

MCEM-8080
MICROCOMPUTER
SYSTEM

HAL MCEM-8080 MICROCOMPUTER SYSTEM

TECHNICAL MANUAL

WARRANTY

The HAL Communications Corp. MCEM-8080 Microcomputer System is fully guaranteed against defects in materials and workmanship for a period of one year. Should repair or replacement parts be required, notify HAL Communications Corp. promptly. Please do not return your unit to the factory for repair or adjustment until you have received a written return authorization.

HAL Communications assumes no responsibility for the repair or replacement of parts or units which have been damaged, abused, improperly installed, or modified and reserves the right to change the design of this equipment without incurring obligation to incorporate such changes into existing units. Operation of this equipment with improper power supply voltages (as described in this manual) will invalidate the warranty.

Copyright © 1976 by HAL Communications Corp., Urbana, Illinois. Printed in the United States of America. All rights reserved. Contents of this publication may not be reproduced in any form without the written permission of the copyright owner.

April, 1976 Printing

MCEM-8080 MICROCOMPUTER SYSTEM

CONTENTS:

1.	SYSTEM COMPONENTS.	1-1
1.1	8080A Microprocessor.	1-1
1.2	Processor Control Circuitry	1-1
1.3	Random Access Memory.	1-1
1.4	Read Only Memory.	1-2
1.5	Serial Input/Output (I/O)	1-2
1.6	Parallel Data Input/Output.	1-3
1.7	Bus Indicators and Control.	1-3
1.8	Connectors used on the MCEM-8080.	1-5
2.	INSTALLATION OF THE MCEM-8080.	2-1
2.1	Initial Installation.	2-1
2.2	Connection of Serial Input/Output Devices	2-1
2.3	Connection of Power Supplies to the MCEM.	2-9
2.4	Connection of Parallel I/O Devices.	2-9
2.5	I/O Connector Pin Assignments	2-10
2.6	Universal Processor Bus Connector	2-12
3.	OPERATION OF THE MCEM-8080	3-1
3.1	Software Monitor.	3-1
3.2	Monitor Subroutines	3-5
4.	SYSTEM ADDRESS ASSIGNMENTS	4-1
4.1	Random Access Memory (RAM).	4-1
4.2	Read Only Memory (ROM).	4-1
4.3	Input/Output (I/O) Assignments.	4-2
5.	OPERATING HINTS.	5-1
5.1	Power-on Start Up	5-1
5.2	Changing the Monitor Mode	5-2
5.3	Manually Writing a Memory Location.	5-3
5.4	Manually Jumping to a Program Address	5-3
5.5	Manually Writing to an Output Port.	5-4
5.6	Using the Break-point Register for Debug.	5-4
5.7	Using the E and R Commands for Debugging.	5-4
6.	PROGRAM EXAMPLES	6-1
7.	SOFTWARE MONITOR LISTINGS.	7-1
8.	CIRCUIT BOARD LAYOUT AND SCHEMATIC DIAGRAMS.	8-1
9.	APPENDICES	9-1
A.	PROM, EPROM, ROM Connections and Markings	9-1
B.	Input/Output Jumper Connections	9-6
C.	Addition of INTREQ, <u>RESTIN</u> , and <u>BUSEN</u> to the UPB	9-10
D.	8080A Assembly Language Coding Forms.	9-12

TABLES:

Table 2.1	ASCII Character Code	2-2
Table 2.2	Baudot Character Code.	2-3
Table 2.3	Oscillator Frequency and Period.	2-5
Table 2.4	Power Requirements and Connections	2-9
Table 2.5	I/O Connections.	2-10
Table 2.6	Universal Processor Bus Connections.	2-12

ILLUSTRATIONS:

Figure 1.1	MCEM-8080 Microcomputer System	iv
Figure 2.1	EIA - RS-232C Serial I/O (standard jumpers).	2-6
Figure 2.2	Dual current loop serial I/O	2-7
Figure 2.3	Single current loop serial I/O	2-8
Figure 2.4	Bit assignment and timing for parallel output.	2-11
Figure 2.5	Bit assignment and timing for parallel input	2-11
Figure 8.1	MCEM-8080 circuit board layout	8-2
Figure 8.2	Schematic drawing conventions.	8-3
Figure 8.3	MCEM-8080 CPU Interface / Control.	8-4
Figure 8.4	MCEM-8080 Processor Memory	8-5
Figure 8.5	MCEM-8080 Input/Output Interface	8-6
Figure A.1	Memory jumpers for 3624-4 PROMs.	9-2
Figure A.2	Memory jumpers for 8704 EPROM.	9-3
Figure A.3	Memory jumpers for 8708 or 8308 ROMs	9-4
Figure A.4	PROM color coding.	9-5
Figure B.1	EIA - RS-232C Serial I/O Jumper Locations.	9-7
Figure B.2	Dual current loop serial I/O Jumper Locations.	9-8
Figure B.3	Single current loop serial I/O Jumper Locations.	9-9
Figure C.1	Addition of <u>RESIN</u> , <u>BUSEN</u> , and <u>INTREQ</u> to UPB.	9-11

INTRODUCTION

The HAL MCEM-8080 Microcomputer System is a single printed-circuit board computer that can be used for program development or for specific control applications. The MCEM-8080 is designed around the Intel 8080A single chip, 8-bit, N-channel microprocessor integrated circuit. The MCEM-8080 printed circuit board contains the microprocessor IC, its timing and control circuitry, both Read Only Memory (ROM) and Random Access Memory (RAM) integrated circuitry, and timing and control for Input / Output (I/O) interfacing. Other accessories such as additional RAM, Keyboard/Video Display unit, tape cassette memory, and power supplies can be used with the basic MCEM circuit board. This manual discusses ONLY the basic MCEM-8080 board - the operation of the accessories is discussed in separate manuals furnished with each unit.

The MCEM-8080A Microcomputer System manual is actually supplied in two publications: this MCEM-8080 Operating Manual, and the Intel 8080 Microcomputer Systems User's Manual. Specific operating instructions and specifications pertaining to the HAL Communications Corp. MCEM-8080 system are discussed in the MCEM-8080 Operating Manual. General information relating to the 8080A and associated integrated circuits is discussed in detail in the Intel 8080 Microcomputer Systems User's Manual (© Intel Corporation). When pertinent, references are made in the operating manual to the detailed discussions in the Intel manual. These references are given in the form: "Intel; pp A-xx to A-yy". In addition, a copy of the Intel 8080 Assembly Language Reference Card (© Intel Corporation) is furnished with the MCEM-8080 Microcomputer System to aid in program development.

Figure 1.1 MCEM-8080 Microcomputer System

1. SYSTEM COMPONENTS

The HAL MCEM-8080 Microcomputer System contains the following basic and optional components:

1.1 8080A Microprocessor

The 8080A is an eight bit microprocessor integrated circuit with an instruction repertoire of 73 instructions. The execution time of these instructions varies from 2.0 μ sec. to 9.0 μ sec. The 8080A integrated circuit itself contains all of the circuitry required to address the memory, address Input / Output (I/O) devices, and manipulate data. A more detailed discussion of the 8080A will be found in pages 2-1 to 2-20 of the Intel manual (Intel; pp 2-1 to 2-20).

1.2 Processor Control Circuitry

Two additional integrated circuits are used in conjunction with the 8080A to provide all of the timing and control signals for the microprocessor system. These are the 8228 Bus Controller IC and the 8224 Clock Generator IC.

1.2.1 8228 Bus Controller (Intel; pp 5-7 to 5-12)

A type 8228 integrated circuit is used to decode signals from the 8080A and generate the required bus control signals. This device also buffers the 8080A data bus signals and will support a single vector interrupt (RST 7).

1.2.2 8224 Clock Generator (Intel; pp 5-1 to 5-6)

A type 8224 integrated circuit generates all system timing signals. An 18 MHz crystal is used with the device to generate the 2.0 MHz processor timing signals, power-on reset signal and ready line synchronization pulses.

1.3 Random Access Memory

The standard MCEM circuit board is provided with 1024 bytes of Random Access Memory (RAM). This memory can be used by the user's programs, but the lower 64 bytes are required for the software monitor program. Additional circuit board space is provided so that an additional 1024 bytes ("1K") of RAM can be installed on the MCEM board (factory installation is recommended). All RAM integrated circuits should be type 8102A-4, a device featuring an access time of 450 nsec. or less. Slower RAM devices should NOT be used as they may cause improper operation of the system. Further information on the 8102A-4 is found on pages 5-79 through 5-82 of the Intel manual (Intel; pp 5-79 to 5-82).

Within the processor memory space, the standard "1K" bytes of RAM occupy locations between 0 and 1023 (0 - 3FF - Hex). The second (optional) "1K" bytes of RAM occupy locations between 1024 and 2047 (400 H - 7FF H). The software monitor uses RAM locations between 0 and 63 (0 - 3F H).

1.4 Read Only Memory

The MCEM system is provided with sufficient circuit board space for 4096 bytes of EPROM (Erasable Programmable Read Only Memory) or 2048 bytes of bi-polar PROM (Programmable Read Only Memory - NOT erasable). The device selection is made by selection of the proper circuit board jumpers. Four socket locations are provided for the ROM - all four must be of the same type (EPROM or PROM). The ROM occupies consecutive memory locations, starting at 32,768 (8000 H).

1.4.1 EPROM

Either a type 8708 or 8704 EPROM integrated circuit (Intel; pp 5-45 to 5-50) can be used on the MCEM board. The 8708 is a 1024 x 8 device and the 8704 is a 512 x 8 device. Refer to Appendix A for proper jumper placement.

1.4.2 PROM

Type 3624 PROM integrated circuits can be used on the MCEM. This IC is the standard device furnished with the MCEM. The 3624 is a bi-polar PROM with a 512 x 8 organization. Up to four 3624's can be used on the MCEM-8080 circuit board. NOTE: Production MCEM-8080 circuit boards are jumpered for use of this device on the circuit board. If it is desired to use other devices, refer to Appendix A for details.

1.4.3 ROM

A type 8308 ROM integrated circuit (Intel; pp 5-59 & 5-60) can also be used in the HAL MCEM-8080. This is a mask-programmed version of the 8708. Refer to Appendix A for jumper details.

1.4.4 Monitor Software ROM

The HAL software monitor can be resident in either 2-3624, 1-8708, or 1-8308 ROM integrated circuits. Either 2-3624 or 1-8308 ROM is standard with the MCEM. The monitor software is 1024 bytes in length and begins at location 32,768 (8000 H).

1.5 Serial Input / Output (I/O)

The standard MCEM-8080 provides for either synchronous or asynchronous serial data interface. The software monitor supports asynchronous serial I/O in either Baudot (5-unit) or ASCII (8-unit) codes.

1.5.1 8251 USART

A type 8251 integrated circuit (Intel; pp 5-135 to 5-146) Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is used to input and output serial data. This device is fully programmable and is controlled by the processor. Parallel-to-serial and serial-to-parallel conversions as well as word length selection and parity are controlled by the 8251.

1.5.2 Serial Timing Oscillator

A type 555 integrated circuit timer is used to generate the serial data baud rate. The data rate is screw-driver adjustable on the circuit board. The actual 555 clock frequency is 4 times the baud rate in ASCII mode and 16 times the baud rate in Baudot mode.

1.5.3 EIA - RS-232C Data Interface

Two operational amplifiers (both halves of a type 1458 IC) are used as RS-232 drivers and receivers. The serial output of the 8251 USART is directly converted to a ± 5 volt signal, with -5 volts representing the "mark" signal condition and +5 volts as "space". The output impedance of the circuit is approximately 400 ohms. For input data, an operational amplifier is used as a sense amplifier and level converter. Input voltages greater than +1.0 volts are interpreted to be in the "space" condition and those less than +1.0 volts as "mark". The input impedance is approximately 2700 ohms. This input will properly sense TTL-level signals, as well as EIA - RS-232C signals.

1.5.4 Current Loop Interface

Current loop signals with either 20 or 60 ma mark currents can also be connected to the MCEM-8080. Two optical isolator integrated circuits are used to convert between the floating current loop circuit and the RS-232 levels. These sensors are separated so that one can be used for data input and the other for output (separate current loops - "full-duplex" operation). The two circuits can also be series connected to provide both data input and output on a single current loop circuit ("half-duplex" operation).

1.6 Parallel Data Input / Output

A type 8255 integrated circuit (Intel; pp 5-113 to 5-133) is provided to allow parallel data interfacing. This device, called the "Programmable Peripheral Interface", consists of three buffered 8-bit parallel data ports. The software monitor utilizes the 8255 for parallel I/O operations.

1.7 Bus Indicators and Control

A number of indicators (small LEDs - Light Emitting Diodes) and switches are installed along the front edge of the MCEM-8080 circuit board to permit evaluation and control of the processor operation.

1.7.1 Address Indicators

The entire 16 bits of the 8080 address bus are displayed on 16 LEDs. The lamps are grouped in four-lamp clusters, four clusters total. Each group of four lamps represents a single hexadecimal (HEX) character, 0 through F. An illuminated lamp indicates a logic "1" condition. Within a four-lamp cluster, the least significant bit (LSB) is represented by the right-hand lamp. Similarly, the right-hand cluster of four lamps represents the least significant hexadecimal character.

1.7.2 Data Indication

Eight lamps (in two four-lamp clusters) are used to indicate the state of the processor data bus. These lamps are immediately to the left of the address lamps. As before, the right-hand lamp represents the LSB and an illuminated lamp represents a logical "1" for that bit.

1.7.3 Bus Control Indication

The four lamps on the extreme left end of the circuit board indicate the state of the I/O Read, I/O Write, Memory Read, and Memory Write (left-to-right order) signals from the processor. An illuminated lamp indicates which of these operations is active. A complete description of the function of these signals is found in the Intel manual (Intel; pp 5-7 to 5-12).

1.7.4 Manual Data Switches

Immediately in front of the eight data lamps are located two, four-section miniature switches. The switches provide manual control of the contents of the data bus. These switches can be used to enter data only when the Data Bus Override (DBO) switch (to the right of the data switches) is in the ON position. The data switch settings at any other time does not affect the processor. The switches are arranged in the same manner as the lamps, LSB to the right.

1.7.5 Run / Stop Switch

A miniature toggle switch on the right-hand section of the board (labeled RUN - STOP) allows manual control of the 8080A Ready line. When this switch is set to the RUN position, the processor will continue to operate (unless halted by the program or some other control). When in the STOP position, the processor is halted and only the manual STEP and RESET switches will cause processor activity.

1.7.6 Reset Switch

The far right-hand push-button switch (labeled RESET) is a momentary contact type that can be used to manually reset the 8080A. A reset operation causes the program counter to set to zero and the interrupt flip-flop to be cleared. Processor execution commences at location 0000 when the reset switch is released. Application of DC power supplies automatically issues a reset function.

1.7.7 Single Step Switch

The STEP switch (located between the RUN - STOP and RESET switches) allows manual stepping of the computer, one MEMORY cycle at a time. This switch only functions when the processor has been halted by either the RUN - STOP switch or the break point register. It is important to remember that some instructions require more than one memory cycle and therefore more than one operation of step switch to complete.

1.7.8 Break Point Register Switches

In the middle of the control area of the circuit board are located four, four-section miniature switches. These 16 switches form a "break point register". Circuitry is provided to compare the value of this switch register with the address bus and cause the 8080A to stop operation if the two are equal. This function is similar to a programmable stop. Once the 8080A is halted due to a break point "match", it can only be caused to continue running by either manual stepping with the STEP switch or by re-setting the break point switches to a new value.

1.7.9 Memory Write and Output Write Switches

Two momentary switches are located on the far left-hand side of the circuit board. These switches allow manual operation of memory or output functions. The MEMORY WRITE switch will cause a manual memory write function when depressed, overriding the normal bus control from the 8228 integrated circuit. Similarly, depression of the OUTPUT WRITE switch will cause an output write function, again overriding the normal control from the 8228.

1.8 Connectors used on the MCEM-8080

There are three connectors used on the basic MCEM-8080 circuit board. These connectors are used for I/O Interface, Power Input, and connection to the Universal Processor Bus. Mating connectors for each are furnished with the MCEM.

1.8.1 I/O Interface Connector

Input / Output (I/O) connections to the MCEM are made through a 36 pin circuit board edge connector (0.156" finger spacing, 18 pin double readout) located on the left edge of the board. All three parallel I/O ports of the 8255 are available on this connector as well as connections for serial data. The form of serial data to be used is selected with circuit board jumpers.

1.8.2 Power Connector

Power connections to the MCEM are made through the 12 pin edge connector (0.156" finger spacing 6 pin double readout) located in the upper right-hand corner of the circuit board. The MCEM requires ± 12 volt and +5 volt power supplies.

1.8.3 Universal Processor Bus Connector

Direct connection to the computer address, data, and control lines can be made through the 40 pin Universal Processor Bus (UPB) connector located in the lower right-hand corner of the board. A mating connector and attached ribbon cable are supplied for use of this feature. Connection of options such as additional memory and the Keyboard/Video Display unit is made through the UPB connector.

2. INSTALLATION OF THE MCEM-8080

2.1 Initial Inspection

Upon receipt of the MCEM-8080, unpack the circuit board and accessories and inspect them for evidence of shipping damage. If evidence of shipping damage is found, contact the carrier immediately. Before discarding the packing material, check that all parts and accessories are accounted for. If any are missing, please notify the factory or distributor in writing. The following parts and accessories are furnished with the MCEM-8080:

Accessories and Parts:

- 1 - 40 pin Universal Processor Bus (UPB) connector with 2 ft. of ribbon cable attached.
- 1 - 36 pin edge connector
- 1 - 12 pin edge connector
- 1 - MCEM Operating Manual
- 1 - Intel 8080 Microcomputer System User's Manual
- 1 - Intel 8080 Assembly Language Reference Card

2.2 Connection of Serial Input / Output Devices

The MCEM-8080 standard circuitry and software will support serial I/O (Input/Output) operations in either the 7-unit ASCII code OR the 5-unit Baudot code at a variety of baud rates. The code to be used is selected with circuit board jumpers. The MCEM-8080 is usually factory connected for the ASCII code.

2.2.1 ASCII Serial I/O Operation

The ASCII mode is selected by strapping pin 22 (DSR) of the 8251 (circuit number 15, left edge of board) to ground (see Appendix B). In ASCII mode, all serial communications is performed with a 7-bit ASCII format. This format is:

- 1 - start bit (space)
- 7 - data bits
- 1 - parity bit (set to space)
- 2 - stop bits (mark)
- 11 - bits per character

The serial baud rate timing is screw driver adjustable from 100 to 600 baud. The unit is factory adjusted for 300 baud (30 characters per second). As noted in section 1.5.2, the 555 timer is set to 16 times the output baud rate (eg., $16 \times 300 = 4800$ Hz for 300 baud). Table 2.1 contains a list of the ASCII character set used and their corresponding hexadecimal values. Common ASCII baud rates and the corresponding oscillator frequencies and periods are listed in Table 2.3.

Table 2.1 ASCII Character Code

		3 Most Significant Bits							
		0	1	2	3	4	5	6	7
4 Least Significant Bits	0	NUL	DLE	SPACE	0	@	P	'	p
	1	SOH	DC1	!	1	A	Q	a	q
	2	STX	DC2	"	2	B	R	b	r
	3	ETX	DC3	#	3	C	S	c	s
	4	EOT	DC4	\$	4	D	T	d	t
	5	ENQ	NAK	%	5	E	U	e	u
	6	ACK	SYN	&	6	F	V	f	v
	7	BEL	ETB	'	7	G	W	g	w
	8	BS	CAN	(8	H	X	h	x
	9	HT	EM)	9	I	Y	i	y
	A	LF	SUB	*	:	J	Z	j	z
	B	VT	ESC	+	;	K	[k	{
	C	FF	FS	,	<	L	\	l	!
	D	CR	GS	-	=	M]	m	}
	E	SO	RS	.	>	N	^	n	~
	F	SI	US	/	?	0	_	o	RUB OUT

- | | |
|---------------------------------|----------------------------|
| ACK = acknowledge | FS = file separator |
| BEL = bell | GS = group separator |
| BS = backspace | HT = horizontal tabulation |
| CAN = cancel | LF = line feed |
| CR = carriage return | NAK = negative acknowledge |
| DC1 = device control 1 | NUL = null |
| DC2 = device control 2 | RS = record separator |
| DC3 = device control 3 | RUB = delete (= DEL) |
| DC4 = device control 4 | SI = shift in |
| DLE = data link escape | SO = shift out |
| EM = end of medium | SOH = start of heading |
| ENQ = WRU = enquiry | STX = start of text |
| EOT = end of transmission | SUB = substitute |
| ESC = escape | SYN = synchronous idle |
| ETB = end of transmission block | US = unit separator |
| ETX = end of text | VT = vertical tabulation |
| FF = form feed | |

Mark = logical 1
Data is transmitted LSB first.

Table 2.2 Baudot Character Code

		Most Significant Bit (1)			
		Letters		Figures	
		0	1	0	1
4 Least Significant Bits	0	BLANK	T	BLANK	5
	1	E	Z	3	+
	2	LF	L	LF)
	3	A	W	-	2
	4	SPACE	H	SPACE	#
	5	S	Y	'	6
	6	I	P	8	0
	7	U	Q	7	1
	8	CR	O	CR	9
	9	D	B	\$?
	A	R	G	4	&
	B	J	FIG	BEL	FIG
	C	N	M	,	.
	D	F	X	!	/
	E	C	V	:	=
	F	K	LTR	(LTR

BEL = bell (or *)
 BLANK = blank (non print or space)
 CR = carriage return
 FIG = figures case
 LTR = letters case
 LF = line feed

Mark = logical 1
 Data is transmitted, LSB first.

circuit board. THE MCEM-8080 CAN BE DAMAGED IF THE I/O CONNECTOR IS REVERSED (particularly if connected to high-voltage current loop circuits).

2.6 Universal Processor Bus Connector *See Addendum 2.*

The processor bus of the 8080A can be extended with a 40 conductor ribbon cable attached to the Universal Processor Bus (UPB) connector. The total length of this cable should not exceed 24 inches. The total external loads should not exceed three standard TTL loads on the address and control lines and 5, LOW CURRENT, bus receiver loads on the data lines. The connections to the UPB connector are shown in Table 2.6.

Table 2.6 Universal Processor Bus Connections

<u>Pin</u>	<u>Function</u>	<u>Pin</u>	<u>Function</u>	<u>Pin</u>	<u>Function</u>
1	A12	15	A1	29	A14
2	+12	16	$\overline{\text{MEMR}}$	30	(NC)
3	A10	17	A3	31	A15
4	+5	18	$\overline{\text{I/O R}}$	32	Locating Key
5	A8	19	A5	33	DB0
6	Ground	20	$\overline{\text{I/O W}}$	34	DB4
7	A6	21	A7	35	DB1
8	Ground	22	RESET	36	DB5
9	A0	23	A9	37	DB2
10	02 (TTL)	24	RDY	38	DB6
11	A2	25	A11	39	DB3
12	(NC)	26	(NC)	40	DB7
13	A4	27	A13		
14	$\overline{\text{MEMW}}$	28	(NC)		

Note: Connections with the (NC) designator may have a function assigned but not connected on the factory standard units.

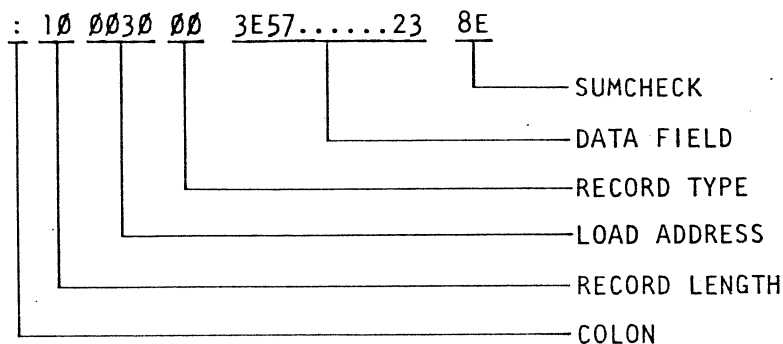
3. OPERATION OF THE MCEM-8080

3.1 Software Monitor

The software monitor supplied with the MCEM-8080 properly interfaces the serial I/O port, the parallel I/O ports, the keyboard display option, or other user-defined I/O devices. The monitor allows the user to perform the following operations. These commands are entered from the console.

3.1.1 Load hex (hexidecimal) files.

Large files can be loaded into the MCEM-8080 RAM from the reader device by using the following format:



COLON: All records must start with a COLON character. Any characters preceding the COLON are ignored.

RECORD LENGTH: The number of load bytes in the data field is specified as a number between 00 and FF (0 to 255). This is a hexadecimal number and is either two characters long or a single character followed by a comma (i.e., 07 = 7,). If a zero length record is entered, the load is terminated and control is restored to the monitor.

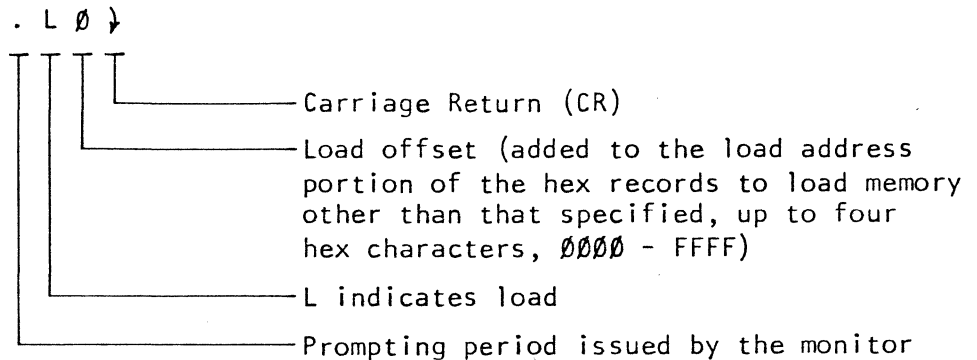
LOAD ADDRESS: The memory location into which the first byte of the data field will be written is specified here. Successive bytes in the data field will be written into successively higher memory locations. This number is either four characters long or less if terminated with a comma (i.e., 032E = 32E,).

RECORD TYPE: The record type is specified here. With the present version of the monitor (Version 1.1), all records are of type zero (enter 00).

DATA FIELD: The actual data to be written into memory is specified here. These are two character hex bytes and each pair of characters is converted to eight bits to be loaded into memory.

SUMCHECK: This hex byte represents the negative sum of all bytes (the load address is two bytes) in the record. The SUMCHECK value is such that, when modulo 256 is added to all of the other bytes of the record, the total will equal zero. This is a validity check on the record. If the SUMCHECK fails, an "X" will be printed on the serial output device. However, the data will still be loaded if the SUMCHECK fails.

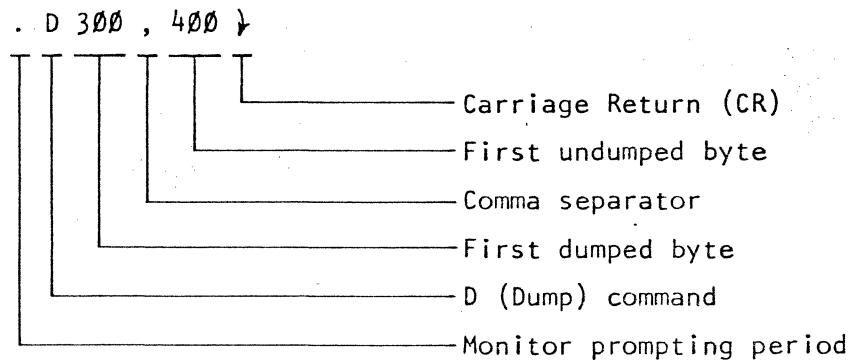
The format used to specify a load file is:



After receiving this command, the monitor will begin searching for the first colon.

3.1.2 Dump or Display

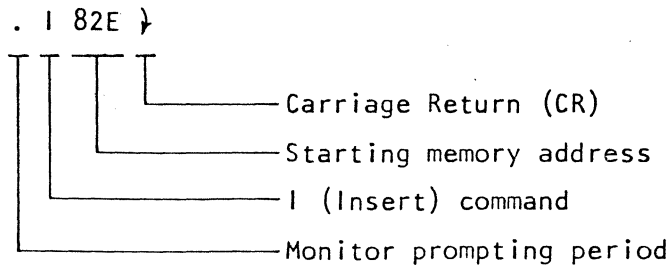
The contents of memory can be dumped (or displayed) by specifying the range to be dumped. The output generated is compatible with the load command so that memory areas can first be dumped and then loaded. The format of the dump is in a number of hex records (of maximum length = 10 H) until the entire range is depleted. For clarity, spaces are inserted between the various bytes but the monitor ignores spaces on input so that the dumped file is compatible with the load file routine. The dumped file is sent to the punch device. The command format is:



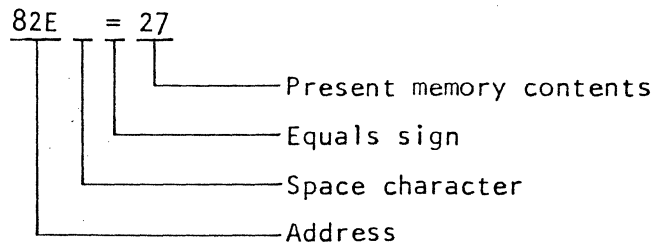
This example command will cause display of all memory contents between locations 300 H to 400 H - 1 as 16, 16 byte records. A zero length record is always added at the end.

3.1.3 Insert Memory Data

Individual locations in memory can be modified by using this command. The command format is:

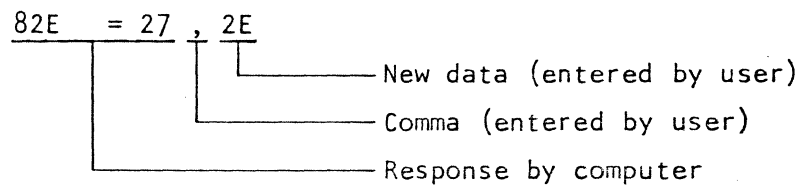


The output generated is of the following format:



After this has been output, a comma is typed followed by a new byte and when done, written into memory. If it is desired to leave the memory location unchanged, any non-comma character can be typed. After the new data has been entered, the address is incremented and displayed again. For example, consider:

. I 82E (Insert memory command, generated by user)

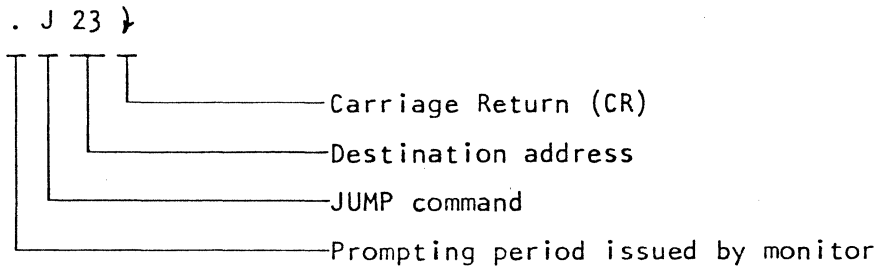


82F = 87 (Computer response indicating contents of next location)

If any character between "G" and "Z" is typed instead of a hex character, control returns to the monitor.

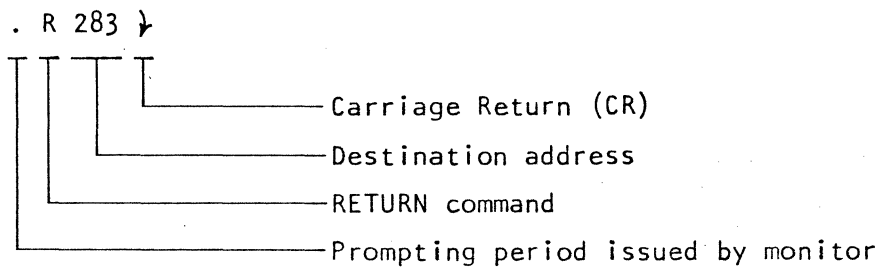
3.1.4 JUMP Command

Program control can be transferred to a specific location through the JUMP command. This command can be used to "jump" to a user program or subroutine. The format for this command:



3.1.5 RETURN command

Program control can be transferred to a specific location and the CPU registers restored to a predetermined value by executing a RETURN command. The format of this command is:



Twelve register values are restored by this command including:

<u>Register</u>	<u>Stored at Memory Location</u>
B	37
C	36
D	35
E	34
A (Accumulator)	33
PSW (Processor Status Word)	32
H	31
L	30
PC (PCH (high order program counter)	2F
(PCL (low order program counter)	2E
SP (SPH (high order stack pointer)	2D
(SPL (low order stack pointer)	2C

The initial value (to be restored) of these registers can be set by using an I (Insert) command to the memory location used for storage. These locations are shown in the above list. Note that, during the process of restoring the registers, the stack area indicated by SP (SPH & SPL) is used as temporary storage and therefore SP should contain a valid RAM address. If the destination address specified in the command is zero, the destination is taken from the storage area.

3.1.6 STOP command

A STOP command can be initiated at any time at which the monitor is expecting a control character by typing an "S" (or any other letter between

"G" and "Z"). As explained in section 3.1.4, this will cause the command to be aborted and control is returned to the monitor. The monitor will then issue a new prompting period.

3.1.7 EXIT command

An exit from a program to the monitor can be executed by entering a RST 7 instruction or a CALL 38 H. The monitor, upon turn-on, establishes an entry at 38 H from which it saves ALL CPU registers and status. This command is intended to permit the examination of all CPU registers and status while in the process of executing a program. The RST 7 instruction saves the PC (Program Counter) on the stack and jumps to location 38 H. From here, it jumps to a routine within the monitor which copies all registers into a special RAM area. When finished, the address of the initial RST 7 instruction is typed out as:

EXIT 232E (hexadecimal)

A prompting period is then issued by the monitor. At this point, the I (Insert) command can be used to examine and/or change individual registers. The memory location used to store the register values is listed under the RETURN command.

The most valuable use of the exit command is accomplished by inserting a RST 7 (0FF H) instruction in the program sequence being de-bugged and an automatic exit will be executed. The RETURN command can be used to return to the program sequence. An interrupt will also cause the exit command to be executed since a RST 7 is used as the interrupt vector.

3.2 Monitor Subroutines

Several general purpose subroutines are included in the software monitor. Some of these subroutines are:

3.2.1 BEGIN (address 8000 H)

This subroutine allows general entrance to the monitor mode. It initializes all parameters and the USART.

3.2.2 CI (Console Input - address 8003 H)

CI is a console input routine that will return an ASCII character (standard serial I/O) from the console control device and place the ASCII code in the A register. The contents of the A and PSW registers are modified. Three levels of the stack are used by this operation.

3.2.3 RI (address 8006 H)

This routine is the same as the CI routine except that the character is originated by the reader input device instead of by the console. Serial ASCII I/O is standard.

3.2.4 CO (address 8009 H)

This subroutine causes an ASCII character in the C register to be output to the console device (serial I/O is standard). The contents of the A and PSW registers are modified and three stack levels are used by this operation.

3.2.5 PO (address 800C H)

This routine is the same as the CO subroutine except that the ASCII character is output to the punch device (serial ASCII I/O is standard).

3.2.6 LO (address 800F H)

This routine is also similar to the CO routine with the exception that the data is output to the list output device. As before, serial ASCII I/O is the standard code format.

3.2.7 CSTS (address 8012 H)

This is a console status request subroutine which evaluates the status of the console input device and returns A = 0 (zero value in the A register) or A = 0FF H if an input character is waiting. Since the CI subroutine will only return if a character is input, a call to CSTS can be used to determine if a call to CI is successful (will result in a character being input and returned).

3.2.8 IOCHK (address 8015 H) IOSET (address 8018 H)

A single memory location in RAM is used to define the four input / output (I/O) devices. The logical devices available are:

CONSOLE:	Referenced by CI, CO, CSTS
READER:	Referenced by RI
PUNCH:	Referenced by PO
LIST:	Referenced by LO

These logical devices can be "assigned" to any one of the following physical devices:

Serial I/O:	Uses the 8251 USART
Keyboard / Display:	Optional MCEM-KB/VDU Keyboard/Video Display Unit
Parallel I/O:	Uses the 8255 Programmable Peripheral Interface IC
User Defined I/O:	

USRIN (address 40 H): A user input subroutine which will return an ASCII character in the A register, similar in operation to the CI subroutine.

USROT (address 43 H): A user subroutine, similar in function to the CO subroutine, which will allow output to the user I/O of an ASCII character in the C register.

USRST (address 46 H): A user status routine which returns
 A = 0 if USRIN will not return a character immediately and
 A = 0FF H if USRIN will immediately return a character.

Serial I/O data is processed through the 8251 USART integrated circuit and may be either serial Baudot (5-unit) code (DSR pin = "1") or serial ASCII (8-unit) code (DSR pin = "0"). In Baudot code, the code conversion to and from ASCII code is performed by the I/O subroutine and need not be done otherwise. For instance, if a call to C0 is performed while the console is assigned to the serial I/O, an ASCII character should always be present in the C register. The monitor routine checks the status of the DSR line and performs a code conversion if necessary.

The Keyboard / Video Display Unit is a HAL Communications option available for the MCEM-8080. If a logical device is assigned to the Keyboard / Video Display Unit, the monitor will automatically write the display screen (output), read the keyboard (input), and check the keyboard status.

Parallel I/O data is processed through the 8255 PPI integrated circuit. Port A of the 8255 is used for data output, Port B for input, and Port C for control. The seven-bit ASCII code (bit 8 = "0", space) is used for parallel I/O. Mode 1 of the 8255 is used (Intel; p 5-123).

The user defined I/O capability is provided so that the user can write his own I/O subroutines to service particular devices (such as an electrically controlled Selectric (© IBM) typewriter, etc.). The monitor automatically calls a set of routines which start at location 40 H (USRIN, USROT, and USRST) for user I/O applications. When the monitor requests a character (CI, RI), a call to 40 H is executed. To output a character, a call to 43 H is executed; if the status of the I/O device is needed, a call to 46 H is executed. The routines in these locations should conform to the CI, CO, and CSTS format. For example:

Address:	40 H	JMP	INPUT
	43 H	JMP	OUTPUT

46 H	[INPUT	[INPUT STATUS ROUTINE
				USER INPUT ROUTINE
]	OUTPUT		USER OUTPUT ROUTINE

Memory location 3 is used to store the I/O device assignments. The format of the assignment byte is:

D7	D6	D5	D4	D3	D2	D1	D0
LIST		PUNCH		READER		CONSOLE	

Contents of Memory Location 3

D0 and D1 define the console device (C0, C1, CSTS)
D2 and D3 define the reader device (R1)
D4 and D5 define the punch device (P0)
D6 and D7 define the list device (L0)

Each two bit set can have one of the following four values:

00 assigns serial I/O
01 assigns Keyboard Video Display option
10 assigns parallel I/O
11 assigns user I/O.

For example:

Memory location 3 = 10110001 B (B1 H) defines that:

- (a) Console operations are via the optional Keyboard / Video Display Unit,
- (b) Reader operations are via the serial I/O device,
- (c) Punch operations are via the user I/O device,
- (d) List operations are via the parallel I/O.

The monitor automatically sets memory location to 0000 0000 B (00 H) upon turn-on and thus assigns all logical devices to the serial I/O port. The monitor also checks to see if the optional Keyboard / Video Display Unit has been attached to the UPB. If so, memory location 3 is set to 01010101 B (55 H) which assigns all logical devices to the Keyboard / Video Display Unit.

If, at any time, it is desired to reassign the I/O system, the I command can be used. For example:

. I 03 ↵

will result in a request to change location 3 which contains the I/O assignments.

A call to IOCHK will return the value of memory location 3 in the A register. A call to IOSET will write the contents of the C register into memory location 3. If it is desired to change the I/O system assignments, these routines should be used.

3.2.9 MEMCK (address 801B H)

This routine returns the contents of memory location 5 into the B register and memory location 6 into the A register. These locations are intended to hold the address of the first non-RAM memory address and are used by the resident assembler and editor to determine how much memory is available to them. The I (INSERT) command should be used to set these values if this routine is used.

4. SYSTEM ADDRESS ASSIGNMENTS

The MCEM-8080 uses Random Access Memory (RAM), Read Only Memory (ROM), Input ports, and Output ports. The address assignments for these sections are discussed below.

4.1 Random Access Memory (RAM)

The random access memory is used by both the monitor and for user storage. The standard MCEM-8080 systems are furnished with 1024 ("1 K") bytes of RAM - this can be doubled to "2 K" of RAM by the addition of more integrated circuits to the circuit board.

4.1.1 Monitor RAM Usage

Memory locations between 0 and 3FF H are reserved for RAM memory. The area between 0 and 40 H is used by the software monitor for the stack and for temporary storage. User programs should not use these storage locations to avoid interference with the monitor. As explained in section 3.2.8, the entry points for user I/O assignments are 40 H (User Input), 43 H (User Output), and 46 H (User Input Status).

4.1.2 User RAM Usage

All RAM in locations higher than 40 H is available to the user for program storage. The monitor stack does not take the possible requirements of a user stack into account. Therefore, user programs should establish a stack in the free RAM area (higher than 40 H). The EXIT command and RETURN commands assume that at least three levels (6 bytes) of user stack are available and that the user stack is not the same as the monitor stack.

4.1.3 Optional RAM

The standard MCEM-8080 circuit board has 1024 bytes ("1 K") of RAM integrated circuits installed. However, additional circuit board space and connections are provided that 1024 bytes of RAM ICs can be added, for a total of "2 K" (2048) bytes of RAM. Only type 8102A-4 integrated circuits should be used to assure compatibility with the rest of the MCEM-8080. It is highly recommended that these integrated circuits be installed by the HAL Communications factory to assure proper system operation. When the second "1 K" of RAM is used, it occupies the address space between 400 H and 7FF H. The installation of this additional RAM does not affect the monitor RAM usage and therefore, all of the additional RAM storage is available for user programs.

4.2 Read Only Memory (ROM)

The MCEM-8080 uses Read Only Memory (ROM) for non-volatile program stage. (Non-volatile = stored data is retained even when power is removed from the MCEM-8080. RAM is a volatile memory; ROM is non-volatile.) Typical uses of the ROM include storage of the monitor, support subroutines for peripheral devices, and user programs.

4.2.1 Monitor ROM

The memory locations between 8000 H and 83FF H are occupied by the software monitor program. This program uses a part of the ROM storage space on the main MCEM-8080 circuit board. It is contained in either 2 - 3624 PROM's, 1 - 8708 EPROM, or 1 - 8308 ROM integrated circuits. When the type 3624 ICs are used, the monitor program consumes one-half of the available on-board ROM space. When either the 8708 or 8308 ICs are used, the monitor consumes one-quarter of the on-board ROM space.

4.2.2 Peripheral ROM

Many MCEM-8080 peripherals require support programs ("software") to operate. Typical such peripheral devices include the Keyboard / Video Display Unit option and the PROM Programmer option. The Keyboard / Video Display Unit support software is physically resident on its circuit board and logically located between memory locations F800 H and F9FF H. Similarly, the ROM containing the software to support the PROM Programmer is also resident on the programmer circuit board and the program is located in memory locations between F200 H and F3FF H. As additional peripheral devices are developed, they will be assigned RAM and/or ROM storage in descending locations below F200 H.

4.2.3 User ROM

Space is provided on the main MCEM-8080 circuit board for user defined ROM storage. These ROMs, however, must be of the same type as that used for the monitor software. For instance, if the monitor has been supplied in type 3624 ROMs, all four ROM positions on the MCEM-8080 board must use the 3624 ROM. However, types 8708 and 8308 ROMs can be intermixed. User ROM storage starts at location 8400 H and extends to 87FF H (for 3624s) or to 8FFF H (for 8708/8308 ICs). HAL Communications provides ROM programming services to MCEM-8080 owners - please consult the factory if it is desired to program a PROM.

4.3 Input / Output (I/O) Assignments

Various input / output ports have been preassigned in the MCEM-8080 system. Among these are the 8251 USART IC, the 8255 PPI IC, the Keyboard / Video Display Unit option, and the PROM Programmer option.

4.3.1 8251 USART Integrated Circuit

The 8251 IC requires two input and two output ports; one input and one output port for control and one input and one output port for data. The control port has been assigned to port 0B H and the data port is assigned to 0A H.

4.3.2 8255 Programmable Peripheral Interface (PPI) IC

The 8255 IC requires four output and three input ports. Three of the ports map directly to the three parallel I/O ports of the IC. The fourth output port is used for PPI mode selection. The 8080 ports

corresponding to the 8255 ports are:

8080A Port	8255 Port
Input 0C H	Port A input
Input 0D H	Port B input
Input 0E H	Port C input
Output 0C H	Port A output
Output 0D H	Port B output
Output 0E H	Port C output
Output 0F H	8255 Mode Select

NOTE: There is no Input port 0F H.

4.3.3 MCEM-KB/VDU Keyboard/Video Display Unit

The optional Keyboard / Video Display Unit requires one output port and two input ports. These are assigned as 8080A ports 0, 4, and 6. See the Keyboard / Video Display Unit manual for further information on this option.

4.3.4 MCEM-7K PROMPROG PROM Programmer

The optional PROM Programmer requires four output ports and three input ports. These are assigned as 8080A ports 80, 81, 82, and 83. See the PROM Programmer manual for further information on this option.

5. OPERATING HINTS

Much of the versatility of the MCEM-8080 and the software monitor system will be best understood only after practical experience with the computer has been gained. This section of the manual contains some examples that will help to gain this needed experience.

5.1 Power-on Start Up

Several items should be checked out and possibly changed when initially installing the MCEM-8080. Among these are the power supplies, baud rate, I/O connections, etc. Once these items have been checked and corrected (if necessary), the following sequence can be used to "power-up" the system:

- a. Set RUN/STOP switch to STOP
- b. Set the DATA BUS OVERRIDE switch (DBO) to ON
- c. Set the DATA BUS REGISTER switches to all zeros (front of rocker switch down)
- d. Set the break point register (ADDRESS switches) to 8000 H
- e. Apply DC power.

The address indicators should momentarily light and then extinguish. When all address lamps are on (logical "1"), the 8080A is being RESET. Once the address indicators are off, the WAIT lamp (far right-hand side of the circuit board) should come on and all DATA lamps should be off.

- f. Set the RUN/STOP switch to RUN.
The 8080A will now run and automatically stop at location 8000 H (the setting of the break-point register).
- g. Set the Data Bus Override (DBO) to OFF.
The DATA indicators should now indicate C3 H (1100 0011 B) which represents the first instruction in the software monitor, a JMP instruction.
- h. Press and release the STEP switch. This causes the 8080A to begin executing the software monitor.
- i. If the proper console device is operational (serial I/O if the Keyboard / Video Display Unit is not attached), the monitor will send the character sequence: "CR, LF, blank, blank, blank, period" to indicate that the monitor is ready to accept a command. The system is now ready to use.

NOTE:

The software monitor writes a jump to monitor instruction into location 0 (0, 1, 2) as it is initializing so that once the monitor has been entered (at 8000 H), a RESET (set the program counter to zero) will automatically cause an entry into the software monitor. Therefore, once an initial entry has been made, it is no longer necessary to go through the DBO = ON, Data Bus = 0, Break-point = 8000 H routine again. If power is removed or a user program writes data in location 0, 1, or 2, the automatic monitor entry on RESET will not operate.

5.2 Changing the Monitor Mode

The software monitor has several operational options. Some options should be selected before power turn-on and some after.

5.2.1 Baudot / ASCII Code

The serial I/O processing routines can be operated in either BAUDOT or ASCII units. The DSR terminal on the 8251 USART (pin 22 of the IC, terminal three of the I/O connector) is used to indicate to the monitor which code is being used. For ASCII code, terminal three is connected to ground; for BAUDOT, to +5 volts. The ASCII connection is normally furnished on the MCEM-8080. This connection must be made BEFORE power is applied to the MCEM-8080. If the state of the DSR connection is changed with power on and without first performing a RESET, the result may be indeterminate (for example transmitting a 5 bit character to an 8 bit USART, etc.). The placement of this jumper is shown in Figure B.1.

5.2.2 Half / Full Duplex (Echo / No echo)

In normal operation, characters that are input to the software monitor in the form of commands or parameters are retransmitted out to the output device so that the operator can view and verify them. This is called echoing of the character. If however, the MCEM-8080 is to be used in a system which automatically echos the input character, external to the MCEM-8080, this feature may be defeated (otherwise a double echo would result, causing repeating of the input characters). An example of a self-echoing I/O system is the serial loop-connected teleprinter in which the keyboard and printer are connected in series. To defeat the echo feature, FF H should be written into location 0D H. This location is normally initialized to 00 upon monitor entry and will be re-initialized upon each new entry into the monitor. The I (Insert) command can be used to perform the change by typing:

```
. I D ↵  
  _ _ _  
000D 00, F F  
000E 00 S _
```

The characters to be typed by the operator are underlined. If the double character transmission is occurring, it will appear as:

```
. I I D D ↵ ↵  
  _ _ _ _  
000D 00 , , F F F F  
000E 00 S _
```

As above, underlined characters indicate those typed. Notice that only one S appeared because the echo has been turned off by that time.

5.2.3 Changing I/O Device Assignments

As discussed in section 3.2.8 with regard to monitor routines IOSET and IOCHK, an eight bit byte is reserved to hold the system I/O assignment. Changing this byte will change the device assignments. Use the I (Insert) command to change the byte as explained in section 3.1.3 and as in the preceding section (5.2.2). Remember that if the console device (Ports 0 and 1) is changed, the new console will be polled for new command strings. The IOBYT is set to zero (all devices set to serial I/O) upon initialization by the monitor. If, however, the Keyboard / Video Display Unit option is attached, all devices are set to it (the monitor checks for the presence of the Keyboard / Video Display Unit). In this case, IOBYT = 55.

5.3 Manually Writing a Memory Location

Two methods can be used to write a memory location, the easiest being to use the I (Insert) command. If it is impractical to use the I command, the following procedure can also be used:

- a. Set the RUN / STOP switch to STOP
- b. Set the DBO switch to ON
- c. Set the DATA switches to zero
- d. Set the break-point register to the desired address
- e. Press and release the RESET switch
- f. Set the RUN / STOP switch to RUN
(The Address indicators should now equal the desired memory address)
- g. Set the DATA switches to the desired new memory value
(number to be stored)
- h. Press and release the MEMORY WRITE switch
- i. Set the DBO switch to OFF
- j. Go to the desired address to proceed with program execution.

5.4 Manually Jumping to a Program Address

If, for some reason, the monitor JMP (Jump) command is unavailable to perform a jump to a desired program point, the following sequence can be used. Note that this is the same sequence as described in section 5.1 for initial entry into the monitor.

- a. Set the RUN / STOP switch to STOP
- b. Set the DBO switch to ON
- c. Set the DATA switches to zero
- d. Set the break-point register to the desired address
- e. Press and release the RESET switch
- f. Set the RUN / STOP switch to RUN
(The address indicators should now equal the desired address)
- g. Set the DBO switch to OFF
- h. The program counter is now set. Press the STEP switch to begin program execution.

5.5 Manually Writing to an Output Port

At times it is desirable to be able to manually write data into an output port. This can be accomplished by:

- (1) Follow steps a. through h. of the previous two examples using the output port address instead of the memory address. Remember that I/O addresses are copied twice, once as the high order address and once as the low order address. For instance, Output Port 23 H is represented on the address bus (and break-point register) as 2323 H.
- (2) Press and release the OUTPUT WRITE switch.
- (3) Go to the desired address to proceed with program execution.

5.6 Using the Break-point Register for Debugging

The break-point register provides a mechanism for selectively stopping the 8080A. During the course of debugging a program, it may be desirable to determine when and if a particular string of instructions is executed. Setting the break-point register to this address (or I/O port) will provide this information. Another use of the break-point register allows use of the STEP switch as a "loop execute" switch. If the software being debugged contains a loop, the break-point register can be set to an address within the loop and the RUN / STOP switch set to RUN. At this point, the 8080A will stop each time it passes through the loop and will continue each time the STEP is pressed and released.

5.7 Using the E and R Commands for Debugging

The software monitor provides two very powerful commands to aid in debugging programs. The E (EXIT) command is a mechanism for saving complete context at any point in a user program and entering into the monitor. The R (RETURN) command allows return to the user program after restoring the complete context previously saved by the E command. The E command is invoked by executing RST 7. For example:

5.7.1 Manual EXIT Command

When a program is being debugged by manually stepping through the program steps (using the STEP switch), it is sometimes desirable to examine the contents of some of the internal registers of the 8080A (for instance the B, A, or PSW registers). However, since these registers are internal to the 8080A, they can not be directly examined on the console. The following procedure can be used to examine these internal registers:

- a. Set RUN / STOP switch to STOP.
(If the STEP switch is being used for debugging, the RUN / STOP switch is probably already set to STOP.)
- b. Press and release the STEP switch as many times as necessary to bring the execution to the first byte of an instruction.

The E command can only be invoked during the fetch cycle of an instruction. For instance, JMP 23F2 is represented by

C3
F2
2B

in memory. The E command can only be invoked when C3 is being read (indicated on the data indicators).

- c. Set the DBO switch to ON.
- d. Set the DATA switches to FF (all "ones" = RST 7).
- e. Press and release the STEP switch once.
- f. Set the DBO switch to OFF.
- g. Set the RUN / STOP switch to RUN.
- h. At this point, the following character stream should be typed on the console device:

EXIT xxxx

Remember that if the users program reassigns the I/O assignments or disturbs the USART mode, the console operation may be inhibited. "xxxx" in the above character stream represents the next address after the one in which RST 7 was inserted.

- i. Type

.D2C,38> :

The saved register values will now be displayed in the following format:

:0C 002C 00 SPL SPH PCL PCH L H PSW A E D C B SC
:00 0038 00

where:

SPL = low order stack pointer, stored at 2C
SPH = high order stack pointer, stored at 2D
PCL = low order program counter, stored at 2E
PCH = high order program counter, stored at 2F
L = L register, stored at 30
H = H register, stored at 31
PSW = Processor Status Word, stored at 32
A = A register, stored at 33
E = E register, stored at 34
D = D register, stored at 35
C = C register, stored at 36
B = B register, stored at 37
SC = Sum Check Character.

The format of the Processor Status Word (PSW) is:

```
D7 D6 D5 D4 D3 D2 D1 D0
S  Z  0  AC  0  P  1  CY  ,
```

where S, Z, AC, P, and CY are the corresponding flags.

- j. If, at this point, it is desired to set a particular register to a new value, the I command can be used. For example,

```
.I33↓
```

will allow the A register to be modified.

- k. After the registers have been examined and changed (if desired), the R (RETURN) command can be used to return to the original program. The return address, however, is not the same as the saved address in this case. (Recall that a RST 7 instruction was inserted instead of a valid instruction and the saved address is one more than the address of the substituted RST 7.) To use the R command, mentally calculate

$$yyyy = xxxx - 1 ,$$

where "xxxx" is the address stored and the address displayed on the console after the E identifier (see step h). Now type

```
.Ryyyy↓
```

Leading zeros can be omitted.

5.7.2 Interrupt EXIT Command

The monitor software and the MCEM-8080 hardware combine to cause interrupts to execute E commands (the interrupt vector is RST 7). Therefore, if it is desired to execute an E command, it can be instituted by placing +5 volts on the INTR (INTERRUPT) line (pin 4 of the 36 pin I/O connector). If the user has not disabled the interrupt or written into low memory (below 3F H), the following should appear on the console device:

```
EXIT xxxx
```

All of the techniques for examining and modifying registers listed above may be used. However, when a R command is desired, it is not necessary to recompute the address because the interrupt method saves the proper address.

A return to zero command,

```
R0↓
```

will return the CPU to the program, restoring the registers to their states just prior to the interrupt.

5.7.3 Programmed EXIT Command

Many programs have error testing subroutines and the E command can be used to perform error exits from these programs. If a RST 7 instruction is inserted in the program in the error branch, it will cause the following to be displayed on the console:

```
EXIT xxxx .
```

If the user tabulates the addresses of all of the RST 7 instructions, it is then a simple matter to correlate the "xxxx" typed against the list. The techniques explained previously can be used to evaluate and modify the CPU parameters that existed at the time of the interrupt (RST 7).

A useful feature that results when the RST 7 instruction is used as the E command driver is prevention of transfer to non-existent areas of memory. Since non-existent memory is generally FF, a RST 7 will be immediately encountered and control then transferred to the E command process. This feature helps prevent the "run away" condition that could conceivably rewrite all of memory otherwise.

6. PROGRAM EXAMPLES

This section contains several example programs to demonstrate the features and capabilities of the MCEM-8080. In no case should any of these example programs be considered "optimum" or "required procedure". They are, however, working routines that can be used as starting points for more elaborate programs, as subroutines in user programs, or simply for ideas as to typical procedures to be used with the MCEM-8080A. Except as noted, all example programs will operate in the basic "1 K" memory furnished with the MCEM-8080.

7. SOFTWARE MONITOR LISTINGS

The following pages contain a complete listing of the MCEM-8080 Microcomputer System software monitor. This listing is provided for the sole benefit of owners of HAL Communications Model MCEM-8080 systems and remains the sole property of HAL Communications Corp. The listing may not be duplicated for any use without the prior permission of HAL Communications Corp. HAL Communications reserves the right to make changes, additions, or deletions to these computer programs without prior notification or obligation to incorporate such changes in prior versions of the programs.

COPYRIGHT 1976 (C)
BY

HAL COMMUNICATIONS CORP.
807 E GREEN STREET
URBANA, ILLINOIS 61801

MCEM MONITOR - DECEMBER 1976
HAL COMMUNICATIONS CORP.

THIS PROGRAM IS CONTAINED IN PROMS (01.0) (FIRST HALF)
AND (02.0) (SECOND HALF)

THE FOLLOWING ARE VALID COMMANDS FOR THE
MCEM MONITOR:

L<BIAS> LOAD A HEX FORMATTED FILE,
CHECK FOR SUMCHECK ERRORS
AND TYPE AN 'X' IF ERROR.
THE VALUE OF THE BIAS IS
ADDED TO THE LOAD ADDRESS
BEFORE THE DATA IS WRITTEN
TO MEMORY. THE READER DEVICE IS
USED AS INPUT

D<START>.<END> DUMP OR DISPLAY MEMORY.
MEMORY LOCATIONS FROM <START>
TO BUT NOT INCLUDING <END> IS
DUMPED. THE FORMAT OF THIS
DUMP IS COMPATIBLE WITH THE
LOAD COMMAND SO MEMORY AREAS
CAN BE DUMPED AND LOADED AT
A LATER TIME. THE PUNCH DEVICE
IS USED AS OUTPUT.

***THE FORM FOR BOTH LOADS AND DUMPS IS:

:<LENGTH><ADDRESS><TYPE><DATA BYTES><SMCHECK>

ALL RECORDS ARE PRECEDED BY A COLON, ALL
CHARACTERS BETWEEN THE SUMCHECK AND THE COLON
ARE IGNORED. ALL SPACES ARE IGNORED (I.E. SPACES
CAN BE CONTAINED IN THE RECORD WITH NO EFFECT)

<LENGTH> IS THE NUMBER OF DATA BYTES IN THE
RECORD (00-FF)

<ADDRESS> IS THE LOAD ADDRESS (0000-FFFF)

<TYPE> IS NOT USED AT THIS TIME AND IS 00 (IGNORED)

<DATA BYTES> ARE THE ACTUAL DATA, <LENGTH> OF THEM

<SMCHECK> IS THE NEG SUM OF ALL BYTES (ADDRESS IS TWO BYTES (HIGH AND LOW)) EXCLUDING THE COLCN, I.E. THE SUM OF ALL BYTES INCLUDING THE SUMCHECK IS 0 FOR NO ERROR. DURING LOAD, IF A SUMCHECK ERROR IS ENCOUNTERED, AN 'X' IS PRINTED ON THE CONSOLE DEVICE.

<LOCATION> INSERT IN THE SPECIFIED LOCATION. THE PREVIOUS CONTENTS OF THE LOCATION IS TYPED. THE CONSOLE DEVICE IS USED FOR INPUT AND OUTPUT. A ',' IS USED TO OPEN A CELL: I.E.

.165
0065 58,74

WOULD BE THE FORMAT FOR CHANGING LOCATION 65 FROM 58 (OLD VALUE) TO 74 (NEW VALUE)

<JLOCATION> GO TO <LOCATION>, A UNCONDITIONAL JUMP IS EXECUTED TO THE INDICATED LOCATION. INTERRUPT (EXIT COMMAND) IS ENABLED BEFORE THE JUMP.

<RLOCATION> RETURN TO LOCATION. A RESTORE REGISTER JUMP IS EXECUTED TO LOCATION UNLESS LOCATION = 0 IN WHICH CASE, THE CONTENTS OF THE PCSAV IS USED AS THE ADDRESS. INTERRUPT (EXIT COMMAND) IS ENABLED BEFORE THE RETURN.

EXIT COMMAND: A RST 7 WILL EXECUTE AN EXIT COMMAND. ALL REGISTERS ARE SAVED IN RAM FOR EXAMINATION AND/OR MODIFICATION. THE R COMMAND IS THE COUNTER OF THE EXIT COMMAND AND WILL RETURN THE PROCESSOR TO ITS ORIGINAL STATE. AN EXIT COMMAND SHOULD BE PERFORMED PRIOR TO AN R COMMAND BECAUSE THE STORED VALUE OF THE SP SHOULD BE INTACT (I.E. POINT TO A VALID STACK AREA. (OTHER THAN THE MONITOR STACK).

```

; I/O ASSIGNMENT: AN EIGHT BIT I/O ASSIGNMENT BYTE
; IS STORED AT LOCATION ICBYTE (03)
; THE VALUE OF THIS BYTE DIRECTS THE
; CONSOLE, READER, AND PUNCH (LIST
; TOO) TO ONE (EACH) OF FOUR POSS-
; IBLE I/O DEVICES. THE FORMAT OF
; IOBYT IS:

```

```

      D7  D6  D5  D4  D3  D2  D1  D0
; /LIST  DEV/PUNCH DEV/READ DEV/CONSOLE /
; EACH DEVICE CAN BE ASSIGNED TO:

```

```

00:SERIAL INPUT/OUTPUT (8251, ASCII CR BAUDOT)
01:KEYBOARD DISPLAY MODULE (OPTIONAL DEVICE)
10:PARALLEL INPUT/OUTPUT (8255)
11:USER INPUT/OUTPUT (USER DEFINED ROUTINES)

```

```

; ALL NUMERIC ENTRIES CAN BE TERMINATED BY
; TYPING ANY OF THE CHARACTERS G-Z

```

```

; TITLE 'MCEM-8080 MONITOR 1.1'

```

```

; CONSTANT DEFINITIONS

```

```

; KEYBOARD DISPLAY I/O CONSTANTS

```

```

F800  DSPCK  EQU  0F800H  ;DISPLAY PRESENT CHECK
F80A  SDFIO  EQU  DSPCK+0AH;SET DISPLAY IO BYTE ENTRY
FEC1  KBIN   EQU  DSPCK+1  ;KBIN ENTRY
F807  KBSTS  EQU  DSPCK+7  ;KBSTS ENTRY
F804  DSPOT  EQU  DSPCK+4  ;DISPLAY OUTPUT ENTRY

```

```

; SERIAL I/O CONSTANTS

```

```

CC8C  DSR     EQU  80H     ;DSR BIT IN USART (0=BAD)
CC83  URIMM  EQU  83H     ;UART MODE FOR BAUDOT
001F  LTRS   EQU  1FH     ;BAUDOT LETTERS
001B  FIGS   EQU  1BH     ;BAUDOT FIGURES
CC07  BCASI  EQU  7      ;BAUDOT CASE (INPUT)
CCCE  BCASO  EQU  0EH     ;BAUDOT CASE (OUTPUT)
0002  RXRDY EQU  2H      ;RX READY TEST MASK
C001  TXRDY EQU  1H      ;TX READY TEST MASK
0027  TXRXE EQU  27H     ;TX,RX ENABLE
CCFA  URIMO  EQU  0FAH   ;7 BITS, EVEN PARITY, 2 STOP
CC8E  CARTI  EQU  8EH     ;INITIAL UART MODE WORD
CC55  UARTR  EQU  55H     ;UART RESET COMMAND
CC0B  URTCT  EQU  0BH     ;UART CONTROL PORT

```

```

CCCC      LRTDA   EQU    0AH      ;UART DATA PORT
          ;
          ; PARALLEL I/O CONSTANTS
          ;
00A6      PARM    EQU    0A6H     ;PPI MODE FOR PARALLEL I/O
000F      PARCT   EQU    0FH      ;PARALLEL CONTROL PORT
0002      PIRDY   EQU    2        ;PARALLEL INPUT READY BIT
000E      PSTAT   EQU    0EH      ;PARALLEL STATUS PORT
000C      PINPT   EQU    0CH      ;PARALLEL INPUT PORT
0080      PORDY   EQU    80H      ;PARALLEL OUTPUT READY BIT
000C      POTPT   EQU    0CH      ;PARALLEL OUTPUT PORT
          ;
          ; USER I/O CONSTANTS
          ;
0040      USRIN   EQU    40H      ;ENTRY FOR USER INPUT
0043      USROT   EQU    43H      ;ENTRY FOR USER OUTPUT
0046      USRST   EQU    46H      ;ENTRY FOR USER INPUT STATUS
          ;
          ; MISC MONITOR CONSTANTS
          ;
00CA      LF      EQU    0AH      ;ASCII LINE FEED
000D      CR      EQU    0DH      ;ASCII CARRIAGE RETURN
          ;
          ; MONITOR MEMORY ALLOCATIONS
          ;
0005      MSZ1    EQU    5        ;MEMSIZE STORE
0006      MSZ2    EQU    6        ;MEMSIZE STORE (HIGH)
0004      IFLAG   EQU    4        ;INDIR REF TO EIT FLAG
0038      RGSAV   EQU    38H     ;START OF RESTART STORAGE
002C      MONST   EQU    RGSAV-12;MONITOR STACK AREA
002E      PCSAV   EQU    RGSAV-10;LOCATION OF PC SAVE
0032      PSSAV   EQU    RGSAV-6  ;LOCATION OF PSW SAVE
0003      IOBYT   EQU    3        ;I/O ASSIGNMENT STORAGE
000D      ECHGM   EQU    0DH     ;STORAGE FOR ECHC MODE FLAG
          ;
          ; MONITOR RAM ALLOCATION:
          ;
          ; 0-2 JUMP TO MONITOR CCDE
          ; 3 IOBYT (I/O DEVICE ASSIGNMENT)
          ; 4 IFLAG (INDIRECT CONSOLE REFERENCE FLAG)
          ; 5 MSZ1 (LOW ORDER MEMORY SIZE BYTE)
          ; 6 MSZ2 (HIGH ORDER MEMORY SIZE BYTE)
          ; 7 BCASI (BAUDOT CASE FOR INPUT)
          ; 8-9 CRSRP (CURSOR POSITION FOR CRT DISPLAY)
          ; A CFSAV (OFFSET REGISTER COPY FOR CRT DISPLAY)
          ; B HIDCH (CHARACTER HIDDEN UNDER CURSOR)
          ; C RPTFG (REPEAT MODE (KEYBOARD) FLAG)
          ; D ECHGM (ECHC/NO ECHC MODE FLAG)
          ; E BCASO (BAUDOT CASE FOR OUTPUT)
          ; F SFARE
          ; 10-2C MONITOR STACK

```

```

; 2D-37 REGISTER SAVE STORAGE
; 38-3A EXIT COMMAND ENTRY JUMP (RST 7)
; 3B-3F SPARE

```

```

;
;
; MACRO DEFINITIONS

```

```

1 TEST MACRC
1 ANA A ;SET FLAGS , CY=0
ENDM
8000 ORG 8000H
;
8000 C32B80 JMP BEGIN ;MONITOR ENTRY
8003 C39381 JMP CI ;CONSOLE INPUT
8006 C3AD81 JMP RI ;READER INPUT
8009 C3D681 JMP CO ;CONSOLE OUTPUT
800C C3F181 JMP PO ;PUNCH OUTPUT
800F C3F981 JMP LO ;LIST OUTPUT
8012 C3C182 JMP CSTS ;CONSOLE STATUS
8015 C38F82 JMP IOCHK ;IO ASSIGN CHECK
8018 C35382 JMP IOSET ;IO ASSIGN SET
801E C35882 JMP MEMCK ;MEMORY SIZE CHECK
801E C31C83 JMP EXPR ;EXPRESSION GETTER
8021 C39A83 JMP TYPMG ;MESSAGE TYPER
8024 C37283 JMP BYTOT ;BYTE TYPER
8027 C3CA81 JMP WRDOT ;WORD TYPER
802A C9 RET ;ENTRY FOR NO SERVICE RINS

```

```

;
; INITIALIZE UART

```

```

802B BEGIN:
802B 2140C0 LXI H,40H ;CLEAR MONITOR RAM AREA
802E BGO:
802E 2D DCR L
802F 74 MOV M,H
8030 C22E80 JNZ BGO
8033 36C3 MVI M,0C3H ;SET UP MONITOR REENTRY
8035 23 INX H ;BUMP ADDRESS
8036 362B MVI M,BEGIN AND OFFH ;ENTER LOW ADDRESS
8038 23 INX H ;BUMP ADDRESS AGAIN
8039 3680 MVI M,BEGIN SHR 8 ;ENTER HIGH ADDRESS
803E 2E38 MVI L,38H ;SET UP RESTART ENTRY (RST 7)
803D 36C3 MVI M,0C3H
803F 23 INX H
8040 3645 MVI M,RSTRT AND OFFH
8042 23 INX H
8043 3681 MVI M,RSTRT SHR 8
8045 3EA6 MVI A,PARMD ;SET PARALLEL I/O MODE
8047 03CF OUT PARCT
8049 3E8E MVI A,UARTI ;UART MODE INSTRUCTION

```

```

8048 D30B OUT URTCT ;ISSUE MODE
804D 3E55 MVI A,UARTR ;UART RESET INSTRUCTION
804F D30B OUT URTCT ;ISSUE INTERNAL & ERROR RESET
8051 3FFA MVI A,URTMO ;SET FINAL MODE
;ABOVE PREDEFINED PRESENTLY
8053 D30B OUT URTCT ;ISSUE UART MODE INSTRUCTION
;NEXT MUST BE COMMAND INSTR
8055 DB0B IN URTCT ;IS SERIAL I/O BAUDOT?
8057 E680 ANI DSR ;DSR=0 IS BAUDOT (+5VOLTS)
8059 C26480 JNZ BG2 ;NO, MODE IS ASCII
805C 3E55 MVI A,UARTR ;RESET UART
805E D30B OUT URTCT
8060 3E83 MVI A,URTBM ;SET MODE FOR 5 BIT, 1 1/2
8062 D30B OUT URTCT ;STCP, NO PARITY, X64 CLK
8064 BG2:
8064 3E27 MVI A,TRXRE ;ENABLE TX AND RX
8066 D30B OUT URTCT
8068 312C00 LXI SP,MONST;SET STACK PCINTER
806B 3AC0F8 LDA DSFCK ;IS DISPLAY ATTACHED?
806E FE45 CPI 0A5H ;THIS IS THE TEST BYTE
8070 CC0AF8 CZ SDPIO ;USE THE DISPLAY ROUTINE

```

```

;
; END OF INITIALIZATION SEQUENCE
;

```

```

;
; MONITOR MAIN LOOP
;

```

```

8073 MAIN:
8073 312C00 LXI SP,MONST;RESET MONITOR STACK POINTER
8076 21B683 LXI H,PMIMG ;PRCPT WITH PERIOD
8079 CD9A83 CALL TYPMG
807C CD8581 CALL ECHO ;GET INFUT
807F 0602 MVI B,2 ;DEFAULT PARAMETER COUNT
8081 FE4C CPI 'L' ;LOAD
8083 CA0E81 JZ LCAD ;YES
8086 FE4A CPI 'J' ;START EXECUTION (JUMP)?
8088 CAE880 JZ GO ;YES
808B FE49 CPI 'I' ;INSERT?
808D CAED80 JZ INSRT ;YES
8090 FE52 CPI 'R' ;RETURN TO PROGRAM?
8092 CA6F81 JZ RETRN ;YES
8095 FE44 CPI 'D' ;DISPLAY/DUMP?
8097 C27380 JNZ MAIN ;NO, BAD COMMAND, TRY AGAIN

```

```

;
; COMMAND SERVICE ROUTINES
;

```

```

;
; DISPLAY (DUMP) MEMORY TO SERIAL OUTPUT
; THE FORMAT OF THIS DUMP IS COMPATIBLE
;

```



```
80E5 C39F80 JMP DISP1 ;DO ANOTHER RECORD
```

```

:
: GO ROUTINE
: EXITS VIA JUMP TO THE USERS ROUTINE
: AFTER RESTORING THE USERS REGISTERS
:

```

```
80E8 CD2683 GO: CALL WRDIN ;GET JUMP ADDRESS
80EB FB EI ;ENABLE EXIT COMMAND
80EC E9 PCHL ;JUMP THERE
```

```

:
: INSERT DATA COMMAND
: ALLOWS HEX DATA TO BE SEQUENTIALLY ENTERED
: THE INSERT FUNCTION IS TERMINATED BY A CHAR
: BETWEEN G AND Z.
:

```

```
80ED CD2683 INSRT: CALL WRDIN ;GET INSERT ADDRESS
80F0 INST1: PUSH H ;SAVE ADDRESS
80F1 E5 LXI H,CRMG ;TYPE CR/LF
80F4 CCA283 CALL TYP1
80F7 E1 POP H ;RETRIEVE ADDRESS
80F8 CDCA81 CALL WRDOT ;TYPE ADDRESS
80FB 7E MOV A,M ;GET PRESENT CELL VALUE
80FC CD7B83 CALL BTOT1 ;TYPE IT
80FF CDF182 CALL EIT ;CHECK IF CHANGE IS DESIRED
8102 FE2C CPI ',' ;USE , TO OPEN THE CELL
8104 C2CB81 JNZ INST2 ;SKIP BYTE READ IF NOT ,
8107 CD4E83 CALL BYTIN ;GET NEW VALUE (EXIT FROM HERE)
810A 77 MOV M,A ;STORE IT
810B INST2: INX H ;MOVE TO NEXT LOCATION
810C C3F080 JMP INST1 ;START NEXT LINE
```

```

:
: LOAD HEX ROUTINE. THIS ROUTINE IS COMPATIBLE WITH
: BOTH THE DATA GENERATED BY THE D COMMAND AS WELL
: AS THE HEX FILES GENERATED BY THE ASSEMBLERS AND
: COMPILERS.
:

```

```
810F LOAD: DCR B ;GET BIAS VALUE
8110 05 CALL EXPR ;USE NORMAL PARAM
8113 CD1C83 LDO: CALL RI ;SEARCH FOR ':'
8116 CCAD81 SUI ':'
8118 DE3A JNZ LDC
811E C21381 MOV D,A ;ZERO SUMCHECK
811F 57
```

```

811C CDC682 CALL BYTIR ;GET RECORD LENGTH
811F CA7380 JZ MAIN ;EXIT IF ZERO LENGTH
8122 C1 POP P ;GET BIAS
8123 F5 PUSH PSW ;SAVE RECORD LENGTH
8124 CDD82 CALL WDIR ;GET LOAD ADDRESS
8127 09 DAD B ;ADD BIAS
8128 F1 POP PSW ;RETRIEVE RECORD LENGTH
8129 C5 PUSH B ;SAVE BIAS
812A 47 MOV B,A ;PUT LENGTH IN B
812E CDC682 CALL BYTIR ;GET RECORD TYPE
812E LD1:
812E CDC682 CALL BYTIR ;GET BYTE
8131 77 MOV M,A ;STORE IT IN MEMORY
8132 23 INX H ;MOVE TO NEXT ADDRESS
8133 05 DCR B ;DECREMENT RECORD LENGTH
8134 C22E81 JNZ LD1 ;LOOP UNTIL RECORD DONE
8137 CDC682 CALL BYTIR ;GET SUMCHECK BYTE
813A CA1381 JZ LDO ;SUMCHECK WAS ZERO
813C 0E58 MVI C,'X' ;TYPE E FOR SUMCHECK ERROR
813F CDD681 CALL CD
9142 C31381 JMP LDO

```

```

; STACK FORMATS FOR RESTARTS:

```

```

; USER STACK

```

```

; PCH

```

```

; PHL

```

```

; H

```

```

; L

```

```

; A

```

```

; PSW

```

```

; MONITOR STACK

```

```

; B (37)

```

```

; C (36)

```

```

; D (35)

```

```

; E (34)

```

```

; A (33)

```

```

; PSW (32)

```

```

; H (31)

```

```

; L (30)

```

```

; PCH (2F)

```

```

; PCL (2E)

```

```

; SPH (2D)

```

```

; SPL (2C)

```

```

; MONITOR STACK STARTS AT 2B

```

```

; EXIT COMMAND PROCESSOR. WE GOT HERE BY DOING A
; RST 7 (INIT SEQUENCE WRITES THIS ADDRESS INTO RAM
; AT 3EH) SO PC IS ALREADY ON USER STACK. SAVE HL
; AND PSW ON USER STACK TO MAKE ROOM TO WORK. SAVE
; REMAINING REGISTERS IN RAM IN SAVE AREA AND THEN
; TRANSFER THOSE STILL ON USER STACK TO SAVE AREA.

```

```

; A TOTAL OF 6 BYTES OF USFR STACK ARE NEEDED.  USER
; STACK MUST NOT BE THE SAME AS THE MONITOR STACK
; ONCE FULL RECCVERY HAS BEEN ACOMPLISHED.  WRITE THE
; ADDRESS THAT GOT US HERE ON THE CONSOLE.

```

```

;
; RSTRT:
8145          RSTRT:
8145      E5          PUSH    H          ;SAVE HL ON USER STACK
8146      F5          PUSH    PSW       ;SAVE PSW
8147      210000     LXI     H,0        ;GET SP VALUE IN HL
814A      39          DAD     SP
814E      313800     LXI     SP,RGSAV ;SET UP SP FOR SAVE AREA
814E      C5          PUSH    B          ;SAVE BC
814F      D5          PUSH    D          ;SAVE DE
8150      5E          MOV     E,M       ;MOVE A,PSW TO SAVE AREA
8151      23          INX     H
8152      56          MOV     D,M
8153      23          INX     H
8154      D5          PUSH    D          ;THEY ARE IN DE
8155      5E          MOV     E,M       ;MOV HL TO SAVE AREA
8156      23          INX     H
8157      56          MOV     D,M
8158      23          INX     H
8159      D5          PUSH    D          ;THEY ARE IN DE
815A      5E          MOV     E,M       ;MOV PC TO SAVE AREA
815B      23          INX     H
815C      56          MOV     D,M
815D      23          INX     H
815E      D5          PUSH    D          ;THEY ARE IN DE
815F      E5          PUSH    H          ;SAVE SP IN SAVE AREA
8160      21B883     LXI     H,RSMG    ;TYPE EXIT IDENTIFIER
8163      CD9A83     CALL    TYPMG
8166      2A2EC0     LHLD   PCSAV    ;GET THE CALLING ADDRESS
8169      CDCA81     CALL    WRDOT    ;TYPE IT ON CONSOLE
816C      C37380     JMP     MAIN     ;GO TO MAIN LOOP

```

```

;
; RETURN PROCESSOR.  RECCVER ALL REGISTERS FROM THE
; SAVE AREA AND THEN RETURN TO WHERE EVER THE PC SAV
; INDICATES.  ONE PARAMETER IS GATHERED, AND IF IT
; IS NGN-ZERO, ITS VALUE IS SUBSTITUTED FOR THE PC
; SAV BEFORE RETURNING.

```

```

;
; RETRN:
816F          RETRN:
816F      CD2683     CALL    WRDIN    ;GET ONE PARAMETER IN HL
8172      7C          MOV     A,H       ;TEST IF IT IS ZERO
8173      B5          ORA     L
8174      CA7A81     JZ     RETR1    ;HL = 0, DONT WRITE PC SAVE
8177      222E00     SHLD   PCSAV    ;STORE NEW DESTINATION VALUE
817A          RETR1:
817A      E1          POP     H          ;GET STACK POINTER
817B      D1          POP     D          ;GET PC
817C      C1          POP     B          ;GET HL

```

```

817D F9          SPHL          ;RESTORE STACK POINTER
817E D5          PUSH         D          ;PUT PC ON STACK
817F C5          PUSH         B          ;PUT HL ON STACK (TEMP)
8180 213200     LXI          H,PSSAV ;MOVE A,PSW FROM STACK
8183 5E          MOV          E,M        ;USE DE AS BUFFER PAIR
8184 23          INX          H
8185 56          MOV          D,M
8186 23          INX          H
8187 D5          PUSH         D          ;SAVE FOR A WHILE ON SACK
8188 5E          MOV          E,M        ;RECCVER DE
8189 23          INX          H
818A 56          MOV          D,M
818B 23          INX          H
818C 4E          MOV          C,M        ;RECCVER BC
818D 23          INX          H
818E 46          MOV          E,M
818F F1          POP          PSW        ;RECCVER A,PSW
8190 E1          POP          H          ;RECCVER HL
8191 FB          EI           ;ENABLE INTERRUPTS (RST 7)
8192 C9          RET           ;RECCVER PC

```

```

;
; CCNSCLE INPUT ROUTINE. THE DEVICE INDICATED BY
; THE BOTTOM TWO BITS OF IOBYT IS CALLED AND EXPECTED
; TO RETURN A CHARACTER. THE CHARACTER RETURNS IN A
;

```

```

8193          CI:
8193 3A0300     LDA          IOBYT ;GET IO ASSIGNMENT
8196          IBRCH:
8196 E603       ANI          3      ;TEST BOTTOM BITS
8198 CA3882     JZ          CHI     ;ZERC IS SERIAL INPUT
819B 3D        DCR          A
819C CA01F8     JZ          KBIN   ;CNE IS KEYBOARD INPUT
819F 3D        DCR          A
81A0 C240C0     JNZ         USRIN  ;NOT TWC IS THREE (PARALLEL)

```

```

;
; PARALLEL INPUT ROUTINE. 7 BIT ASCII IS EXPECTED
; ON THE B PORT OF THE PPI. TRUE DATA IS EXPECTED.
;

```

```

81A3          PARIN:
81A3 D0CE       IN          PSTAT  ;GET THE PORT STATUS
81A5 E602       ANI          PIRDY  ;TEST INPUT READY BIT
81A7 CA3381     JZ          PARIN  ;LOCP TIL IT IS READY
81AA D90D       IN          PINPT  ;READ THE CHARACTER
81AC C9        RET

```

```

;
; READER INPUT. BITS 2 AND 3 OF IOBYT ARE USED TO DIRECT
; CONTROL TO THE PROPER DEVICE. AN ASCII CHAR IS
; RETURNED IN THE A REGISTER.
;

```

```

81AD          RI:
81AD 3A0300     LDA          IOBYT ;GET IO ASSIGNMENT
81E0 1F        RAR          ;MOVE THE BITS IN QUESTION

```



```

;
; PARALLEL OUTPUT ROUTINE. ALPHA CHARACTERS ARE
; OUTPUT AND CONTROL CHARACTERS ARENT. A LINE FEED IS
; USED TO DETERMINE THE END OF THE LINE AND DRIVES A
; RUN OUTPUT. THIS INTERFACE IS INTENDED TO DRIVE
; BASIC LINE PRINTERS.
;

```

```

81EE PAROT:
81EE DECE IN PSTAT ;GET THE PORT STATUS
81EE E6E0 ANI PCRDY ;TEST OUTPUT READY BIT
81EA CAE681 JZ PAROT ;LOCP TIL IT IS REALY
81EC 79 MOV A,C ;THE CHARACTER IS IN C
81EE D30C CUT POTPT ;OUTPUT THE DATA
81F0 C9 RET

```

```

;
; PUNCH OUTPUT ROUTINE. BITS 4 AND 5 OF THE IOBYT
; DETERMINE THE PUNCH DEVICE. A CHARACTER IS EXPCETED
; IN THE C REGISTER
;

```

```

81F1 PG:
81F1 3A0300 LDA IOBYT ;GET IO ASSIGNMENT
81F4 07 RLC ;MOVE BITS 4 AND 5 TO 6 AND 7
81F5 07 RLC ;SO THAT THEY CAN BE MOVED
81F6 C3FC81 JMP LO1 ;TO THE CONSOLE POSITION

```

```

;
; LIST OUTPUT ROUTINE. BITS 6 AND 7 OF THE IOBYT
; DETERMINE THE LIST DEVICE.
;

```

```

81F9 LO:
81F9 3A0300 LDA IOBYT ;GET THE IO ASSIGNMENT
81FC LO1:
81FC 07 RLC ;MOVE BITS 6 AND 7 TO THE
81FD 07 RLC ;CONSOLE POSITION
81FE C3D981 JMP OBRCH ;BRANCH TO PROPER ROUTINE

```

```

;
; CONSOLE INPUT STATUS ROUTINE. THE SELECTED
; CONSOLE DEVICE IS INTERGATED TO DETERMINE IF A
; CHARACTER IS WAITING. A ZERO IS RETURNED IF NO
; CHARACTER AND A -1 IF THERE IS A CHARACTER.
;

```

```

8201 CSTS:
8201 3AC300 LDA IOBYT ;WHAT IS CONSOLE DEVICE?
8204 E603 ANI 3 ;BRANCH ON CONSOLE ASSIGN
8206 C21182 JNZ CSTS2 ;ZERO IS SERIAL, DO IT
8209 DBCB IN URTCT ;GET USART INPUT STATUS
820E CSTS1:
820E E6C2 ANI RXRDY
820D CE RZ ;ZERO IS NO CHAR
820E 3EFF MVI A,0FFH ;NON ZERO IS CHAR, USE -1
8210 C9 RET
8211 CSTS2:

```

```

8211 3D          DCR    A          ;ONE IS KEYBOARD
8212 CAC7F8     JZ      KBSTS
8215 3D          DCR    A          ;TWC IS PARALLEL
8216 C24600     JNZ     USRST

```

```

;
; PARALLEL INPUT STATUS ROUTINE
;

```

```

8219 DBCE       IN      PSTAT    ;GET THE PORT STATUS
821B C3CB82     JMP     CSTSI    ;USE UART TEST AND SET

```

```

;
; CHARACTER INPUT ROUTINE (SERIAL). IF THE MODE IS
; BAUDOT, THE INPUT CHARACTER IS CONVERTED TO ASCII.
; IF THE CHARACTER INPUT IS A CASE SHIFT CHARACTER,
; THAT SHIFT IS DONE AND ANOTHER CHARACTER IS WAITED
; FOR.
;

```

```

821E          CHI1:
821E E5         PUSH   H          ;SAVE SCME REGISTERS
821F D5         PUSH   D
8220 21BE83     LXI    H,BDTAB  ;GET SET TO CONVERT
8223 DECA      IN      URDIA    ;GET THE 5 BIT CHAR
8225 5F         MOV    E,A      ;SAVE IT IN E
8226 3A0700     LDA    BCASI    ;GET THE CASE TO MAKE IT 6
8229 B3         ORA    E        ;APPEND THE CASE (0 OR 20H)
822A 5F         MOV    E,A      ;GET TOTAL IN DE
822B 16C0      MVI    D,0
822C 19         DAD    D        ;CALCULATE THE TABLE ADDRESS
822E 7E         MOV    A,M      ;GET THE ASCII CHAR
822F D1         POP    D        ;RECCVER SCME REGISTERS
8230 E1         POP    H
8231 1 A7      +     TEST   A        ;NEGATIVE INDICATES THAT IT
8232 1 F0      +     ANA    A        ;SET FLAGS, CY=0
8233 F0         RP      ;CASE CHAR, IF NCT, RETURN
8234 E620      ANI    20H      ;ALLCW ONLY 0 OR 20H AS CASE
8235 3207C0     STA    BCASI    ;PUT CASE AWAY FOR LATER

```

```

CHI:

```

```

8238 DB0B      IN      URDCT    ;GET USART STATUS
823A E6C2      ANI    RXRDY    ;IS THERE A CHARACTER?
823C CA3882     JZ      CHI      ;NO, LOOP UNTIL THERE IS
823F DB0B      IN      URDCT    ;GET THE STATUS AGAIN (MODE)
8241 E680      ANI    DSR     ;WHAT IS THE MODE?
8243 CA1E82     JZ      CHI1    ;ZERO IS BAUDOT, CONVERT
8246 DB0A      IN      URDIA    ;NOT BAUDOT, SIMPLY INPUT
8248 E67F      ANI    7FH     ;BUI MAKE IT 7 BITS FIRST
824A C9         RET

```

```

;
; SERIAL CHARACTER OUTPUT ROUTINE. THE MODE IS
; CHECKED AND IF IT IS BAUDOT, THE CHARACTER IN THE
; C REGISTER IS CONVERTED TO BAUDOT PRIOR TO SENDING.
; OTHERWISE, IT IS SIMPLY OUTPUT TO THE USART. IF
; A CASE CHANGE IS INDICATED, THE CASE CHARACTER IS

```



```

; TRANSMITTED PRIOR TO THE ACTUAL CHARACTER (BAUDOT
; MODE ONLY).
;
824E      CHO:
824E      D80B      IN      URTCT      ; DETERMINE MODE FIRST
824C      E66C      ANI      DSR        ; 0=BAUDOT
824F      CA5D82    JZ      CHC2      ; BAUDOT MODE, GO CONVERT IT
8252      CHO1:
8252      D8CB      IN      URTCT      ; TEST FOR READY AND OUTPUT
8254      E6C1      ANI      TXRDY     ; THE CHARACTER IN THE C REG
8256      CA52E2    JZ      CH01
8259      79        MOV      A,C
825A      D30A      OUT      URTDA
825C      C5        RET
825D      CHO2:
825D      E5        PUSH     H          ; SAVE SOME REGISTERS
825E      D5        PUSH     D
825F      21EE83    LXI      H,BDTAB ; SEARCH THE TABLE FOR A MATCH
8262      1640      MVI      D,64    ; THE TABLE IS 64 CHARS LONG
8264      79        MOV      A,C          ; COMPARISONS ARE DONE IN A
8265      CHO3:
8265      BE        CMP      M          ; DOES THIS ONE MATCH?
8266      CA7082    JZ      CHC4      ; YES, EXIT LOOP
8269      23        INX      H          ; BUMP ADDRESS
826A      15        DCR      C          ; DECREASE COUNT
826E      C26582    JNZ     CH03      ; CONTINUE TO LOOP TILL ZERO
826E      1640      MVI      D,64    ; IF NO MATCH, PRETEND BLANK
8270      CHO4:
8270      3E40      MVI      A,64    ; ACTUAL VALUE IS 64-COUNT
8272      92        SUB      D          ; CALCULATE ACTUAL VALUE
8273      57        MOV      D,A      ; SAVE A COPY IN D FOR LATER
8274      21CE00    LXI      H,BCASO ; TEST AND SET CASE IF APPROP
8277      E620      ANI      20H     ; 6TH BIT IS CASE (0=LETTERS)
8279      0E        CMP      M          ; COMPARE TO PREVIOUS CASE
827A      CA8982    JZ      CH06      ; MATCH, NO NEED TO SEND CASE
827C      77        MOV      M,A      ; NO MATCH, SET CASE RIGHT
827E      0E1F      MVI      C,LTRS ; SEND CASE (0=LETTERS)
;
; CASE 0 OR 20H?
8280      1 A7      ANA      A          ; SET FLAGS, CY=0
8281      CA8682    JZ      CHC5      ; LETTERS CASE, SEND IT
8284      0E1B      MVI      C,FIGS ; NOT LETTERS, SEND FIGURES
8286      CHO5:
8286      CD5282    CALL     CHC1      ; SEND THE CASE SHIFT CHAR
8289      CHO6:
8289      4A        MOV      C,D      ; RECOVER THE ORIGINAL CHAR
828A      D1        POP      D          ; RECOVER SOME REGISTERS
828E      E1        POP      H
828C      C35282    JMP      CH01      ; GO SEND THE CHARACTER

```

```

;
; IO CHECK ROUTINE. THE VALUE OF THE IOBYT IS
; RETURNED IN THE A REGISTER

```

```

;
828F      ; IOCHK:
828F      3A0300      LDA      IOBYT
8292      CS          RET

```

```

; IO SET ROUTINE. THE VALUE OF THE C REGISTER IS
; SUBSTITUTED FOR THE VALUE OF THE IOBYT.

```

```

8293      ; IOSET:
8293      79          MOV      A,C      ;SET NEW IOBYT
8294      320300      STA      IOBYT
8297      CS          RET

```

```

; MEMORY CHECK ROUTINE. TWO RAM LOCATIONS ARE USED
; TO STORE THE TOP OF RAM. THEY ARE RETURNED IN
; A AND B. LEAST SIGNIFICANT IN A.

```

```

829E      ; MEMCK:
829E      3A0600      LDA      MSZ2      ;MOST SIGNIFICANT BYTE HERE
829F      47          MOV      E,A      ;PUT IT IN E
829C      3A0500      LDA      MSZ1      ;LEAST SIGNIFICANT BYTE HERE
829F      CS          RET

```

```

; PUNCH BYTE OUTPUT ROUTINE. THE CONSOLE BYTE OUTPUT
; ROUTINE IS USED EXCEPT THAT THE IOBYT IS FIRST
; ROTATED SUCH THAT THE CONSOLE ASSIGNMENT AND PUNCH
; ASSIGNMENT ARE INTERCHANGED. WHEN FINISHED, THEY
; ARE AGAIN INTERCHANGED SO THAT ALL IS OK.

```

```

82AC      ; BYTOP:
82A0      CDA682      CALL     PUCO      ;INTERCHANGE PUNCH AND CONSOLE
82A3      CD7283      CALL     BYTOT     ;DO BYTE OUTPUT

```

```

; ROUTINE TO INTERCHANGE CONSOLE AND PUNCH ASSIGN.

```

```

82A6      ; PUCO:
82A6      F5          PUSH     PSW      ;SAVE SCME REGISTERS
82A7      E5          PUSH     H
82A8      210300      LXI     H,IOBYT ;POINT HL AT IOBYTE
82AE      7E          MOV      A,M      ;INTERCHANGE TOP AND BOTTOM
82AC      0F          RRC
82AD      0F          RRC
82AE      0F          RRC
82AF      0F          RRC
82EC      77          MOV      M,A      ;PUT IT BACK
82B1      E1          POP      H      ;RECCVER SOME REGISTERS
82B2      F1          POP      PSW
82B3      CS          RET

```

```

; WORD PUNCH OUTPUT ROUTINE. PUNCH AND CONSOLE
; ASSIGNMENTS ARE INTERCHANGED TO USE CONSOLE

```

; ROUTINES.

```

82B4      WRDOP:
82B4      CDA682      CALL      PUCO
82B7      CDCA81      CALL      WRDOT
82EA      C3A682      JMP       PUCO

```

```

; PUNCH MESSAGE TYPER. AGAIN, CONSOLE AND PUNCH
; ASSIGNMENTS ARE INTERCHANGED TO USE CONSOLE
; ROUTINES.

```

```

82BC      TYPMP:
82BD      CDA682      CALL      PUCO
82C0      CD9A83      CALL      TYPMG
82C3      C3A682      JMP       PUCO

```

```

; READER BYTE INPUT ROUTINE. THE READER ASSIGNMENT
; IT ROTATED INTO THE CONSOLE POSITION SO THAT THE
; CONSOLE ROUTINES CAN BE USED. THE ASSIGNMENTS ARE
; RESTORED WHEN DONE. SINCE AN EXIT CAN BE DONE FROM
; THE EIT ROUTINE, A FLAG IS SET TO INDICATE THAT THE
; SWITCH HAS BEEN MADE. IF AN EXIT IS TAKEN THE
; PROPER ASSIGNMENT WILL BE RESTORED.

```

```

82C6      BYTIR:
82C6      CDE482      CALL      RDRCO      ;MOVE READER DEVICE TO CONSOLE
82C9      CD4E83      CALL      BYTIN      ;USE CONSOLE BYTE INPUT

```

```

; ROUTINE TO RECOVER THE OLD CONSOLE ASSIGNMENT AND
; OLD READER ASSIGNMENT AFTER USING CONSOLE ROUTINES.

```

```

82CC      RCVR1:
82CC      F5          PUSH     PSW      ;SAVE SOME REGISTERS
82CD      E5          PUSH     H
82CE      21C300     LXI     H,IOBYT ;PCINT HL AT IOBYTE
82C1      7E          MOV     A,M      ;ROTATE IOBYT 2 LEFT
82D2      07          RLC
82D3      07          RLC
82D4      77          MOV     M,A      ;RETURN IOBYTE
82D5      2C          INR     L      ;PCINT HL AT IFLAG
82D6      3600      MVI     M,0      ;RESET IT TO ZERO
82D8      E1          POP     H      ;RECCVER SOME REGISTERS
82D9      F1          POP     PSW
82DA      C9          RET

```

```

; READER WORD INPUT ROUTINE. AGAIN, THE CONSOLE
; ROUTINE IS UTILIZED BY MOVING THE IOBYTE.

```

```

82DB      WRDIR:
82DE      CDE482      CALL      RDRCO
82DE      CD2683      CALL      WRDIN

```

```

82E1 C3CC82 JMP RCVR1
;
; READER TO CONSOLE ASSIGNMENT SWITCHER.
;
82E4 RDRCO:
82E4 E5 PUSH H ;SAVE HL
82E5 21C300 LXI H,IOBYT ;POINT HL AT IOBYTE
82E8 7E MOV A,M ;SHIFT IOBYT 2 LEFT
82E9 0F RRC
82EA 0F RRC
82EB 77 MOV M,A ;RESTORE IT
82EC 2C INR L ;POINT AT IFLAG
82ED 36FF MVI M,-1 ;SET FLAG TO -1
82EF E1 POP H ;RECOVER HL
82F0 C9 RET
;
; ECHC INPUT AND TEST
;
82F1 EIT:
82F1 CDB581 CALL ECHO ;GET CHAR AND ECHO
82F4 FE20 CPI ' '
82F6 CAF182 JZ EIT ;IGNCRE BLANKS
82F9 FE2C CPI ',' ;COMMA IS A DELIMITER
82FE 37 STC
82FC CE RZ ;RETURN WITH CY SET FOR DELIM
82FD FECD CPI CR ;CR IS A DELIMITER
82FF 37 STC
8300 C8 RZ ;RET WITH CY SET FOR DELIM
8301 FE47 CPI 'G' ;STCF IF LARGER THAN F
8303 3F CMC ;INVERT CARRY BIT
8304 DC RNC ;NO CARRY IS OK CHARACTER
8305 3AC400 LDA IFLAG
;
;
;
8308 1 A7 + TEST
8309 1 A7 + ANA A ;SET FLAGS , CY=0
8309 CA7380 JZ MAIN
830C CDCC82 CALL RCVR1
830F C37380 JMP MAIN
;
; NIBBLE ROUTINE, CONVERY ASCII TO HEX
;
8312 NIBBL:
8312 D641 SUI 'A' ;COMPARING FOR >=10
8314 F21983 JP GTA
8317 C6C7 ADI 7 ;ADJLST FOR GAP BETWEEN 9 & A
8319 GTA:
8319 C6CA ADI 10 ;MAKE IT BINARY
831E C9 RET ;THATS ALL FOR PERFECT INPUT
;
; EXPRESSICN (PARAMETER LIST) GRABBER
831C EXPR:

```

```

831C CD2683 CALL WRDIN ;READ 16 BIT GROUP INTO HL
831F E3 XTHL ;EUBBLE PC DOWN ON STACK
8320 E5 PUSH H ;STACK IS PARM LIST
8321 05 DCR B ;DEC PARM COUNT
8322 C21C83 JNZ EXPR ;QUIT WHEN B EMPTY
8325 C9 RET

```

```

;
; WRDIN - READ IN 16 BIT ADDRESS
;

```

```

8326 WRDIN: LXI H,0 ;CLEAR BUFFER
8326 210000 PUSH B ;SAVE PARM COUNT
8329 C5 MVI B,4 ;GET 4 HEX DIGITS IN ASCII
832A 06C4 WRD1: CALL EIT ;READ/CHECK CHARACTER
832C CDF182 JNC WRD2 ;CARRY SET IMPLIES DELIMITER
832C D23B83 MOV A,B ;TEST FOR NEW WORD
8332 78 CPI 4 ;B=4 MEANS JUST STARTED
8333 FE04 JZ WRD1 ;IGNRE . AT BEGINNING
8335 CA2C83 JMP EXIT ;VALID DELIMITER IF NOT 1ST
8338 C34883 WRD2: CALL NIBBL ;EAT SOME CHARS
833E CD1283 DAD H ;HL*2
833E 29 DAD H ;HL*2
833F 29 DAD H ;HL*2
8340 29 DAD H ;LONG LEFT SHIFT 4 KIDDIES
8341 29 CRA L ;BRING IN NEW 4 BITS
8342 B5 MOV L,A ;HL NOW HAS GOOD DATA
8343 6F DCR B ;REDUCE CHAR CCUNT
8344 05 JNZ WRD1 ;3,2,1,0 AND OUT
8345 C22C83 EXIT: MOV A,D ;UPDATE CHECKSUM
8348 7A ADD L ;CRDER IMMATERIAL
8348 85 ADD H ;CHK SUM IN A CORRECT
8349 84 MOV D,A ;CHECKSUM UPDATE COMPLETE
834A 57 POP B ;RESTORE
834C C1 RET
834D C9

```

```

;
; BYTE INPUT RIN
;

```

```

834E BYTIN: PUSH B ;SAVE IT
834E C5 MVI B,0 ;INIT BINARY VALUE BUFFER
834F 06C0 CALL EIT ;READ NEW DIGIT
8351 CDF182 JC EXITB ;LEAVE IF DELIMITER, A=B=0
8354 DA6C83 CALL NIBBL ;GET BINARY VALUE IN A
8357 CD1283 MOV B,A ;SAVE BINARY VALUE IN B
835A 47 CALL EIT ;READ 2ND (LOW) DIGIT, IF ANY
835E CCF182 JC EXITB ;QUIT IF DELIMITER, A=B=DIGIT
835E DA6C83 CALL NIBBL ;CONVERT 2ND DIGIT TO BINARY
8361 CD1283

```

```

8364 4F      MOV      C,A      ;SAVE CHAR
8365 78      MOV      A,B      ;GET FIRST CHAR
8366 07      RLC          ;MOVE TC TOP CHAR
8367 07      RLC
8368 07      RLC
8369 07      RLC
836A B1      CRA      C      ;OR IN FIRST CHAR
836E 47      MOV      B,A      ;SAVE IN B
836C          EXITB:
836C 78      MOV      A,B      ;IN CASE A HOLDS DELIM CHAR
836D 82      ADD      D          ;ADD CHECKSUM
836E 57      MOV      D,A      ;RETURN UPDATED CKSUM
836F 78      MOV      A,B      ;RESTORE $A FROM $B
8370 C1      POP      B          ;RESTORE B
8371 C9      RET

```

```

;RET DONE ELSEWHERE

```

```

;
; BYTE OUTPUT RTN
;
; OUTPUTS 2 HEX ASCII CODED DIGITS
;
; TERMINATES PRINT FIELD WITH SPACE
;

```

```

8372          BYTOT:
8372 C5      PUSH     B          ;CALLS USE $B
8373 CD78B3  CALL     BTOT1      ;DECODE 2 HEX DIGITS
8376 CDD481  CALL     SPACO      ;PRINT A SPACE TRAILER
8379 C1      POP      B          ;RESTORE
837A C9      RET

```

```

;
; BASIC BYTE OUTPUT DECODER RTN
;
; PERFORMS SUMCHECK ADDITION ON OUTPUT
;
; HIGH CRDER DIGIT OUTPUT FIRST
;
; LCW CRDER HEX DIGIT OUTPUT SECOND
;

```

```

837B          BTOT1:
837E 47      MOV      B,A      ;SAVE BYTE IN B
837C 82      ADD      D          ;ADD TO CHECKSUM
837D 57      MOV      D,A      ;SAVE NEW SUM
837E 78      MOV      A,B      ;RETRIEVE BYTE
837F 07      RLC          ;SWAP HEX DIGITS
8380 07      RLC
8381 07      RLC
8382 07      RLC
8383 CD8783  CALL     HXOUT      ;PRIMED W/HIGH DIGIT
8386 78      MOV      A,B      ;INSTEAD OF SHIFT, MOV
;FALL THRU TO HXOUT

```

```

;
; HEX OUTPUT RTN
;
; OUTPUTS ASCII CODED HEX DIGIT
;
; FROM LOW 4 BITS OF A REGISTER
;

```

```

8387          HXOUT:          ;C-9,A-F NOT CONTIG-

```

```

8387 E6CF ANI 0FH ;UOUS CODES, 2 CASES
8389 D6CA SUI 0AH ;STRIP HIGH DIGIT
838B F29483 JP HXC1 ;>9 TEST
838E C63A ADI '9'+1 ;BRANCH...GE...10
;ADD ASCII 9 +1 FOR
;ASCII CHAR CODE OF #
8390 4F MOV C,A ;PRIME ARGUMENT FOR CALL
8391 C3C681 JMP CO ;OUTPUT IT AND RET
8394 HX01: ;LETTER CASE
8394 C641 ADI 'A' ;ADD ASCII 'A' FOR
;ASCII CODE OF DIGIT
8396 4F MOV C,A ;PRIME ARGUMENT FOR CALL
8397 C3C681 JMP CO ;OUTPUT AND RET
;NO ERRCR ON INPUT
;POSSIBLE, SO NO CHKS

```

```

;
; TYPE MESSAGE ROUTINE
; (HL) POINTS TO MESSAGE ADDRESS
; MESSAGE TERMINATED BY NEGATIVE BYTE
;

```

```

839A TYPMG:
839A E5 PUSH H ;PRINT PREFACE MSSG
839E 21AD83 LXI H,CRMG ;POINT TO PREFACE
839E CDA283 CALL TYP1 ;CR,LF,...
83A1 E1 POP H ;POINT TO 1ST MSSG AGN
;FALL THRU AND PRINT
83A2 TYP1:
83A2 7E MOV A,M ;FETCH CHAR FROM MEM
1 + TEST ;CHK IT 1ST,
83A3 1 A7 + ANA A ;SET FLAGS , CY=0
83A4 F8 RM ;NEGATIVE IMPLIES END
83A5 4F MOV C,A ;PRIME WITH ARG
83A6 CDC681 CALL CO ;PUT IT OUT
83A9 23 INX H ;POINT TO NEXT CHAR
83AA C3A283 JMP TYP1 ;LOOP TILL NEG CHAR
83AD 0DCA0000 CRMG: DB CR,LF,0,0,0,0,-1
83B1 03C0FF
83B4 3AFF CRCO: DB '.,,-1
83B6 2EFF PMIMG: DB '.,,-1
83B8 48884954 RSMG: DB 'EXIT ',-1
83BC 2CFF
83BE 00450A41 EDTAB: DB 0,'E',LF,'A SIU',CR,'DRJNFCK'
83C2 20E34955
83C6 0D44524A
83CA 4E46434B
83CE 545A4C57 DB 'TZLWHYPGCBG',0A0H,'MXV',80H
83D2 48895051
83D6 4F4247A0
83DA 4D885680
83DE 0C330A20 DB 0,'3',LF,'- *87',CR,'$4'',',21H,':('

```

83E2 2C2A3937
83E6 0D243427
83EA 2C213A28
83EE 352B2932
83F2 23263031
83F6 393F26A0
83FA 2E2F3DB0

DB 15+)2#6015?E!,0A0H,!,/=!,80H

END

NO PROGRAM ERRORS

SYMBOL TABLE

* 01

A	0007	B	0000	BCASI	0007	BCASO	000E
PCTAE	E3BE	BEGIN	802B	BG0	802E	BG2	8064
ETGT1	E37B	BYTIN	834E	BYTIR	E2C6	BYTOP	82A0
EYTOT	E372	C	0001	CHI	E238	CHI1	821E
CHF	E24E	CHO1	8252	CHO2	825D	CHO3	2265
CHCA	E270	CHO5	8286	CHO6	8289	CI	8193
CC	E1D6	CR	000D	CRCO	83B4	CRMG	E3AD
CSTS	E201	CSTS1	820B	CSTS2	8211	D	0002
DISP1	809F	DISP2	80BB	DISP3	80D6	DISP5	80B9
DSFCK	F80C	DSPT	F804	DSR	0080	E	0003
ECHF	E1E5	ECHO1	81C8	ECHCM	000D	EIT	82F1
EXIT	E348	EXITB	836C	EXPR	831C	FIGS	001B
GC	E0EE	GTA	8319	H	0004	HXC1	E394
FXCLT	E387	IBRCH	8196	IFLAG	0004	INSRT	80ED
INST1	80F0	INST2	810E	IOBYT	0003	ICCHK	828F
ICSET	8293	KBIN	F801	KBSTS	F807	L	0005
LDC	E113	LD1	812E	LF	000A	LO	81F9
LC1	E1FC	LOAD	810F	LTRS	001F	M	0006
MAIN	E073	MEMCK	8298	MONST	002C	MSZ1	0005
MSZ2	0006	NIBL	8312	OBCH	81D9	PARCT	000F
PARIN	E1A3	PARMD	00A6	PARUT	81E6	PCSAV	002E
PINFT	000D	PIRDY	0002	PMTMG	83B6	PO	81F1
PCRDY	008C	POTPT	000C	PSSAV	0032	PSTAT	000E
PS*	0006	PUCO	82A6	RCVR1	82CC	RRCO	82E4
RETR1	E17A	RETRN	816F	RGSAV	0038	KI	81AD
RSMG	E3BE	RSTRT	8145	RXRDY	0002	SDPIO	F80A
SP	0006	SPACO	8104	TEST	03E0	TXRDY	0001
TXRXE	0027	TYPI	83A2	TYPMG	839A	TYPMP	82BD
UARTI	008E	UARTR	0055	URTBM	0083	URTCT	000B
URTD	000A	URTMO	00FA	USRIN	0040	USROT	0043
USFST	0046	WRD1	832C	WRD2	833E	WRDIN	8326
WRCIF	E2DB	WRDOP	82B4	WRDOT	81CA		

* 02

* 03

* 04

* 05

* 06

* 07

* C

8. CIRCUIT BOARD LAYOUT AND SCHEMATIC DIAGRAMS

The following pages contain complete diagrams of the HAL Communications Corp. Model MCEM-8080 Microcomputer System. These diagrams reflect the current circuit connections as of the printing date of this manual. HAL Communications reserves the right to make changes in the circuitry without incurring any obligation to make such changes in previously sold units. The diagrams may not be duplicated in any form without the express permission of HAL Communications Corp.

MCEM-8080
Addendum No. 2
June, 1976

Use of the UPB cable connections:

The MCEM-8080 is furnished with a two foot length of 40 conductor ribbon cable with a connector on one end that mates to the UPB (Universal Processor Bus) connector of the MCEM. The cable connections are explained in Table 2.6 on page 2-12 of the MCEM manual (the red stripe on the cable corresponds to pin 1). When plugging the cable into the MCEM, be sure to align it correctly as indicated by the small arrows embossed on the plastic connectors. If your cable connector has NOT been polarized, it should be by putting a small piece of bare wire into the cable connector pin 32 location. This should correspond to the missing pin of the MCEM UPB connector.

Connection to the cable can be made by simply separating the conductors of the ribbon cable, stripping and tinning each one required, and then connecting the wires as required. The cable SHOULD NOT BE EXTENDED BEYOND THE TWO FOOT LENGTH FURNISHED. Alternately, the same 3M connector can be attached to the ribbon cable. The cable connector is a 3M part no. 3417-0000, which can be obtained from a 3M distributor or from HAL Communications Corp. for \$6.00. The mating circuit board connector is a 3M no. 3432-1002 connector at \$4.00 each from HAL. The cable connector is designed so that it clamps directly over the cable and several can therefore be attached to the same cable. The following procedure should be used to attach the 3M connector to the cable.

Installation of additional connectors to the UPB cable:

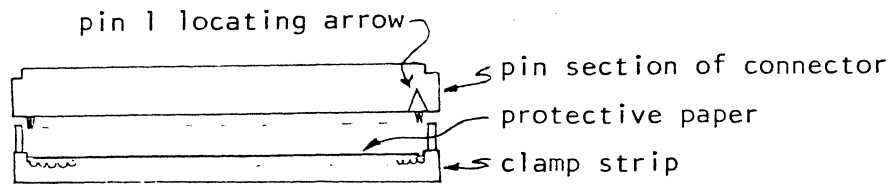
The 40 conductor cable polarity is distinguished in two ways:

1. The RED stripe side corresponds to pin 1 of the UPB connector
2. The ribbing of the cable is heavier on one side than the other. This can best be determined by looking at the END of the cable, although dragging of your finger-nail across the two sides will also indicate which is roughest and therefore has the heavier ribbing.

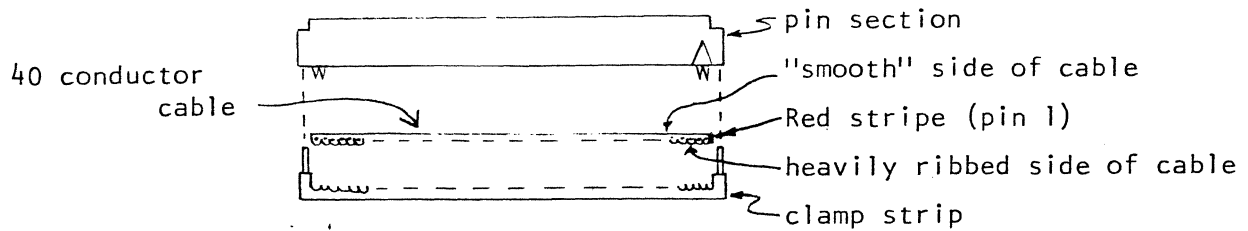
Refer to the attached Figure 2 for the following instructions.

The 40 pin connector (3M No. 3417-0000) is a two-piece assembly, the larger section with the connector pins and a smaller clamp strip. The clamp strip has a protective paper covering over an adhesive. In assembly, the protective strip is removed to expose the adhesive and the cable is "sandwiched" between the forks of the connector pins and the clamp strip. Attach the connector to the cable using the following procedure:

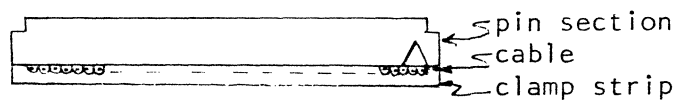
1. Locate and mark the desired connector location on the 40 conductor cable. Note that several connectors can be placed on the same cable since the connectors simply clamp around the cable, allowing it to pass through. Therefore, if several connectors are intended, do not cut the cable until the end-most connector has been installed.
2. Carefully remove the protective paper BUT NOT the adhesive from the clamp piece.
3. Put the adhesive surface of the clamp piece on the the heavily ribbed side of the ribbon cable, taking care to remove only the paper and not the adhesive with it.
4. Locate the embossed arrow on the connector pin section. This indicates the location of pin 1 of the connector.
5. Place the pin section of the connector on the opposite side of the cable from the clamp assembly, aligning the arrow with the red stripe on the cable.
6. Align the guide pins of the clamp piece into the mating holes of the pin section and press the two pieces together with your fingers until the forks of the connector pins start to "bite" into the cable.
7. RECHECK THE CABLE AND CONNECTOR ALIGNMENT TO BE SURE THAT:
 - a. The red stripe of the cable is adjacent to the arrow
 - b. The heavily ribbed side of the cable is against the clamp.
 - c. The connector is perpendicular to the cable.
8. After checking, the two sections can be completely pressed together in a bench vice. Use only enough pressure to close the gaps between the cable - too much pressure will break the connector. If the vise has rough surfaced jaws, you may wish to prevent scratching of the connector by using cardboard protective shims.
9. If additional connectors are required, they can be attached at any cable location using the above procedure. If it is desired to end the cable after the connector, use a VERY SHARP knife or razor-blade and cut the cable off flush with the outside edge of the connector. Be careful to cut on the "scrap" end of the connector and not on the processor end! After cutting, inspect the cut edge to be sure that adjacent wires have not been shorted together in the process of cutting.
10. The completed connector should now be keyed by inserting a short piece of No. 22 bare wire into connector pin no. 32 position. Notice that the numbers are marked on the face of the connector - odd numbers down the arrow side (starting at the arrow) and even numbers on the other side.



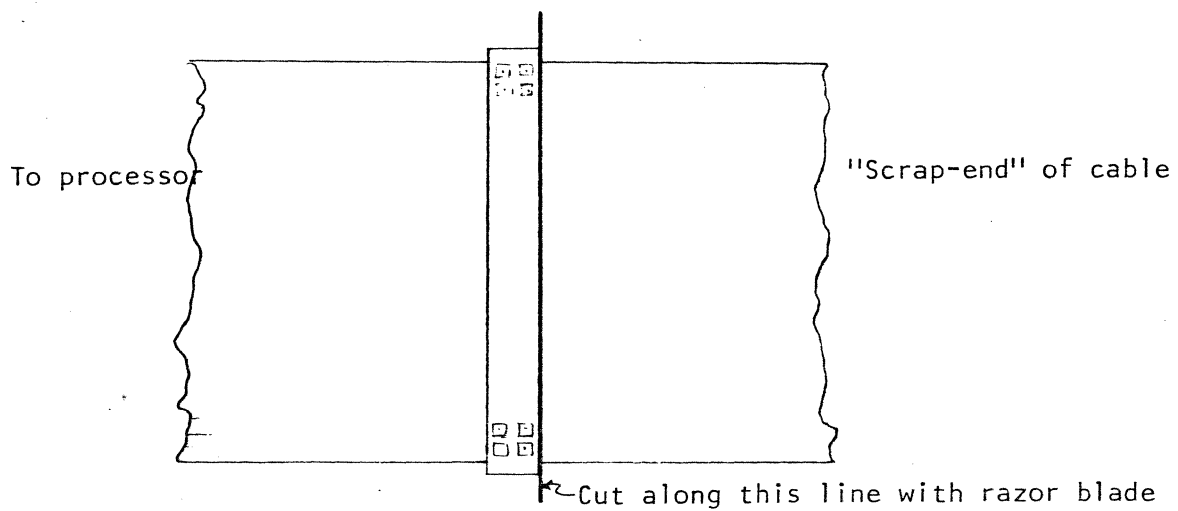
a. 3M 3417-0000 Cable Connector



b. Assembly of connector



c. Assembled connector



d. Cutting-off excess cable

Figure 2. Preparation of UPB Cable Connector