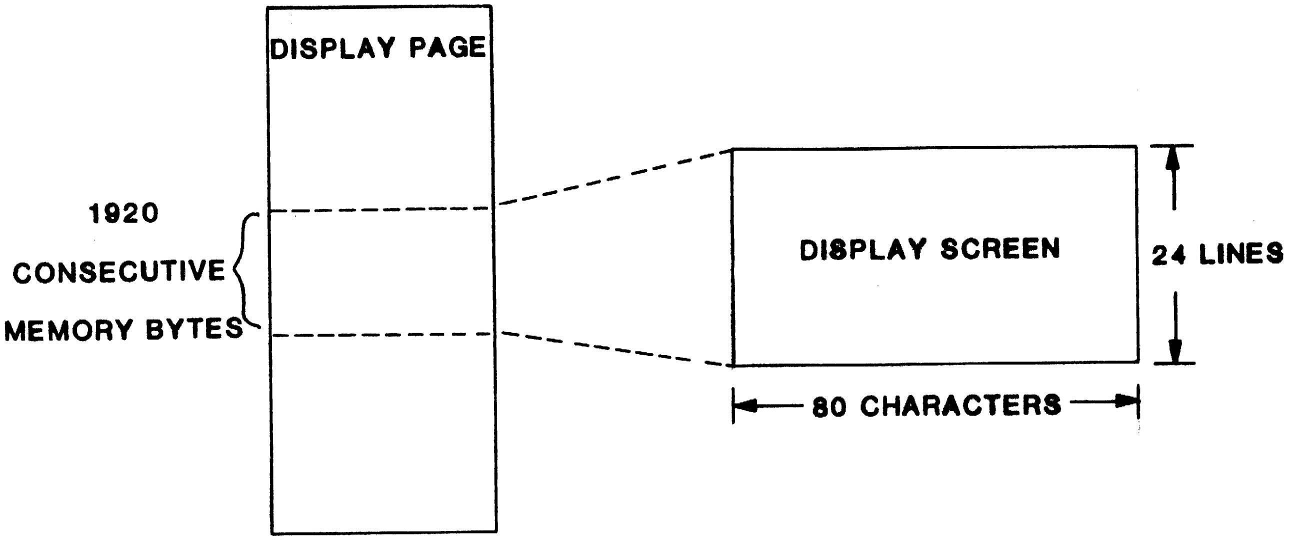


Figure 9R-2. Display Window

The video generator codes are illustrated in Table 8R-1. Tagged Characters (codes from 80 to FF) can be displayed with special emphasis as described in Display Status Commands.

OP-1/R STANDARD FONT

BITS				B7 →	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
				B6 →	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
				B5 →	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				B4 →	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
B3	B2	B1	B0	COL / ROW	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0	→	¶	sp	Ø	@	P	\	p	→	¶	sp	Ø	@	P	\	p
0	0	0	1	1	↓	▁	!	1	A	Q	a	q	↓	▁	!	1	A	Q	a	q
0	0	1	0	2	÷	1/4	"	2	B	R	b	r	÷	1/4	"	2	B	R	b	r
0	0	1	1	3	▶	1/2	#	3	C	S	c	s	▶	1/2	#	3	C	S	c	s
0	1	0	0	4	≤	£	$	4	D	T	d	t	≤	£	$	4	D	T	d	t
0	1	0	1	5	≥	┌	%	5	E	U	e	u	≥	┌	%	5	E	U	e	u
0	1	1	0	6	≠	┬	&	6	F	V	f	v	≠	┬	&	6	F	V	f	v
0	1	1	1	7	¢	┐	'	7	G	W	g	w	¢	┐	'	7	G	W	g	w
1	0	0	0	8	Λ	├	(	8	H	X	h	x	Λ	├	(	8	H	X	h	x
1	0	0	1	9	■	┼	)	9	I	Y	i	y	■	┼	)	9	I	Y	i	y
1	0	1	0	A	¬	┤	*	:	J	Z	j	z	¬	┤	*	:	J	Z	j	z
1	0	1	1	B	/	└	+	;	K	[	k	{	/	└	+	;	K	[	k	{
1	1	0	0	C	:=	┴	,	<	L	\	l	¦	:=	┴	,	<	L	\	l	¦
1	1	0	1	D	◀	┘	–	=	M	]	m	}	◀	┘	–	=	M	]	m	}
1	1	1	0	E	│	─	.	>	N	↑	n	~	│	─	.	>	N	↑	n	~
1	1	1	1	F	sp	sp	/	?	O	←	o	≣	LEOL	FLEOL	/	?	O	←	o	≣
					UNTAGGED								TAGGED							

The standard character font includes limited line drawing capability.

Table 9R-1-1 Video Generator Codes

DISPLAY COMMANDS

Display commands are passed to the Display Microprocessor via memory locations 0800 to 0806 as follows:

	7	6	5	4	3	2	1	0		
0800	0		0-79						CURSOR	CURSOR POSITION COMMANDS
0801	0	0	0	0-23					CURVRT	
0802	DISPLAY STATUS								DISPLAY	
0803	0 TO MAX MEMORY LOCATION								HOMEH	DISPLAY WINDOW COMMANDS
0804									HOMEL	
0805	0 TO MAX MEMORY LOCATION								WRAPH	
0806									WRAPL	

Table 9R-1-2 Display Commands

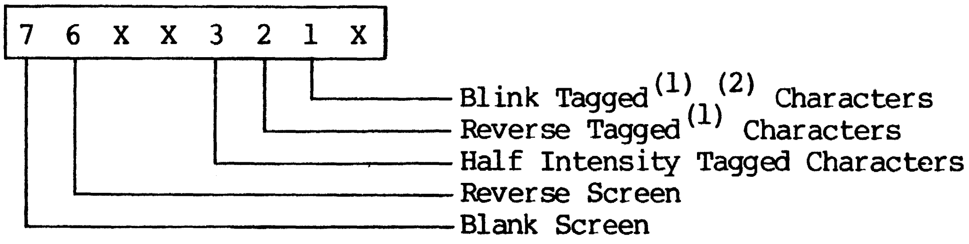
CURSOR POSITION COMMANDS

CURHOR - LOCATION 0800 Cursor Horizontal Position (0-79)

CURVRT - LOCATION 0801 Cursor Vertical Position (0-23)

DISPLAY STATUS COMMANDS

DISPLY - LOCATION 0802 Display Status in the following format:


7 6 X X 3 2 1 X
Blink Tagged(1) (2) Characters
Reverse Tagged(1) Characters
Half Intensity Tagged Characters
Reverse Screen
Blank Screen

(1) Characters from Table 9R-1 with codes from 80 to FF are defined as Tagged Characters.

(2) Simultaneous Blink and Half intensity tagged characters will be displayed Half intensity only.

DISPLAY WINDOW COMMANDS

HOMEH - LOCATION 0803	Home Position address (High)
HOMEL - LOCATION 0804	Home Position address (Low)
WRAPH - LOCATION 0805	Wrap Position address (High)
WRAPL - LOCATION 0806	Wrap Position address (Low)

The display memory is illustrated in Figure 9R-3. Note that the size of the page is variable by changing the WRAP command. The last character in the display buffer should be the last physical location in memory in order to enable wraparound capabilities. The display window may be moved by changing the HOME command.

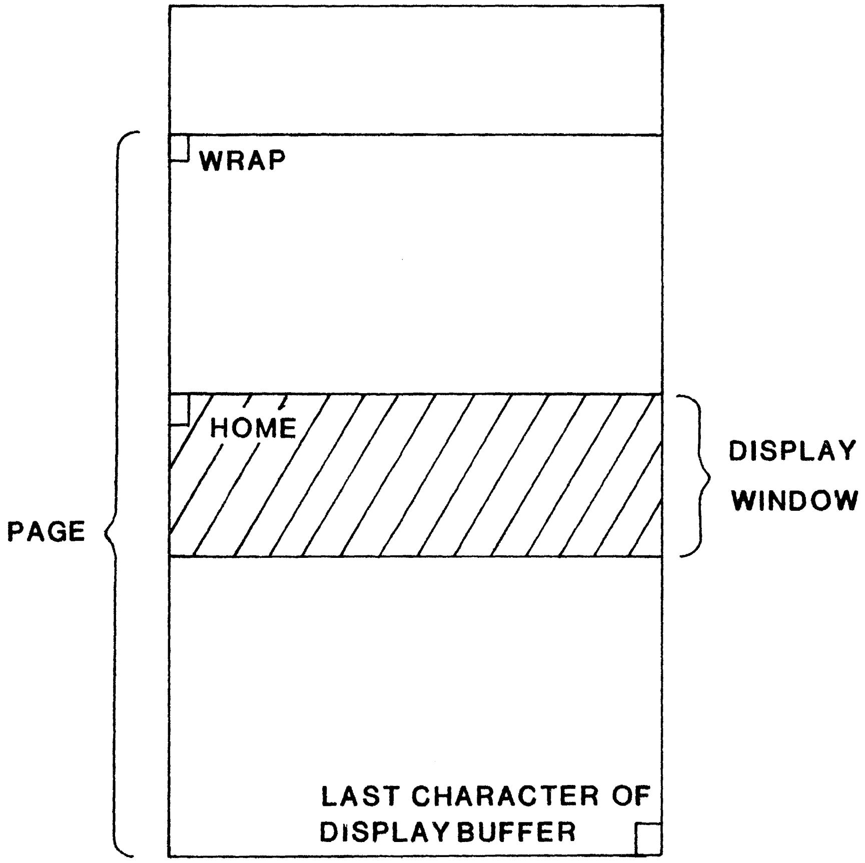


Figure 9R-3. Display Memory

If the HOME position is less than 24 lines above the end of the display buffer, the Display Mircroprocessor automatically wraps around to the location specified by the WRAP command as illustrated in Figure 8R-4. The HOME position must be a whole number of lines above the end of the display buffer so that the last character buffer is at the end of a line.

The display microprocessor refreshes memory as it accesses data for the T.V. screen. The 16K RAM used on the OP-1/R requires 128 consecutive accesses; two consecutive character rows must be read to refresh memory. Therefore, the WRAP address must be limited so address (XXXX XXXX XX00 0000) to (XXXX XXXX XX10 1111) are accessed on the first row after the screen wraps.

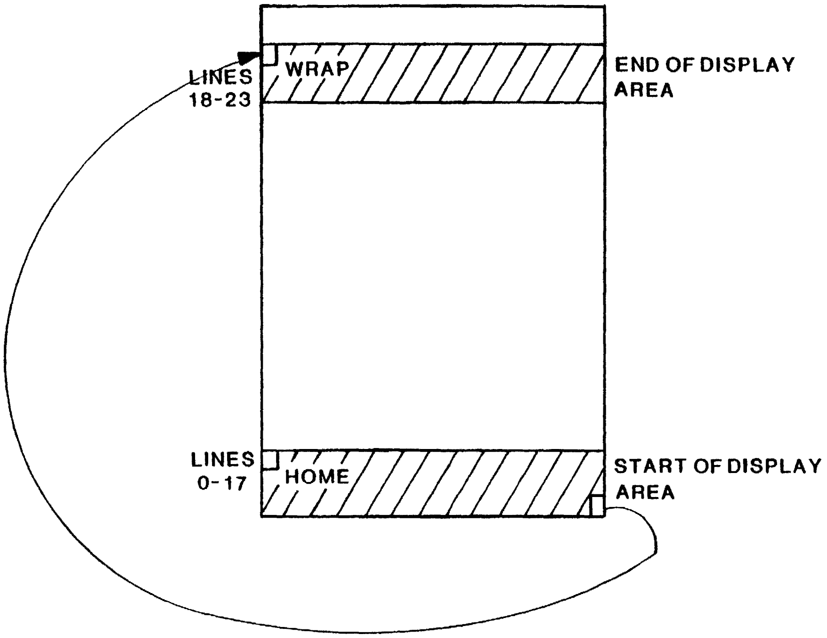


Figure 9R-4. Display Window with Wraparound

DISPLAY BLANKING

The display may be blanked line-by-line by selectively storing an end of line character in the body of the display data. Two characters are implemented:

LEOL - Logical End of Line Character - Code 8F: Unconditionally blanks all characters appearing after it until the end of the display line.

FLEOL - Forms Logical End of Line Character - Code 9F: Blanks untagged characters appearing after it until the first LEOL or end of the display line, whichever appears first.

TIMING

Transfers are made by the Display Microprocessor to the video generator on a cycle steal basis transparent to the CPU operation. The timing periods are:

1. Synchronization wait: The Display Microprocessor requests access to the memory once during each character line display to obtain the 80 characters to be displayed on the next character line. The CPU is allowed to complete the current cycle before access is granted. The synchronization delay requires 0 - 2 usec.

2. Transfer of the 80 characters of a line: 55 usec.

3. Transfer of display commands: 15 usec.

4. The display screen is refreshed every 16,666 usec. The following calculation determines the ratio of time the CPU can use the memory to the total available time.

$$\frac{16{,}666 \text{ usec} - (26 \times 55 + 15)}{16{,}666} = .913 \quad (50 \text{ or } 60 \text{ Hz}$$

SUMMARY OF DISPLAY SPECIFICATIONS

Screen Size:	14-inch diagonal
Screen Capacity:	1920 characters
Display Format:	24 lines of 80 characters
Character Size:	0.21 x 0.09 inch
Character Generation:	Dot Matrix: 7 x 9 within a 9 x 12 matrix
TV Raster:	312 lines, non-interlaced
Refresh Rate:	50 or 60 times per second
Phosphor:	P4 White or P31 Green
Displayable Character Set:	128 Characters
Cursor:	Blinking reversed video square

Program Controlled Functions:

Black on White
Size of Page
Erase and Edit
Blink Characters
Half Intensity Characters
White on Black
Window Location
Roll and Scroll
Reverse Characters

SECTION 10R

ALTERNATE I/O ADAPTER

The Alternate I/O Adapter interfaces the CPU I/O bus to a secondary external device such as a printer or second communication line. All I/O functions are under program control. The CPU can be interrupted by the Alternate I/O Adapter after completion of a transfer of every character.

The following configurations are available. They determine which device attaches to the terminal.

- A bit serial Asynchronous Adapter with RS232 interface
- A bit serial Asynchronous Adapter with 20ma current loop interface
- A bit serial Asynchronous Adapter with 2 wire direct interface
- A printer adapter with parallel interface

In addition, the alternate I/O adaptor may be equipped with two eight bit data switches that are similar in function to the Fixed Data switches described in Section 5R.

ALTERNATE ASYNCHRONOUS I/O ADAPTER

The Alternate Asynchronous I/O Adapter connects the CPU I/O bus to an external communications modem. Operation is performed in full duplex in a bit serial start-stop format. The adapter is designed to operate at 50 to 19,200 bits-per-second. All communication functions are program controlled.

Three interfaces are available for use with the adapter (Factory installation only) as follows:

1. EIA RS 232 C
2. 20ma current loop
3. 2-wire direct

For 20ma current loop interface the baud rate is limited to 9600 bits per second.

The CPU can be interrupted by the adapter after completion of every character received or transmitted if the adapter is not in Ring Mode. In Ring Mode only the Ring Detector signal can interrupt the CPU.

COMMANDS

Commands to the Alternate Asynchronous I/O Adapter may be executed if it has been selected by the CPU as the active I/O device. The adapter will remain selected until a different I/O device selection is made.

SELECT

Command: SEL

Command Byte: 2B

Selects the Alternate Asynchronous Adapter for I/O operations.

STOP

Command: DVCL

Command Byte: NONE

Resets the Alternate Asynchronous I/O Adapter and aborts any current activity. DVCL should be used prior to issuing any operational command when the status of the Alternate Asynchronous Adapter is uncertain. The DVCL command will reset the Character Received and Available flag, the Parity, Overrun or Framing Error flag, the output control signals to the modem, the Input Data Mode, Ring Mode and the Low Data Bit Mode. The DVCL command also sets the Character Needed for Transmission flag.

STATUS

Command: IFL

Status Byte:

Bit 7 Character Received and Available.
Bit 6 Character Needed for Transmission.
Bit 5 Parity, Overrun or Framing Error. This bit remains high until the next character is received.
Bit 4 Clear to Send signal is on.
Bit 3 Supervisory Received Data signal is on.
Bit 2 Carrier Detector signal is on.
Bit 1 Data Set Ready signal is on.
Bit 0 Ring Detector signal is on.

Loads the accumulator with an operational status byte from the Alternate Asynchronous I/O Adapter, provided that the Input Status Mode was previously selected by the COM2 instruction.

OUTPUT

Command: OUT

Transfers a data byte from the accumulator to the Alternate Asynchronous I/O Adapter for transmission. It should be issued only after the adapter has indicated that a character is needed for transmission. The Character Needed for Transmission status bit will be reset until the character is transmitted and a new character is required.

INPUT

Command: IFL

Transfers a received data byte from the Asynchronous I/O Adapter to the accumulator, provided that the Input Data Mode was previously set by the COM2 instruction. It should be issued only after the adapter has indicated that a received character is available. The Input Command resets the Character Received and Available Flag.

SET MODEM

Command: OFL

Command Byte:

Bit 3 Break Transmitted Data.
Bit 2 Hold Supervisory Transmit Data signal on.
Bit 1 Hold Data Terminal Ready signal on.
Bit 0 Hold Request to Send signal on.

SET COMMUNICATIONS PARAMETERS

Command: COM1

Command Byte:

Bit 7 Two Stop Bits (vs one).
Bit 6 Eight Data Bits (vs. 7) or 6 Data Bits (vs. 5) (See COM3).

Bit 5 No Parity.
Bit 4 Even Parity (vs. Odd).
Bit 3
Bit 2 Baud rate selection for both transmit and receive in the
Bit 1 following format:
Bit 0

BIT	BAUD RATE
3 2 1 0	
0 0 0 0	50
0 0 0 1	75
0 0 1 0	110
0 0 1 1	134.5
0 1 0 0	150
0 1 0 1	300
0 1 1 0	600
0 1 1 1	1200
1 0 0 0	1800
1 0 0 1	2000
1 0 1 0	2400
1 0 1 1	3600
1 1 0 0	4800
1 1 0 1	7200
1 1 1 0	9600
1 1 1 1	19200

Establish communications parameters by transferring a command byte from the accumulator to the Asynchronous I/O Adapter. The baud rate for both transmit and receive are set equal by this command. A subsequent COM3 command may change the baud rates for the receive only so as to have different baud rates for transmit and receive. The program should ignore the most significant bits of the data which are not used when 5, 6 or 7 data bits modes are used.

SET INPUT MODE

Command: COM2
Command Byte: Bit 0 = 1 SET INPUT DATA MODE
Bit 0 = 0 SET INPUT STATUS MODE

Conditions the adapter to interpret the next IFL command. DVCL will set Input Status Mode.

AUXILIARY COMMAND

Command: COM3
Command Byte: See Below

Bit 7	Low Data Bit Mode
Bit 6	Ring Detect Mode
Bit 5	Enable Transmit Interrupt
Bit 4	
Bit 3	Receive baud rate selection as per table under COM1 description
Bit 2	
Bit 1	
Bit 0	

Bit 7 sets the bits per character to 6 or 5 as opposed to 8 or 7.

Bit 6 sets the Ring Detect Mode which locks out the interrupts from the character received and character needed flags and instead allows the interrupt to be set by the Ring Detector Signal. The Data Terminal Ready signal should be off in this mode so that the Ring Detector Signal will continue until it is detected by the Status Command (IFL).

Bit 5 allows character needed for transmission signal (IFL Bit 6) to generate an interrupt to the CPU.

Bits 0 thru 3 set the receive data baud rate independent of the transmit baud rate. The COM1 commands set the baud rate for both transmit and receive.

The Low Data bit and Ring Modes are both reset by the DVCL command. Resetting the Ring Mode also resets the Interrupt caused by the Ring Detector Signal.

NOTE: The hardware does not allow bit 7 and bits 0-3 to be chosen with the same COM3 command. In order to choose both the low data bit mode and a different Transmit baud rate from Receive rate, the software must be in the following sequence.

1) A COM3 choosing Bit 7 high for 6 (5) data bits family.

2) A COM1 choosing either 6 or 5, and setting the Transmit/Receive combined baud rate.

3) A COM3 choosing the Receive baud rate (and continuing to hold bit 7 high).

INTERRUPT CONTROL

The Alternate Asynchronous I/O Adapter provides a single interrupt request whenever a character is received and available or when a character is needed for transmission or a ring signal is received when the adapter is in Ring Detect Mode. The Status command must be used to interrogate status bits 6, 7 and 0 to determine the source of the interrupt. The priority level is 4.

PRINTER ADAPTER

The Printer Adapter interfaces the CPU I/O bus to a parallel interface printer. All printer functions are program controlled. The CPU can be interrupted by the printer adapter whenever it is ready to receive a character. For detailed interface information see Table 9R-1.

COMMANDS

Commands to the Printer Adapter may be executed only if the adapter has been selected by the CPU as the active I/O device. The adapter will remain selected until a different I/O device selection is made.

SELECT

Command:	SEL
Command Byte:	2B

Selects the print controller for I/O operation.

PRINT

Command:	OUT

Transfer a character from the accumulator to the printer via the Printer Adapter. The OUT command also resets the interrupt request and the character request status bit.

STOP

Command:	DVCL
Command Byte:	None

Resets the adapter and aborts any current activity. This command should be used prior to issuing a print command when the status of the controller is uncertain, such as at the start of a program.

STATUS

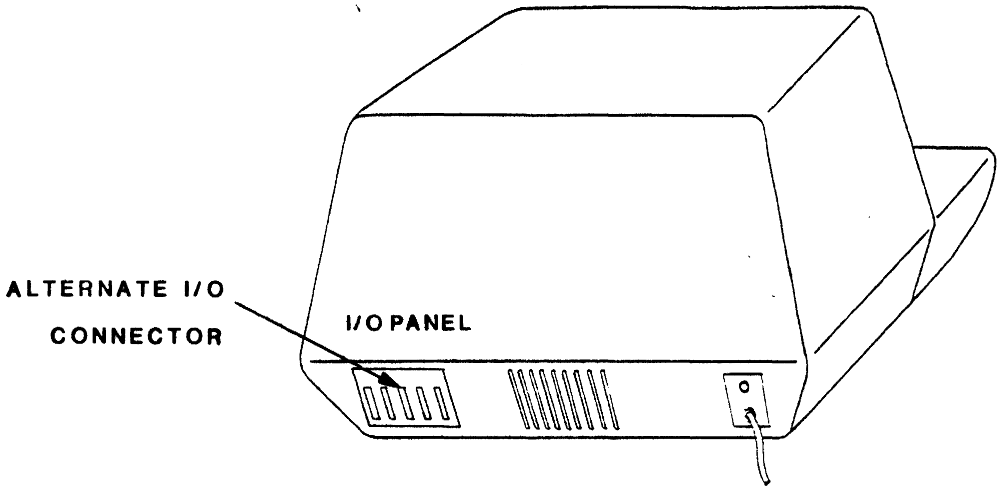
Command: IFL
Status Byte:

Bit 7	NOT BUSY	Set when the Printer Adapter is ready to receive a new print or control character.
Bit 6	PRINTER SELECTED	Set when the printer has been selected either locally or by CPU command.
Bit 1	PRINTER NOT READY	Set when the printer is not ready to receive data, i.e. it is not connected or is out of paper, etc.
BIT 0	PRINTER BUSY	Set when the printer is executing a print operation or when the printer is ready but not selected.

Loads the accumulator with an operational status byte from the printer adapter.

INTERRUPT CONTROL

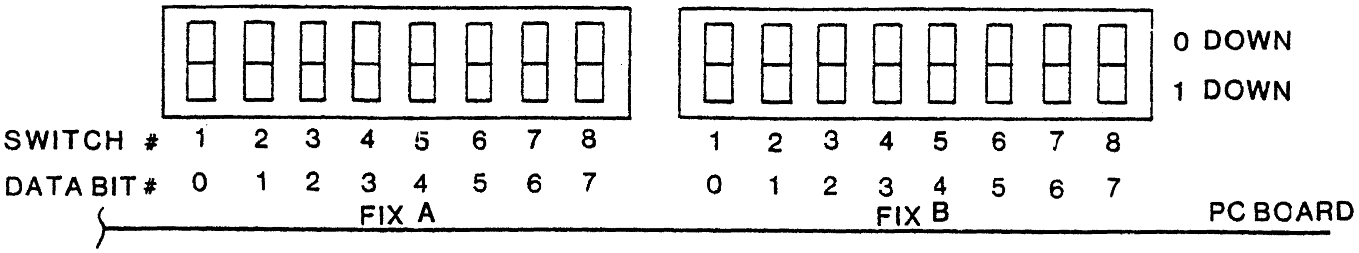
The Printer Adapter provides an interrupt request whenever it is ready to receive a print or control character. The priority level is 4.



FIXED DATA SWITCHES

Two eight-bit Fixed Data Switches can be provided for general programming purposes. The switches can be manually set at the time of installation to specify an identifying address or any other general function such as mode selection.

The switch configuration is illustrated below: (see figure 5R-1 for location)



COMMANDS

Commands to the Fixed Data Switches may be executed if it has been selected by the CPU as the active I/O device. The switches will remain selected until a different I/O device selection is made.

SELECT

Command:	SEL
Command Byte:	1D

Selects the data switches for subsequent input to the CPU by an IFL instruction.

SWITCH SELECT A

Command:	COM1
Command Byte:	None

Selects switch A for subsequent data transfer with the IFL command.

SWITCH SELECT B

Command:	COM2
Command Byte:	None

Selects switch B for subsequent data transfer with the IFL command.

INPUT DATA

Command:	IFL

Transfers the contents of Switch A or Switch B to the accumulator, depending on whether it was preceded by a COM1 or a COM2. The contents of switch bit n is loaded into accumulator bit n.

PIN	CRT	DESCRIPTION
1	AA	Protective Ground
2	BA	Transmitted Data
3	BB	Received Data
4	CA	Request to Send
5	CB	Clear to Send
6	CC	Data Set Ready
7	AB	Signal Ground
8	CF	Carrier Detector
11	SA	Supervisory Transmitted Data
12	SB	Supervisory Received Data
20	CD	Data Terminal Ready
22	CE	Ring Detector

Connector: Cannon DBC-25S

Table 10R-1-1 Alternate I/O Adapter Pin Assignments
EIA/RS232 Signals

This interface satisfies the requirements of the standard asynchronous interface between data terminal and data communications equipment as defined by EIA standard RS 232.

PIN	DESCRIPTION
1	Protective Ground
2	Serial Output Neg.
3	Serial Output Pos.
5	Serial Input Neg.
6	Serial Input Pos.
7	Signal Ground

Connector: Cannon DBC-25S

Table 10R-1-2 Alternate I/O Adapter Pin Assignments
20 ma Current Loop Signal

This interface uses a 20 ma current source supplied by the host computer. The driver will switch currents up to 20 ma (voltage not to exceed 30 VDC). The opto-isolator received requires a minimum of 20 ma with a maximum of 25 ma. The data lines can be strapped to allow current flow to be either a mark or space condition.

PIN	DESCRIPTION
1	Protective Ground
4	Data (Bidirectional)
7	Signal Ground

Connector: Cannon DBC-25S

Table 10R-1-3 Alternate I/O Adapter Pin Assignments
2 Wire Direct Signals

This is a half duplex interface utilizing two wires: a data line that switches from -12V (mark) to ground (space) and a ground line.

NOTE: RTS turns off received data.

PIN	DESCRIPTION
1	Protective Ground
2	Printer Acknowledge
3	Data Strobe
4	Printer Selected Signal
5	Printer Busy
6	Printer Fault
7	Signal Ground
8	Data Bit 2
9	Data Bit 5
10	Data Bit 6
11	Data Bit 7
15	Data Bit 4
17	Data Bit 3
20	Data Bit 0
21	Printer Prime
22	Data Bit 1

Connector: Cannon DBC-25S

Table 10R-1-4 Alternate I/O Adapter Pin Assignments
Printer Adapter

This is a parallel output interface using TTL compatible signals. Each byte is output together with a 1.5 usec strobe and must be acknowledged before the next byte.

SECTION 11R

OP-1/RW

WORD PROCESSING DISPLAY MICROPROCESSOR AND CRT

The CRT display unit consists of a Display Microprocessor, video generator and a 14-inch diagonal, non-glare CRT. All display functions are program-controlled. The Display Microprocessor features a high resolution display with a 7 x 12 dot matrix. A character set of 128 characters is available, user definable at manufacturing time.

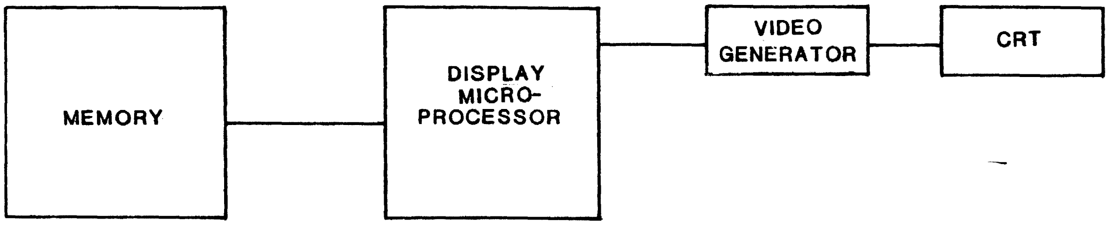


FIGURE 11R-1 DISPLAY MICROPROCESSOR AND CRT

The Display Microprocessor interfaces memory to the CRT display. For display purposes, the memory is converted into a continuous 80 or 160 character wide display page. Data is transferred from the memory to the video generator to refresh the screen. Access to the memory is obtained during each character line display to obtain the 80 characters to be displayed on the next line.

The total memory can be assigned as a display page. The display screen is a moveable window in the display page as illustrated below:

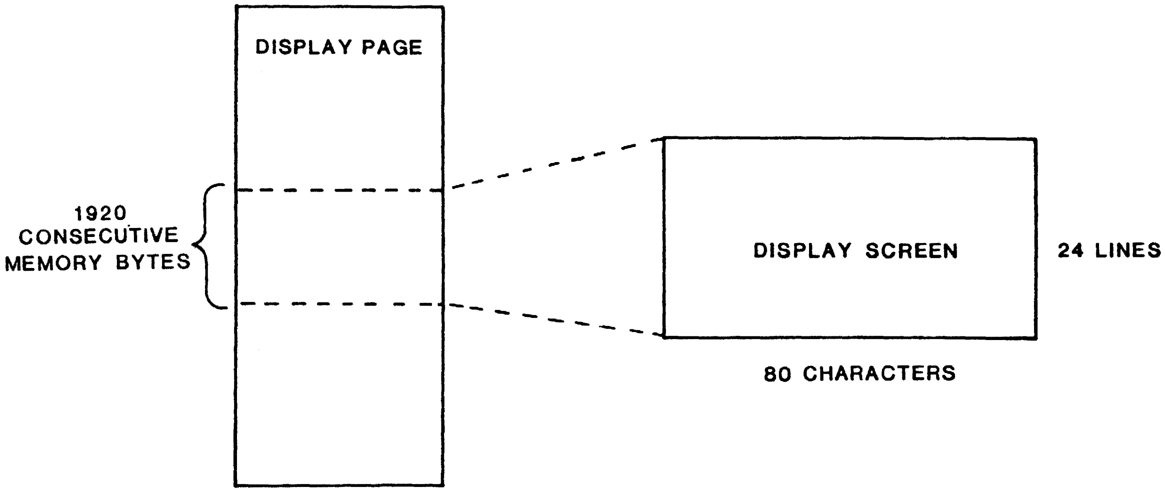


FIGURE 11R-2 DISPLAY WINDOW

CHARACTER GENERATOR

The display Microprocessor character fonts are stored in ROM and can be user defined at time of manufacture. A distinct, user definable character will be generated for each data code from 00 to 7F. This character set will repeat for codes 80 to FF. However, for these codes the characters are considered tagged and can be presented as normal characters, or with special emphasis according to the DISPLY1 command.

Any character can be included in a family of Suppressable Characters at time of manufacture. Suppressable Characters can be displayed as spaces via the DISPLY1 command.

CURSOR DISPLAY

The cursor is displayed as a Blinking reversed block at the horizontal and vertical location specified by CURHOR and CURVRT.

OPERATING MODES

The Display Microprocessor always operates in the Visual Enhancement mode which provides three different enhancement capabilities needed in text editing applications. Enhancements are provided on a text string by means of start and end delimiter characters where all characters between a pair of delimiters are highlighted. Two sub-modes are available:

a) 80 column text page

   Allows the screen to be scrolled up and down through the full memory and the wrap at the end of memory to be utilized fully.

b) 160 column text page

   Allows the 80 column display window to be scrolled horizontally across the 160 column text page. The use of the wrap at the end of the memory is limited. If the display page is allowed to wrap at the end of memory, all enhancement start characters prior to the end of memory must be kept within an 80 column display window or an unpredictable display of enhancement will result as the window is horizontally moved across the area.

DISPLAY COMMANDS

Display commands are passed to the Display Microprocessor via memory locations 0800 to 0806 as follows:

LOCATION	7	6	5	4	3	2	1	0	NAME
0800	0	0-79							CURHOR
0801	*	0	0	0-23					CURVRT
0802		0		0		0		0	DISPLY1
0803									HOMEH
0804									HOMEL
0805									WRAPH
0806									WRAPL

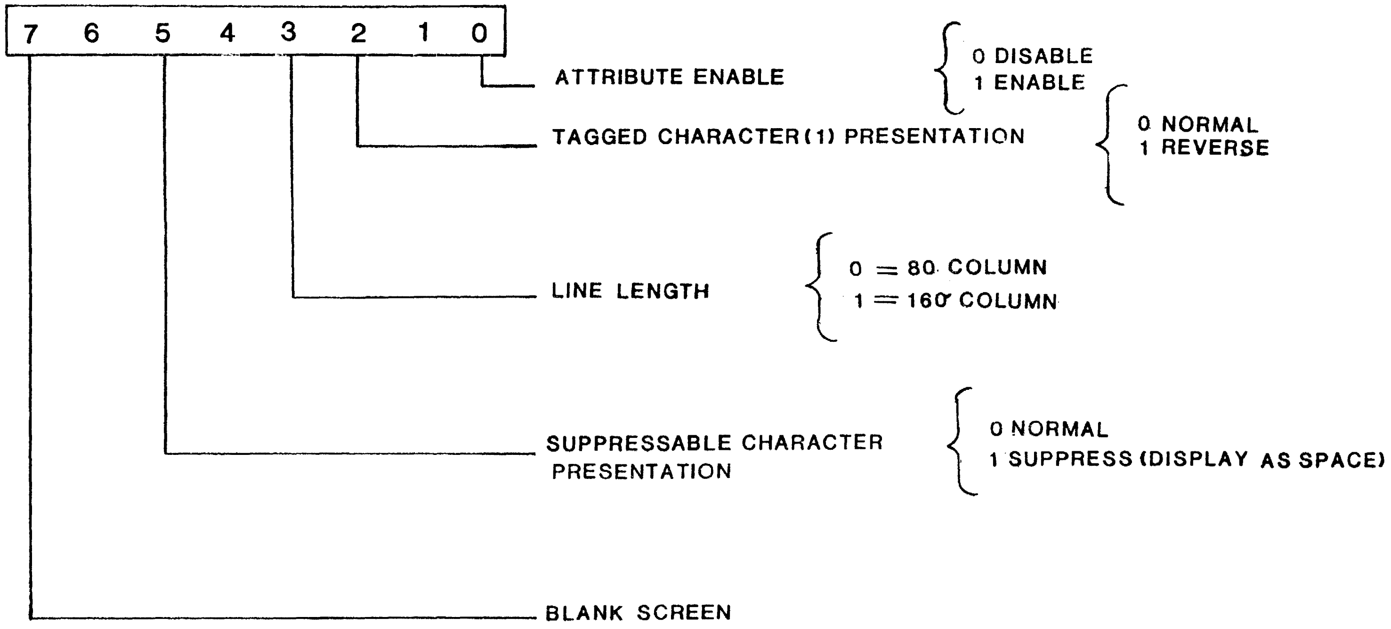
FIGURE 11R-3 DISPLAY COMMAND

CURSOR POSITION COMMANDS

CURHOR - LOCATION 0800	Cursor Horizontal Position (0-79)
CURVRT - LOCATION 0801	Cursor Vertical Position (0-23)
*Bit 7=0	Start enhancement from HOME position
Bit 7=1	Start enhancement from WRAP location

DISPLAY STATUS COMMANDS

DISPLY1 - LOCATION 0802



(1) CHARACTERS WITH CODES FROM 80 TO FF ARE DEFINED AS TAGGED CHARACTERS.

DISPLAY WINDOW COMMANDS

HOMEH - LOCATION 0803	Home Position Address (High)
HOMEL - LOCATION 0804	Home Position Address (Low)
WRAPH - LOCATION 0805	Wrap Position Address (High)
WRAPL - LOCATION 0806	Wrap Position Address (Low)

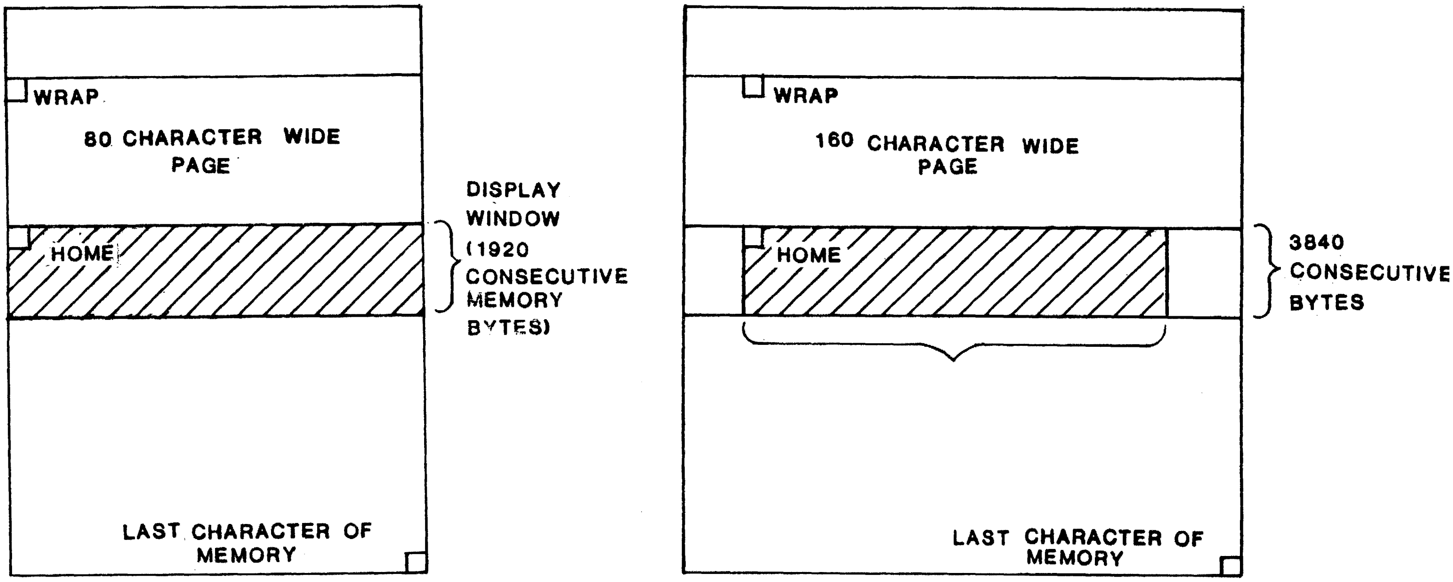


FIGURE 11R-4 DISPLAY MEMORY

The HOMEH and HOMEL bytes are concatenated to define the address of the first character visible in the display window. A line scrolling effect is created by changing this address by whole line increments (80 or 160). Horizontal scrolling, in the 160 wide mode, is effected by changing the address by single increments.

If the HOME position is less than 24 lines above the end of the memory, the Display Microprocessor automatically wraps around to the location specified by the WRAP command as illustrated in Figure 11R-5. The HOME position must be a whole number of lines above the end of the display buffer so that the last character in the display buffer is at the end of a line.

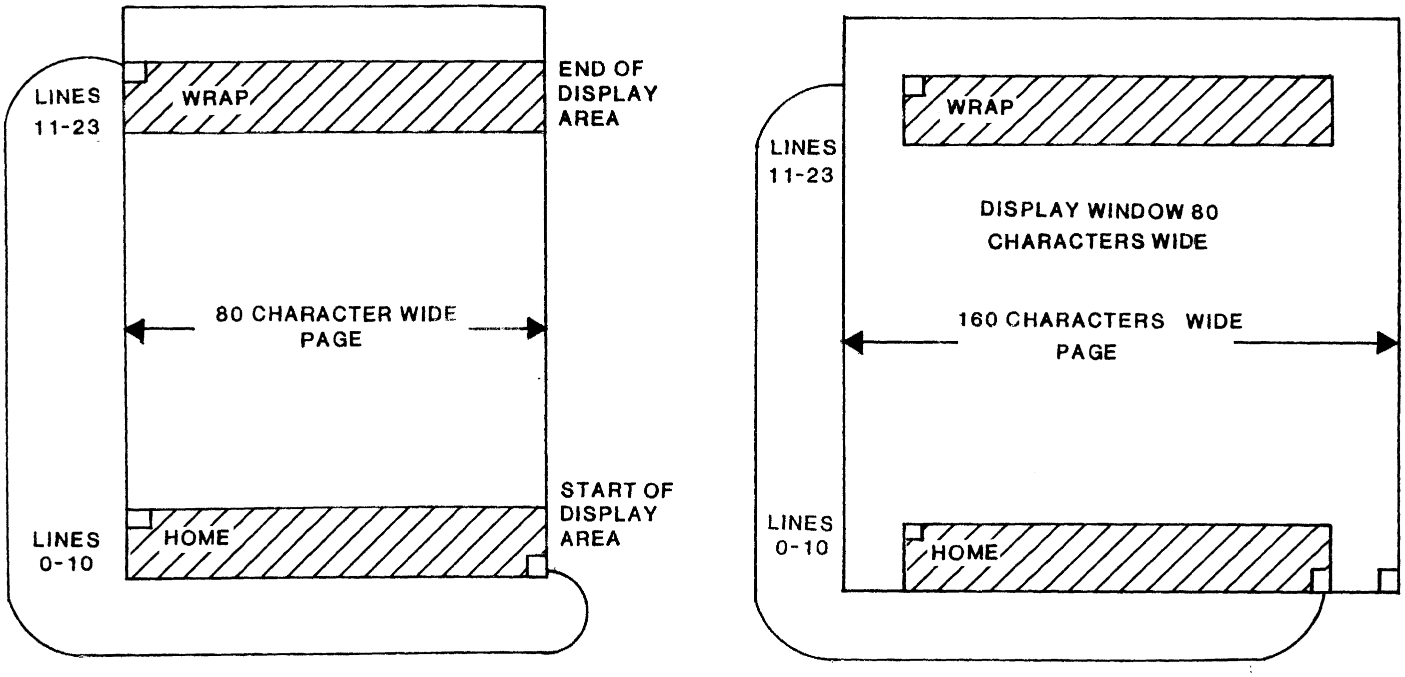


FIGURE 11R-5 DISPLAY WINDOW WITH WRAPAROUND

The display microprocessor refreshes memory as it accesses data for the T.V. screen. The 16K RAM used on the OP-1/R requires 128 consecutive accesses; two consecutive character rows must be read to refresh memory. Therefore, the WRAP address must be limited so address (XXXX XXXX XX00 0000) to (XXXX XXXX XX10 1111) are accessed on the first row after the screen wraps.

VISUAL ENHANCEMENT SPECIFICATION

At time of manufacture the various suppressible and enhancement character codes must be specified.

Three attributes are available.

A. **Underline**

Underlines a group of characters that are delimited between the starting code and stop code.

B. **Bright**

Normal intensity a group of characters that are delimited between the starting code and stop code.

(NOTE: Standard display is half-intensity)

C. **Blink**

Blinks a group of characters that are delimited between the starting code and stop code.

NOTES:

1. Any code may be factory assigned as an enhancement delimiter or a suppressable character. All enhancement delimiters are automatically suppressible. More than one character can be assigned to a particular enhancement delimiter.

2. The Display Microprocessor is capable of providing the enhancement even if the starting code of a particular enhancement pair is not visable on the screen. Since the display is a "window" in the memory, this capability is limited to the occurrence of at least one ending delimiter of every group within the display window.

3. Provision is made to allow for using the display window wraparound as continuous or non-continuous text in association with the display enhancement. At the programmer's option (by setting bit 7 of the CURVRT command to 1) the enhancement capability as described above in 2 will take place beginning at the WRAP location.

TIMING

Transfers are made by the Display Microprocessor to the video generator on a cycle steal basis transparent to the CPU operation.

The timing periods are:

1. Synchronization wait: The Display Microprocessor requests access to the memory once during each character line display to obtain the 80 characters to be displayed on the next character line. The CPU is allowed to complete the current cycle before access is granted.

2. For 80 column operation one block of 80 characters is transferred for each character line.

   a) Transfer time — 55 usec, 8.3% of CPU time

3. For 160 column operation two blocks of 80 characters are transferred for each character line.

   a) Transfer time — 110 usec, 16.6% of CPU time

SUMMARY OF SPECIFICATIONS

Screen Size:	14-inch diagonal
Screen Capacity:	1920 characters
Display Format:	24 lines of 80 characters
Character Size:	0.21 x 0.09 inch
Character Generation:	Dot Matrix: 7 x 9 (upper case) 7 x 11 (lower case)
Cursor:	Blinking Reversed Block
TV Raster:	312 lines, non-interlaced
Refresh Rate:	50 or 60 times per second
Displayable Character Set:	One set of 128 characters

ENHANCEMENT PROM CODING

The enhancement PROM is a 256 x 4 PROM located at A48. The hex code of the selected character is used as the PROM address.

FUNCTION	PROM ADDRESS	PROM CODE (In Hex)
START BOLD	XX	E
STOP BOLD	XX	6
START UNDERLINE	XX	D
STOP UNDERLINE	XX	5
START BLINK	XX	B
STOP BLINK	XX	3
STOP ALL	XX	0
SUPPRESS	XX	7
ALL OTHER ADDRESSES		F

CHARACTER FONT PREPARATION

A program has been designed which allows user defined character fonts to be created and modified. Refer to the Miscellaneous 8080 Programmer's Manual for directions.

The diskette containing the character description should be sent to ONTEL CORPORATION to be burned-in a 2716 PROM.

SECTION 12R

SYNCHRONOUS I/O ADAPTER

The Synchronous I/O Adapter connects the CPU I/O bus to an external communications modem. Operation is performed in full duplex in a serial synchronous format. The adapter is designed to operate at up to 50,000 bits-per-second using an EIA RS232C interface. All communications functions are program controlled. The CPU can be interrupted by the adapter after completion of every character received or transmitted.

COMMANDS

Commands to the Synchronous I/O Adapter may be executed if it has been selected by the CPU as the active I/O device. The adapter will remain selected until a different I/O device selection is made.

SELECT

Command: SEL

Command Byte: F0

Selects the Synchronous I/O Adapter for I/O operations.

STOP

Command: DVCL

Command Byte: None

Resets the Synchronous I/O Adapter and aborts any current activity. DVCL should be used prior to issuing any operational command when the status of the Synchronous Adapter is uncertain. The DVCL command will reset the Character Received and Available flag, the Parity, or Overrun Error flag and the Output control signals to the modem (Bits 7 and 5 of IFL and OFL Bits 3, 1 and 0). DVCL also sets the Character Needed for Transmission flag (IFL bit 6).

STATUS

Command: IFL

Status Byte:

Bit 7	Character Received and Available
Bit 6	Character Needed for Transmission
Bit 5	Parity, or Overrun Error. This bit remains High until the next character is received.
Bit 4	Clear to Send signal is On
Bit 3	Not Used
Bit 2	Carrier Detect signal is On
Bit 1	Data Set Ready signal is On
Bit 0	Not Used

Loads the accumulator with an operational status byte from the Synchronous I/O Adapter. Bits 1 and 2 follow the modem signals.

INPUT

Command: INP

Transfers a received data byte from the Synchronous I/O Adapter to the accumulator. It should be issued only after the adapter has indicated that a received character is available. INP resets the Character Received and Available flag.

SET MODEM

Command: OFL

Command Byte:

Bit 3	Break Transmitted Data
Bit 2	Not Used
Bit 1	Data Terminal Ready
Bit 0	Hold Request to Send signal On

SELECT REGISTER

Command:	COM1

Command Byte:

1. HEX 80	Select Transmit Synch Register
2. HEX 40	Select Control Register
3. HEX 20	Select Receive Synch Register
4. HEX 10	Select Transmit Data Register

OUTPUT

Command:	OUT

Transfers a byte from the accumulator to the register in the Synchronous I/O Adapter that has been selected by a COM1 instruction.

1. Transmit Synch Register - Contains the Synchronizing or Idle Character for transmission.

2. Control Register - Defines communication parameters as follows:

Bit 7	1 = Even Parity 0 = Odd Parity
Bit 6	1 = No Parity 0 = Parity

   Bit 5, Bit 4

0	0	5 Data Bits
0	1	6 Data Bits
1	0	7 Data Bits
1	1	8 Data Bits

3. Receive Synch Register - Contains the character that will be detected during Receive as the Synchronization Character.

4. Transmit Data Register - Contains the Data Character to be transmitted. It should be loaded only after the adapter has indicated that a character is needed for transmission. The character needed for transmission flag will be reset until the character is transmitted and a new character is required.

INTERRUPT CONTROL

The Synchronous I/O Adapter provides two interrupt request signals to the CPU:

Priority Level No. 2 (No. 3 is highest) - A character is received and available. Identical to IFL status bit 7.

Priority Level No. 1 - A character is needed for transmission. Identical to IFL status bit 6.

PIN	CKT	DESCRIPTION
1	AA	Protective Ground
2	BA	Transmitted Data
3	BB	Received Data
4	CA	Request to Send
5	CB	Clear to Send
6	CC	Data Set Ready
7	AB	Signal Ground
15	DB	Transmit Clock
17	DD	Receive Clock
20	CD	Data Terminal Ready

Connector: Cannon DBC-25S

Table 12R-1-1 Synchronous I/O Adapter Pin Assignment EIA-RS232C Signals

This interface satisfies the requirements of the standard synchronous interface between data terminal and data communications equipment defined by EIA Standard RS 232C.

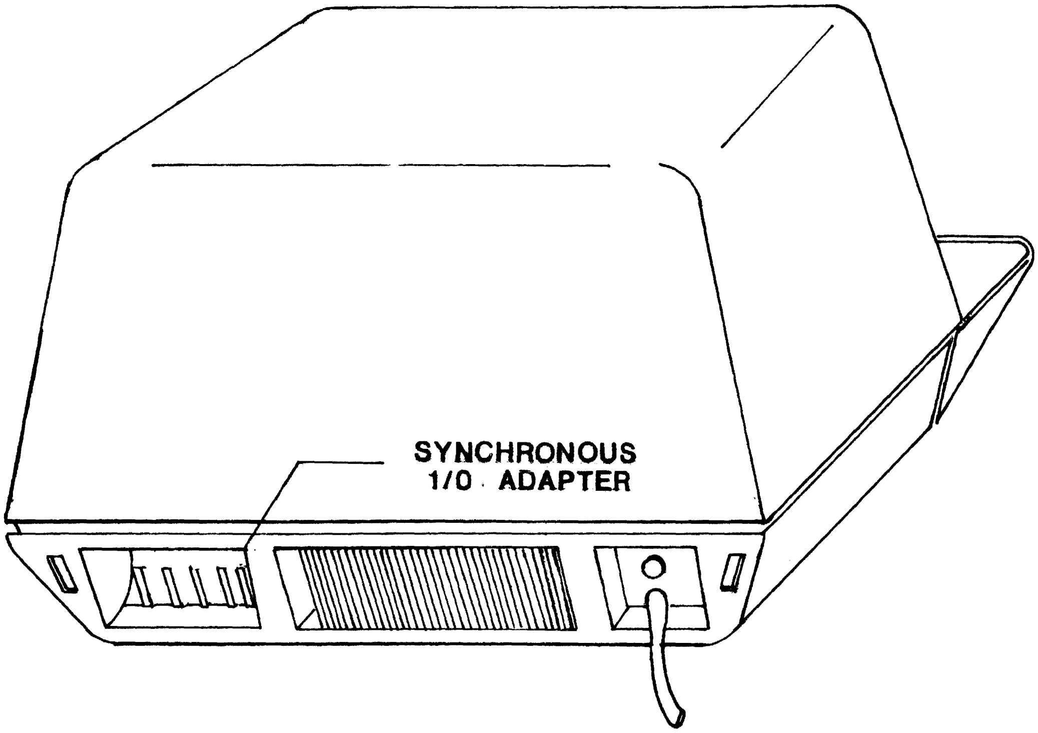


Figure 12R-1. External Connector to Synchronous I/O Adapter

Figure 12R-1 illustrates the Synchronous I/O Adapter connector location.

SECTION 13R

INPUT/OUTPUT MICROPROCESSOR

The Input/Output Microprocessor (IOM) manages all data transfers between the memory and the device controller housed in the OP-1/R. All data transfers are performed on a cycle steal basis transparent to the CPU activity. The CPU is interrupted after the completion of an I/O operation. Data is transferred at a rate of 7 microseconds/byte.

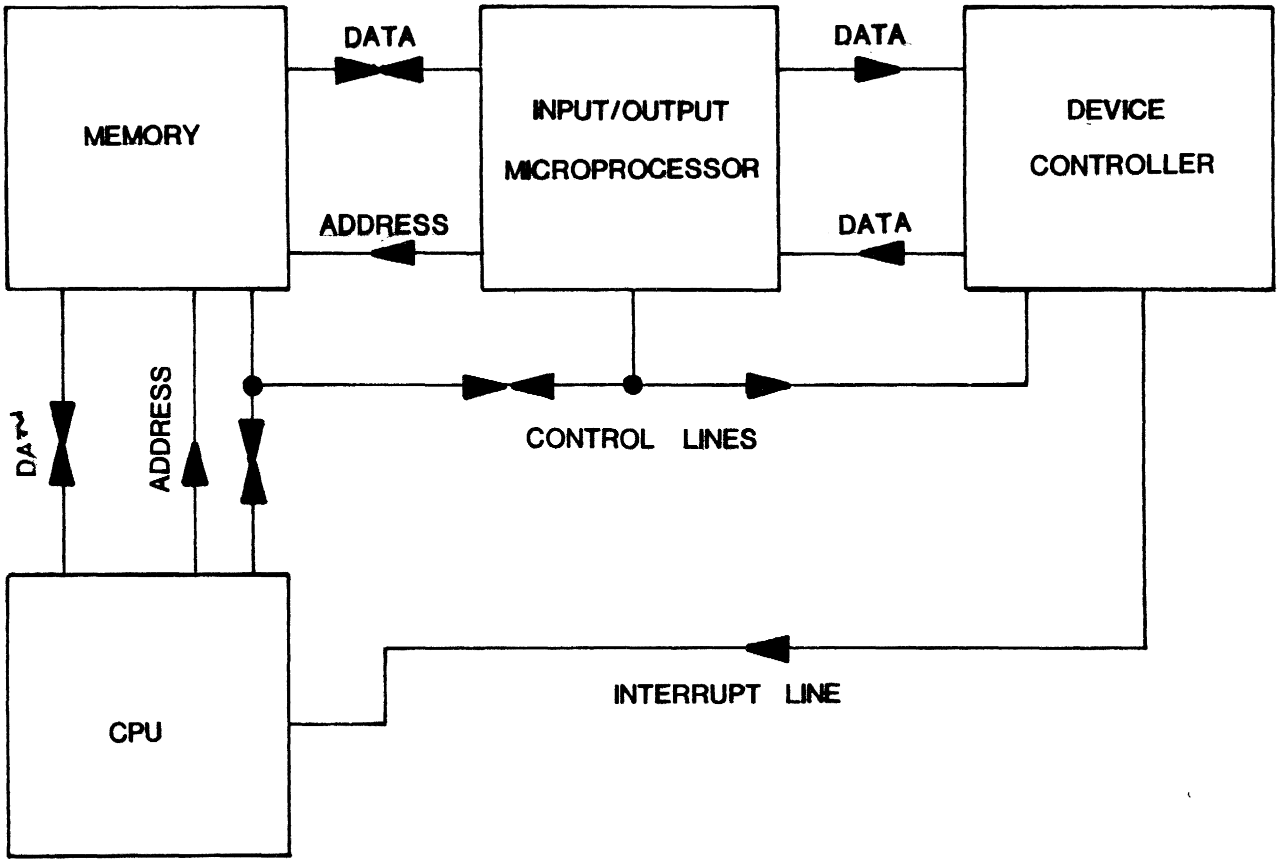


FIGURE 13R-1 BLOCK DIAGRAM

The device controller is wired to the Interrupt Bus and has two sets of reserved memory locations assigned to it.

Standard assignments have been made for purposes of software development only. (See Figure 13R-2). Reserved memory and interrupt assignments may be changed by altering jumpers on the IOM board.

INTERRUPT PRIORITY	COMMAND LOCATIONS (PRIMARY)	COMMAND LOCATIONS (SECONDARY)
7, 6	0808 - 080E	0848 - 084E
5	0810 - 0817	0850 - 0857
4	0818 - 081E	0858 - 085E
3	0820 - 0827	0860 - 0867

FIGURE 13R-2

Refer to OP-1 Reference Manual for Device Controller information.

This single channel IOM will support one Device Controller.

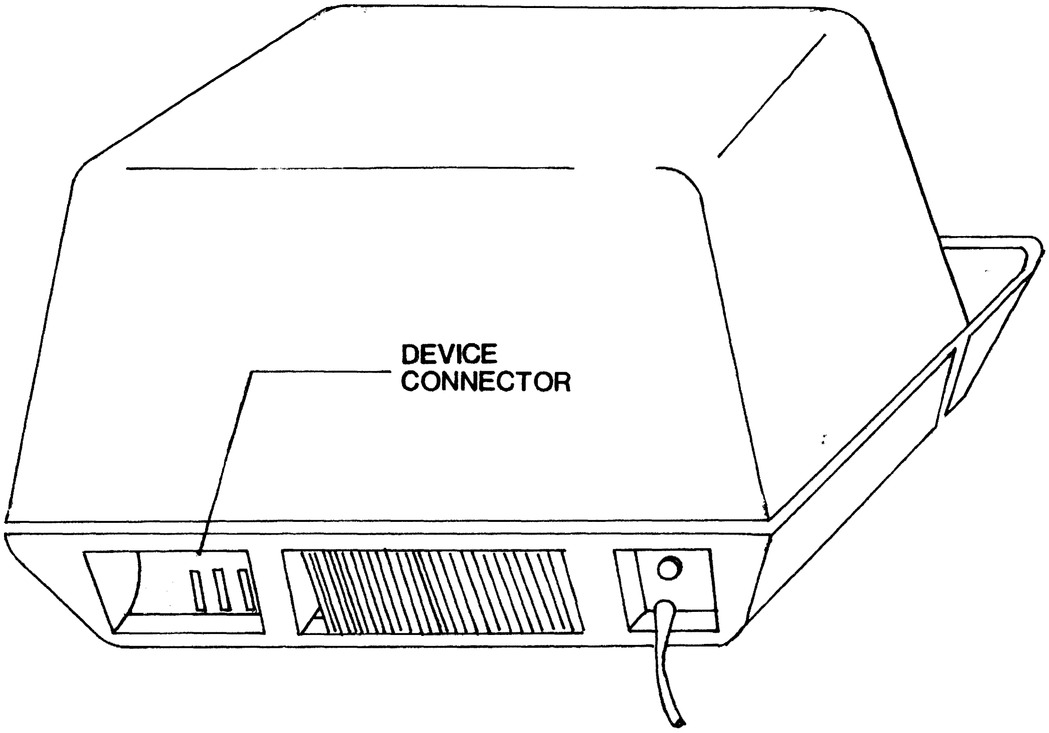
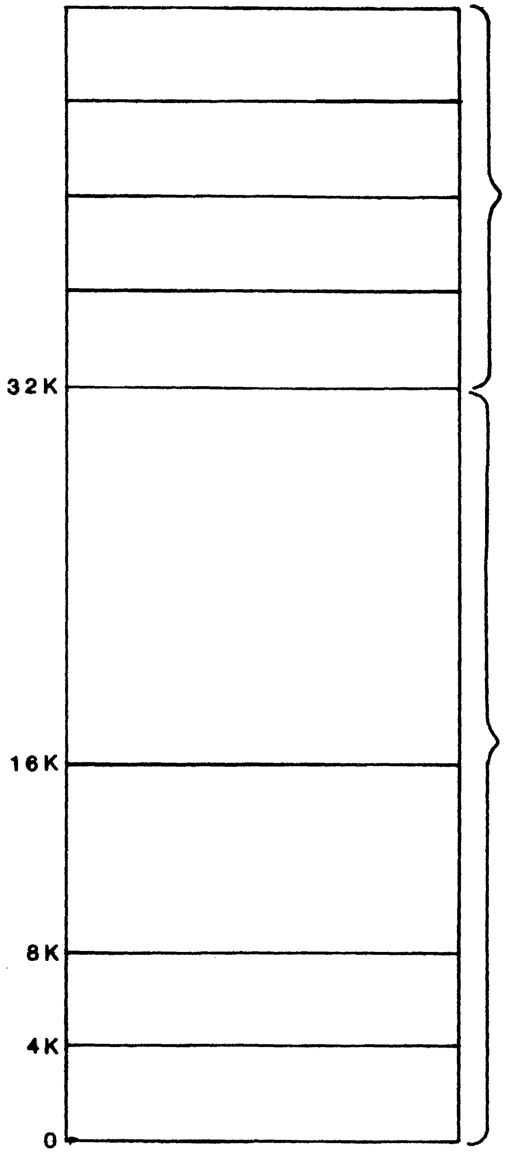


FIGURE 13R-3 EXTERNAL CONNECTOR FOR PERIPHERAL DEVICE

APPENDIX R1

AVAILABLE MEMORY CONFIGURATIONS



READ ONLY MEMORY SECTION

UP to 4 sockets, each capable to be equipped with 1K or 2K byte PROM/EPROM/ROM

READ/WRITE MEMORY SECTION

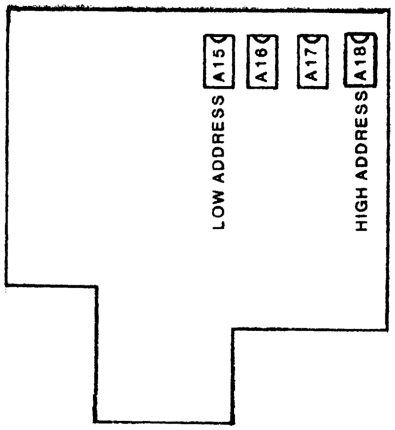
Available Memory Configuration

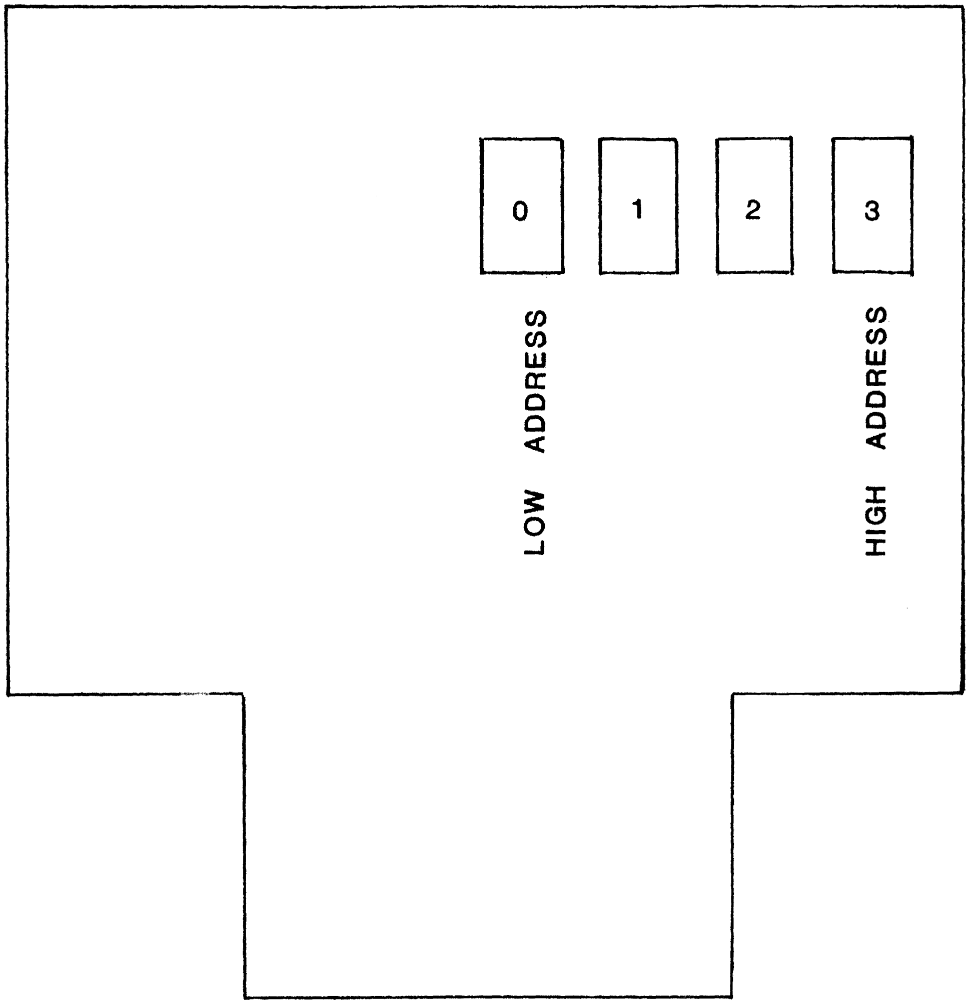
4K Bytes (standard equipment)

8K Bytes

16K Bytes

32K Bytes





	0	1	2	3
5100-1101	A15	A16	A17	A18
-1108	A50	A51	A52	A53
-1109	A50	A51	A52	A53
-1110	A50	A51	A52	A53

READ ONLY MEMORY LOCATION

APPENDIX R-2: INSTRUCTIONS IN OPCODE ORDER

Hex	MACASM	ASM80	
Opcode	Mnemonic	Mnemonic	Operation
00	NOP	NOP	No Operation
01	LXI B	LBC	Load Immediate bytes into BC
02	STAX B	A.@BC	Store at (BC)
03	INX B	INBC	Increment BC
04	INR B	INB	Increment B
05	DCR B	DCB	Decrement B
06	MVI B	I.B	Load Immediate byte into B
07	RLC	RLC	Rotate A Left
08			
09	DAD B	ADBC	Add BC to HL
0A	LDAX B	@BC.A	Load byte at (BC) into A
0B	DCX B	DCBC	Decrement BC
0C	INR C	INC	Increment C
0D	DCR C	DCC	Decrement C
0E	MVI C	I.C	Load Immediate byte into C
0F	RRC	RRC	Rotate A Right
10			
11	LXI D	I.DE	Load Immediate into DE
12	STAX D	A.@DE	Store A at (DE)
13	INX D	INDE	Increment DE
14	INR D	IND	Increment D
15	DCR D	DCD	Decrement D
16	MVI D,	I.D	Load Immediate byte into D
17	RAL	RAL	Rotate A Left Thru Carry
18			
19	DAD D	ADDE	Add DE to HL
1A	LDAX D	@DE.A	Load byte at (DE) Into A
1B	DCX D	DCDE	Decrement DE
1C	INR E	INE	Increment E
1D	DCR E	DCE	Decrement E
1E	MVI E	I.E.	Load Immediate byte into E
1F	RAR	RAR	Rotate A Right Thru Carry
20			
21	LXI H	I.HL	Load Immediate bytes into HL
22	SHLD addr.	HL.@I	Store HL at immed. addr
23	INX H	INHL	Increment HL
24	INR H	INH	Increment H
25	DCR H	DCH	Decrement H
26	MVI H	I.H	Load Immediate byte into H
27	DAA	DAA	Decimal Adjust A
28			
29	DAD H	ADHL	Add HL to HL
2A	LHLD Adr	@I.HL	Load bytes at immed. addr into HL
2B	DCX H	DCHL	Decrement HL
2C	INR L	INL	Increment L
2D	DCR L	DCL	Decrement L
2E	MVI L	I.L	Load Immediate byte into L
2F	CMA	CMA	Complement A
30			
31	LXI SP	I.SP	Load Immediate bytes into SP
32	STA addr	A.@I	Store at immed. addr.

Hex	MACASM	ASM80	
Opcode	Mnemonic	Mnemonic	Operation
33	INX SP	INSP	Increment SP
34	INR M	INM	Increment byte at (HL)
35	DCR M	DCM	Decrement byte at (HL)
36	MVI M	I.M	Copy Immediate byte to (HL)
37	STC	STC	Set Carry Flag
38			
39	DAD SP	ADSP	Add SP to HL
3A	LDA addr	@I.A	Load byte at immed. addr. into A
3B	DCX SP	DCSP	Decrement SP
3C	INR A	INA	Increment A
3D	DCR A	DCA	Decrement byte into A
3E	MVI A	I.A	Load Immediate byte into A
3F	CMC	CMC	Complement Carry Flag
40	MOV B,B		Copy B to B
41	MOV B,C	C.B	Copy C to B
42	MOV B,D	D.B	Copy D to B
43	MOV B,E	E.B	Copy E to B
44	MOV B,H	H.B	Copy H to B
45	MOV B,L	L.B	Copy L to B
46	MOV B,M	M.B	Load byte at (HL) into B
47	MOV B,A	A.B	Copy A to B
48	MOV C,B	B.C	Copy B to C
49	MOV C,C		Copy C to C
4A	MOV C,D	D.C	Copy D to C
4B	MOV C,E	E.C	Copy E to C
4C	MOV C,H	H.C	Copy H to C
4D	MOV C,L	L.C	Copy L to C
4E	MOV C,M	M.C	Load byte at (HL) into C
4F	MOV C,A	A.C	Copy A to D
50	MOV D,B	B.D	Copy B to D
51	MOV D,C	C.D	Copy C to D
52	MOV D,D		Copy D to D
53	MOV D,E	E.D	Copy E to D
54	MOV D,H	H.D	Copy H to D
56	MOV D,M	M.D	Load byte at (HL) into D
57	MOV D,A	A.D	Copy A to D
58	MOV E,B	B.E	Copy B to E
59	MOV E,C	C.E	Copy C to E
5A	MOV E,D	D.E	Copy D to E
5B	MOV E,E		Copy E to E
5C	MOV E,H	H.E	Copy H to E
5D	MOV E,L	L.E	Copy L to E
5E	MOV E,M	M.E	Load byte at (HL) into E
5F	MOV E,A	A.E	Copy A to E
60	MOV H,B	B.H	Copy B to H
61	MOV H,C	C.H	Copy C to H
62	MOV H,D	D.H	Copy D to H
63	MOV H,E	E.H	Copy E to H

Hex Opcode	MACASM Mnemonic	ASM80 Mnemonic	Operation
64	MOV H,H		Copy H to H
65	MOV H,L	L.H	Copy L to H
66	MOV H,M	M.H	Load byte at (HL) into H
67	MOV H,A	A.H	Copy A to H
68	MOV L,B	B.L	Copy B to L
69	MOV L,C	C.L	Copy C to L
6A	MOV L,D	D.L	Copy D to L
6B	MOV L,E	E.L	Copy E to L
6C	MOV L,H	H.L	Copy H to L
6D	MOV L,L		
6E	MOV L,M	M.L	Load byte at (HL) into L
6F	MOV L,A	A.L	Copy A to L
70	MOV M,B	B.M	Store B at (HL)
71	MOV M,C	C.M	Store C at (HL)
72	MOV M,D	D.M	Store D at (HL)
73	MOV M,E	E.M	Store E at (HL)
74	MOV M,H	H.M	Store H at (HL)
75	MOV M,L	L.M	Store L at (HL)
76	HLT	HLT	Halt
77	MOV M,A	A.M	Store A at (HL)
78	MOV A,B	B.A	Copy B to A
79	MOV A,C	C.A	Copy C to A
7A	MOV A,D	D.A	Copy D to A
7B	MOV A,E	E.A	Copy E to A
7C	MOV A,H	H.A	Copy H to A
7D	MOV A,L	L.A	Copy L to A
7E	MOV A,M	M.A	Load byte at (HL) into A
7F	MOV A,A		Copy A to A
80	ADD B	ADB	Add B to A
81	ADD C	ADC	Add C to A
82	ADD D	ADD	Add D to A
83	ADD E	ADE	Add E to A
84	ADD H	ADH	Add H to A
85	ADD L	ADL	Add L to A
86	ADD M	ADM	Add byte at (HL) to A
87	ADD A	ADA	Add A to A
88	ADC B	ACB	Add B to A with Carry
89	ADC C	ACC	Add C to A with Carry
8A	ADC D	ACD	Add D to A with Carry
8B	ADC E	ACE	Add E to A with Carry
8C	ADC H	ACH	Add H to A with Carry
8D	ACD L	ACL	Add L to A with Carry
8E	ADC M	ACM	Add byte at (HL) to A w/Carry
8F	ACD A	ACA	Add A to A with Carry
90	SUB B	SUB	Subtract B from A
91	SUB C	SUC	Subtract C from A
92	SUB D	SUD	Subtract D from A

Hex Opcode	MACASM Mnemonic	ASM80 Mnemonic	Operation
93	SUB E	SUE	Subtract E from A
94	SUB H	SUH	Subtract H from A
95	SUB L	SUL	Subtract L from A
96	SUB M	SUM	Subtract byte at (HL) from A
97	SUB A		Subtract A from A
98	SBB B	SBB	Subtract B from A with borrow
99	SBB C	SBC	Subtract C from A with borrow
9A	SBB D	SBD	Subtract D from A with borrow
9B	SBB E	SBE	Subtract E from A with borrow
9C	SBB H	SBH	Subtract H from A with borrow
9D	SBB L	SBL	Subtract L from A with borrow
9E	SBB M	SBM	Subtract M from A with borrow
9F	SBB A	SBA	Subtract A from A with borrow
A0	ANA B	NDB	AND B with A
A1	ANA C	NDC	AND C with A
A2	ANA D	NDD	AND D with A
A3	ANA E	NDE	AND E with A
A4	ANA H	NDH	AND H with A
A5	ANA L	NDL	AND L with A
A6	ANA M	NDM	AND Memory with A
A7	ANA A		AND A with A
A8	XRA B	XRB	Exclusive OR B with A
A9	XRA C	XRC	Exclusive OR C with A
AA	XRA D	XRD	Exclusive OR D with A
AB	XRA E	XRE	Exclusive OR E with A
AC	XRA H	XRH	Exclusive OR H with A
AD	XRA L	XRL	Exclusive OR L with A
AE	XRA M	XRM	Exclusive OR byte at (HL) with A
AF	XRA A	XRA	Exclusive OR A with A
B0	ORA B	ORB	OR B with A
B1	IRA C	ORC	OR C with A
B2	ORA D	ORD	OR D with A
B3	ORA E	ORE	OR E with A
B4	ORA H	ORH	OR H with A
B5	ORA L	ORL	OR L with A
B6	ORA M	ORM	OR Memory with A
B7	ORA A	TST	Test A (OR A with A)
B8	CMP B	CPB	Compare B with A
B9	CMP C	CPC	Compare C with A

Hex	MACASM	ASM80	
Opcode	Mnemonic	Mnemonic	Operation
BA	CMP D	CPD	Compare D with A
BB	CMP E	CPE	Compare E with A
BC	CMP H	CPH	Compare H with A
BD	CMP L	CPL	Compare L with A
BE	CMP M	CPM	Compare byte at (HL) with A
BF	CMP A	CPA	Compare A with A
C0	RNZ	RFZ	Return if Not Zero
C1	POP B	ST.BC	Pop Stack Into BC
C2	JNZ addr	JFZ	Jump if Not Zero
C3	JMP addr	JMP	Jump (uncondtional)
C4	CNZ addr	CFZ	Call if Not Zero
C5	PUSH B	BC.ST	Push BC Onto Stack
C6	ADI	ADI	Add Immediate byte to A
C7	RST 0	RST0	Restart 0
C8	RZ	RTZ	Return if Zero
C9	RET	RET	Return (unconditional)
CA	JZ addr	JTZ	Jump if Zero
CB			
CC	CZ addr	CTZ	Call if Zero
CD	CALL addr	CAL	Call (unconditional)
CE	ACI	ACI	Add Immediate byte to A w/Carry
CF	RST 1	RST1	Restart 1
D0	RNC	RFC	Return if No Carry
D1	POP D	ST.DE	Pop Stack Into DE
D2	JNC addr	JFC	Jump if No Carry
D3	OUT	OPT	Output Instruction
D4	CNC addr	CFC	Call if No Carry
D5	PUSH D	DE.ST	Push DE onto Stack
D6	SUI	SUI	Subtract Immediate byte from A
D7	RST 2	RST2	Restart 2
D8	RC	RTC	Return if Carry (set)
D9			
DA	JC addr	JTC	Jump if Carry (set)
DB	IN	IPT	Input Instruction
DC	CCc addr	CTC	Call if Carry (set)
DD			
DE	SBI	SBI	Subtract Imm. byte from A w/borrow
DF	RST 3	RST3	Restart 3
E0	RPO	RFP	Return if Parity Odd (reset)
E1	POP H	ST.HL	Pop Stack Into HL
E2	JPO addr	JFP	Jump if Parity Odd (reset)
E3	XTHL	HL\ST	Exchange HL with Stack
E4	CPO addr	CFP	Call if Parity Odd (reset)
E5	PUSH H	HL.ST	Push HL onto Stack
E6	ANI	NDI	AND Immediate byte with A
E7	RST 4	RST4	Restart 4

Hex Opcode	MACASM Mnemonic	ASM80 Mnemonic	Operation
E8	RPE	RTP	Return if Parity Even (set)
E9	PCHL	J@HL	Jump to (HL)
EA	JPE addr	JTP	Jump if Parity Even (set)
EB	XCHG	HL\DE	Exchange HL with DE
EC	CPE addr	CTP	Call if Parity Even (set)
ED			
EE	XRI	XRI	Exclusive OR Immed. byte with A
EF	RST 5	RST5	Restart 5
F0	RP	RFS	Return if Positive (sign reset)
F1	POP PSW	ST.A	Pop A and Flags from Stack
F2	JP addr	JFS	Jump if Positive (sign reset)
F3	DI addr	DIN	Disable Interrupts
F4	CP addr	CFS	Call if Positive (sign reset)
F5	PUSH PSW	A.ST	Push A with Flags onto Stack
F6	ORI	ORI	OR Immediate byte with A
F7	RST 6	RST6	Restart 6
F8	RM	RTS	Return if Minus (sign set)
F9	SPHL	HL.SP	Copy HL to SP
FA	JM addr	JTS	Jump if Minus (sign set)
FB	EI	EIN	Enable Interrupts
FC	CM	CTS	Call if Minus (sign set)
FD			
FE	CPI	CPI	Compare Immediate byte with A
FF	RST 7	REST7	Restart 7

APPENDIX R-3
INSTRUCTIONS IN MNEMONIC ORDER WITHIN GROUP

Hex Code	MACASM Mnemonic	ASM80 Mnemonic	Operation
		DATA TRANSFER GROUP	
0A	LDAX B	@BC.A	Load (BC) into A
1A	LDAX D	@DE.A	Load (DE) into A
3A	LDA addr	@I.A	Load byte at addr into A
2A	LHLD addr	@I.HL	Load bytes at addr into HL
01	LXI B	I.BC	Load Immed. bytes into BC
11	LXI D	I.DE	Load Immed. bytes into DE
21	LXI H	I.HL	Load Immed. bytes into HL
31	LXI SP	I.SP	Load Immed. bytes into SP
47	MOV B,A	A.B	Copy A to B
4F	MOV C,A	A.C	Copy A to C
57	MOV D,A	A.D	Copy A to D
5F	MOV E,A	A.E	Copy A to E
67	MOV H,A	A.H	Copy A to H
6F	MOV L,A	A.L	Copy A to L
77	MOV M,A	A.M	Store A at (HL)
78	MOV A,B	B.A	Copy B to A
48	MOV C,B	B.C	Copy B to C
50	MOV D,B	B.D	Copy B to D
58	MOV E,B	B.E	Copy B to E
60	MOV H,B	B.H	Copy B to H
68	MOV L,B	B.L	Copy B to L
70	MOV M,B	B.M	Store B at (HL)
79	MOV A,C	C.A	Copy C to A
41	MOV B,C	C.B	Copy C to B
51	MOV D,C	C.D	Copy C to D
59	MOV E,C	C.E	Copy C to E
61	MOV H,C	C.H	Copy C to H
69	MOV L,C	C.L	Copy C to L
71	MOV M,C	C.M	Store C at (HL)
7A	MOV A,D	D.A	Copy D to A
42	MOV B,D	D.B	Copy D to B
4A	MOV C,D	D.C	Copy D to C
5A	MOV E,D	D.E	Copy D to E
62	MOV H,D	D.H	Copy D to H
6A	MOV L,D	D.L	Copy D to L
72	MOV M,D	D.M	Store D at (HL)
7B	MOV A,E	E.A	Copy E to A
43	MOV B,E	E.B	Copy E to B
4B	MOV C,E	E.C	Copy E to C
53	MOV D,E	E.D	Copy E to D
63	MOV H,E	E.H	Copy E to H
6B	MOV L,E	E.L	Copy E to L
73	MOV M,E	E.M	Store E at (HL)

Hex Code	MACASM Mnemonic	ASM80 Mnemonic	Operation
DATA TRANSFER GROUP - Continued			
7C	MOV A,H	H.A	Copy H to A
44	MOV B,H	H.B	Copy H to B
4C	MOV C,H	H.C	Copy H to C
54	MOV D,H	H.D	Copy H to D
5C	MOV E,H	H.E	Copy H to E
6C	MOV L,H	H.L	Copy H to L
74	MOV M,H	H.M	Store H at (HL)
7D	MOV A,L	L.A	Copy L to A
45	MOV B,L	L.B	Copy L to B
4D	MOV C,L	L.C	Load L to C
55	MOV D,L	L.D	Copy L to D
5D	MOV E,L	L.E	Load L to E
65	MOV H,L	L.H	Load L to H
75	MOV M,L	L.M	Store L at (HL)
7E	MOV A,M	M.A	Load byte at (HL) into A
46	MOV B,M	M.B	Load byte at (HL) into B
4E	MOV C,M	M.C	Load byte at (HL) into C
56	MOV D,M	M.D	Load byte at (HL) into D
5E	MOV E,M	M.E	Load byte at (HL) into E
66	MOV H,M	M.H	Load byte at (HL) into H
6E	MOV L,M	M.L	Load byte at (HL) into L
3E	MVI A	I.A	Load Immed. byte into A
06	MVI B	I.B	Load Immed. byte into B
0E	MVI C	I.C	Load Immed. byte into C
16	MVI D	I.D	Load Immed. byte into D
1E	MVI E	I.E	Load Immed. byte into E
26	MVI H	I.H	Load Immed. byte into H
2E	MVI L	I.L	Load Immed. byte into L
36	MVI M	I.M	Store Immed. byte at (HL)
22	SHLD addr	HL.@I	Store HL Direct
F9	SPHL	HL.SP	Copy HL to SP
02	STAX B	A.@BC	Store A at (BC)
12	STAX D	A.@DE	Store A at (DE)
32	STA addr	A.@I	Store A at Immed. addr.
EB	XCHG	HL\DE	Exchange HL with DE

Hex Code	MACASM Mnemonic	ASM80 Mnemonic	Operation
		ARITHMETIC GROUP	
8F	ADC A	ACA	Add A to A with Carry
88	ADC B	ACB	Add B to A with Carry
89	ADC C	ACC	Add C to A with Carry
8A	ADC D	ACD	Add D to A with Carry
8B	ADC E	ACD	Add E to A with Carry
8C	ADC H	ACH	Add H to A with Carry
CE	ACI	ACI	Add Immed. byte to A w/Carry
8D	ADC L	ACL	Add L to A with Carry
8E	ADC M	ACM	Add Byte at (HL) to A w/Carry
87	ADC L	ACL	Add A to A
80	ADD B	ADB	Add B to A
81	ADD C	ADC	Add C to A
82	ADD D	ADD	Add D to A
83	ADD E	ADE	Add E to A
84	ADD H	ADH	Add H to A
C6	ADI	ADI	Add Immediate byte to A
85	ADD L	ADL	Add L to A
86	ADD M	ADM	Add Byte at (HL) to A
09	DAD B	ADBC	Add BC to HL
19	DAD D	ADDE	Add DE to HL
29	DAD H	ADHL	Add HL to HL
39	DAD SP	ADSP	Add SP to HL
27	DAA	DAA	Decimal Adjust A
3D	DCR A	DCA	Decrement A
05	DCR B	DCB	Decrement B
0D	DCR C	DCC	Decrement C
15	DCR D	DCD	Decrement D
1D	DCR E	DCE	Decrement E
25	DCR H	DCH	Decrement H
2D	DCR L	DCL	Decrement L
35	DCR M	DCM	Decrement Byte at (HL)
0B	DCX B	DCBC	Decrement BC
1B	DCX D	DCDE	Decrement DE
2B	DCX H	DCHL	Decrement HL
3B	DCX SP	DCSP	Decrement SP
3C	INR A	INA	Increment A
04	INR B	INB	Increment B
0C	INR C	INC	Increment C
14	INR D	IND	Increment D
1C	INR E	INE	Increment E
24	INR H	INH	Increment H
2C	INR L	INL	Increment L
34	INR M	INM	Increment Byte at (HL)
03	INX B	INBC	Increment BC
13	INX D	INDE	Increment DE
23	INX H	INHL	Increment HL
33	INX SP	INSP	Increment SP

Hex Code	MACASM Mnemonic	ASM80 Mnemonic	Operation

ARITHMETIC GROUP - Continued

Hex Code	MACASM Mnemonic	ASM80 Mnemonic	Operation
9F	SBB B	SBA	Subtract A from A w/Borrow
98	SBB B	SBB	Subtract B from A w/Borrow
99	SBB C	SBC	Subtract C from A w/Borrow
9A	SBB D	SBD	Subtract D from A w/Borrow
9B	SBB E	SBE	Subtract E from A w/Borrow
9C	SBB H	SBH	Subtract H from A w/Borrow
DE	SBI	SBI	Subtract Immediate byte from A w/Borrow
9D	SBB L	SBL	Subtract L from A w/Borrow
9E	SBB M	SBM	Subtract byte at (HL) from A w/Borrow
37	STC	STC	Set Carry
9F	SUB A		Subtract A from A
90	SUB B	SUB	Subtract B from A
91	SUB C	SUC	Subtract C from A
92	SUB D	SUD	Subtract D from A
93	SUB E	SUE	Subtract E from A
94	SUB H	SUH	Subtract H from A
D6	SUI	SUI	Subtract Immed. byte from A
95	SUB L	SUL	Subtract L from A
96	SUB M	SUM	Subtract byte at (HL) from A

STACK & MACHINE CONTROL GROUP

Hex Code	MACASM Mnemonic	ASM80 Mnemonic	Operation
F3	DI	DIN	Disable Interrupts
FB	EI	EIN	Enable Interrupts
76	HLT	HLT	Halt
F1	POP PSW	ST.A	Pop Accumulator and Flags
C1	POP B	ST.BC	Pop Stack into BC
D1	POP D	ST.DE	Pop Stack into DE
E1	POP H	ST.HL	Pop Stack into HL
F5	PUSH PSW	A.ST	Push A and Flags onto Stack
C5	PUSH B	BC.ST	Push BC onto Stack
D5	PUSH D	DE.ST	Push DE onto Stack
E5	PUSH H	HL.ST	Push HL onto Stack
E3	XTHL	HL\ST	Exchange HL with Stack

INPUT/OUTPUT GROUP

DB	IN	IPT	Input Instruction
D3	OUT	OPT	Output Instruction

APPENDIX R-4

OP-1 INSTRUCTION TIMING

MNEMONIC	uCYCLES	uSECS	MNEMONIC	uCYCLES	uSECS
ADC m	9	2.88	MVI r ,data	9	2.88
ADC r	5	1.60	NOP	5	1.6
ADD m	9	2.88	ORA M	9	2.88
ADD r	5	1.6	ORA r	5	1.6
ACI data	9	2.88	ORI data	9	2.88
ADI data	9	2.88	OUT	13	4.16
ANA m	9	2.88			
ANA r	5	1.6	PCHL	7	2.24
ANI data	9	2.88	POP PSW	13	4.16
CALL addr	23	7.36	POP rp	13	4.16
Ccond addr	11/23	3.52/7.36	PUSH PSW	15	4.8
CMA	5	1.6	PUSH rp	15	4.8
CMC	5	1.6	RAL	5	1.6
			RAR	5	1.6
CMP M	9	2.88	RLC	5	1.6
			Rcond	7/15	2.24/4.8
CMP r	5	1.6	RRC	5	1.6
CPI data	9	2.88	RET	13	4.16
DAA	5	1.6	RST n	15	4.8
DAD rp	11	3.52	SBB M	9	2.88
DCR1M	13	4.16	SBB r	5	1.6
DCR r	5	1.6	SBI data	5	2.88
DCX rp	7	2.24	SHLD addr	21	6.72
			SPHL	7	2.24
DI	5	1.6	STA addr	17	5.44
EI	5	1.6	STAX rp	9	2.88
HLT	6	1.92	STC	5	1.6
IN	13	4.16	SUB M	9	2.88
INR M	13	4.16	SUB r	5	1.6
INR r	5	1.6	SUI data	9	2.88
INX rp	7	2.24	XCHG	5	1.6
Jcond addr	9/13	2.88/4.16	XRA M	9	2.88
JMP addr	13	4.16	XRA r	5	1.6
LDA addr	17	5.44	XRI data	9	2.88
LDAX rp	9	2.88	XTHL	21	6.72
LHLD addr	21	6.72			
LXI rp,data	13	4.16			
MOV M,r	9	2.88			
MOV rd,rs	5	1.6			
MOV r ,M	9	2.88			
MVI M,data	13	4.16			

NOTE: Cycle time = 32µs.
µcycles include wait states for
memory references

Ontel Corporation

250 Crossways Park Drive
Woodbury, New York 11797
(516) 364-2121