MOTOROLA Semiconductor Products Inc.

ASSEMBLY INSTRUCTIONS

MOTOROLA M6800 MICROCOMPUTER SYSTEMS DESIGN EVALUATION KIT

Benchmark Family
For Microcomputer Systems

ASSEMBLY INSTRUCTIONS FOR THE MOTOROLA M6800 DESIGN EVALUATION KIT

Benchmark Family For Microcomputer Systems

First U.S. Printing

Motorola's M6800 Microcomputer Systems Design Evaluation Kit contains devices which — with the peripherals of your choice — will allow you to build a complete microcomputer system which meets your design goals at minimum cost in the shortest time.

This Kit is composed of the following NMOS components:

Qty.	Function	Reference
1	MPU	MC6800L
2	PIA	MC6820L
1	ACIA	MC6850L
2†	RAM	MCM6810L-1
1	ROM	MCM6830L-7

The MEK Board (MEC6800C) is specifically developed for assembly of the Kit parts into a microcomputer system.

In order to help the systems designer evaluate a basic, working M6800 microcomputer system, a simplified evaluation kit was designed for easy construction. A complete interconnection schematic is also provided to assist in assembly and help give the designer a functional understanding of basic system integration and potential expansion capabilities.

†Space is provided on the card for four additional MCM6810L-1 RAMs if needed.

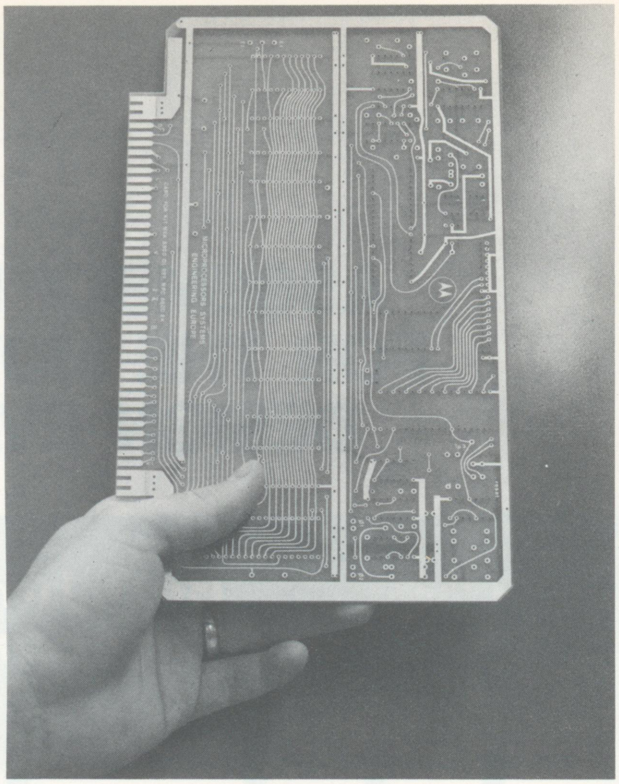


Photo 1a. MEK Board for MEK6800D1 Kit

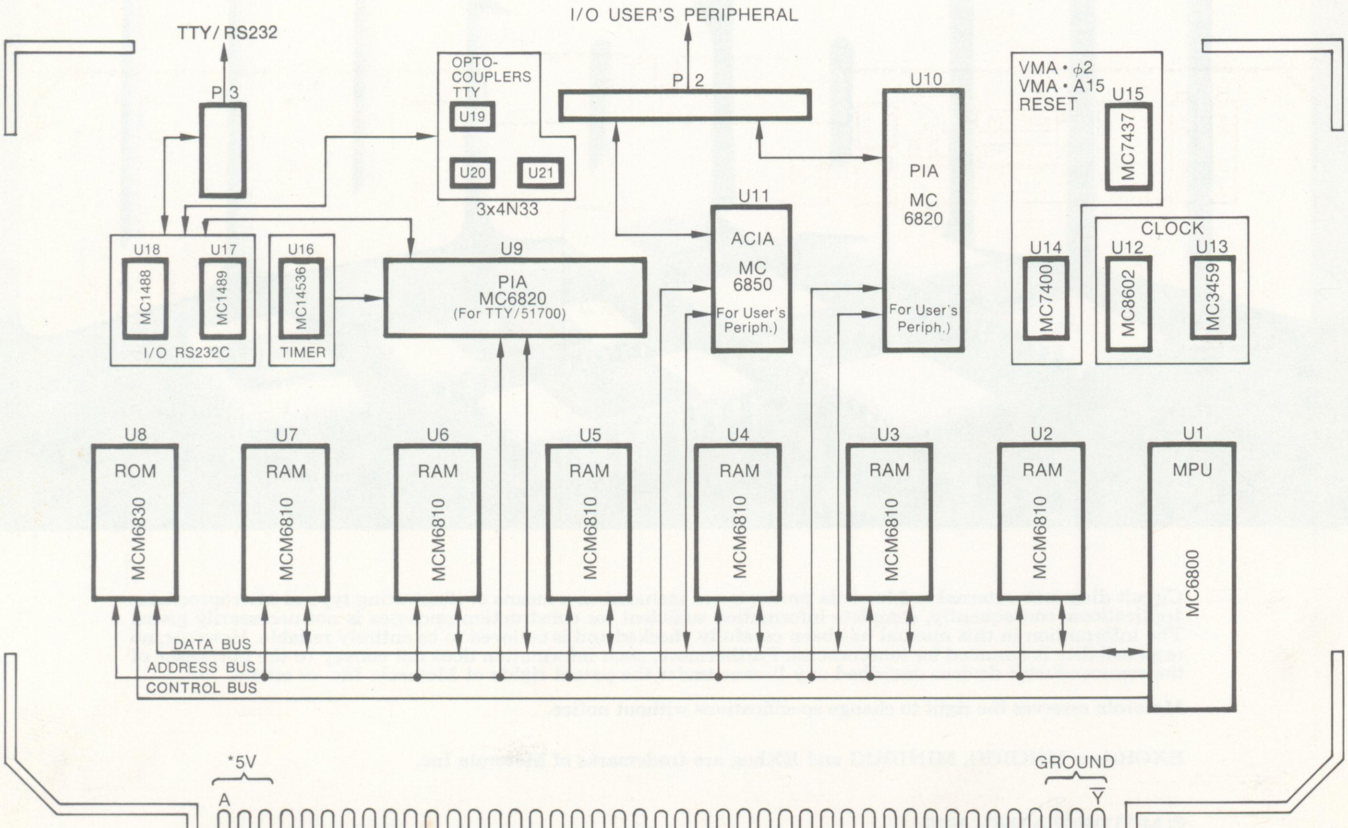


Figure 1a — Basic Semiconductor Positioning Diagram — PC Board Evaluation Kit MEK6800D1

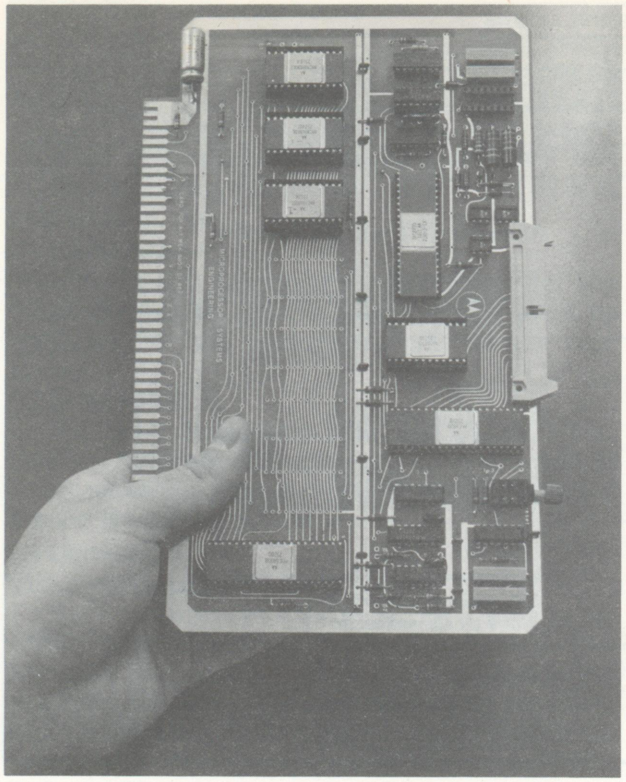


Photo 1b. MEK Board with Mounted Components

1 SYSTEMS OVERVIEW

When assembled, the MEK Board provides a microcomputer system contained on a single printed circuit board. The purpose of this system is to facilitate user evaluation of the individual Microcomputer Family parts: MPU, PIA, ACIA, RAM and ROM.

This Motorola microcomputer systems design kit also provides the user with a totally functional system, obviating any need to debug the hardware before bringing up the system the first time.

It demonstrates the minimum system concept and illustrates the ease of use and flexibility of the M6800 Family.

2 MAIN FEATURES OF MEK6800D1 KIT USING THE MEK BOARD

- Single PC Board. (MEC6800C).
- RS232 or TTY interface with 10 cps or 30 cps operation.
- Reset Switch.
- MIKBUG loader and diagnostic control program.
- PIA and ACIA provided for user peripheral device interface.
- May be adapted for Expanded Memory and Resident Software.

Photos 1a, 1b and Figures 1a, 1b show parts positioning

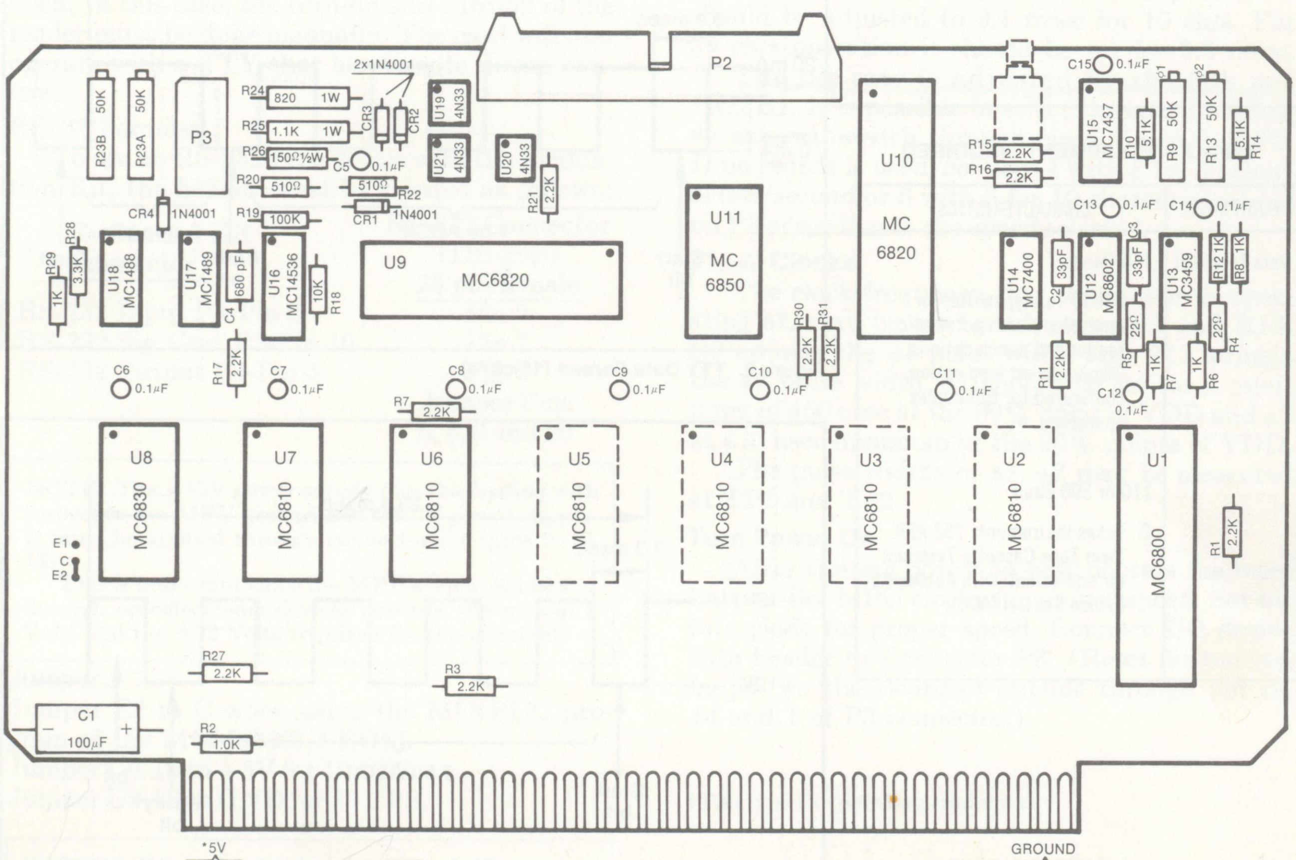


Figure 1b — Complete Parts Positioning Diagram

TABLE 1 — KIT SPECIFICATIONS

CHARACTERISTICS	SPECIFICATIONS
Power Requirements:	+ 5 VDC ± 5% at 1A typ +12 VDC ±10% at 100 mA typ −12 VDC ±10% at 50 mA typ
Clock Frequency:	100 KHz to 1 MHz (adjustable)
Signal Characteristics: Connector (P1)	
Address bus	Three-state TTL voltage compatible
Data bus	
Input	TTL voltage compatible
Output	Three-state TTL voltage compatible
Input and output commands	TTL voltage compatible
MC6820 Peripheral Interface Adapter (P2)	
Data Signals	
PAO-PA7 input/output lines	TTL voltage compatible
PBO-PB7 input/output lines	Three-state TTL voltages compatible
Control Signals	
CA1, CA2 and CB1	TTL voltage compatible
CB2	Three-state TTL voltages compatible
M6850 ACIA Adapter (P2)	
Transmit data	Up to 500K bps transmission rate
Receive data	TTL voltage compatible 8 or 9 bit transmission
Terminal Interface Specifications (P3)	
Data transfer rate	110 or 300 Baud
Signal characteristics	TTY or RS-232C compatible
Reader control signal	Control signal for modified TTY devices
Data format	Standard ASCII code format

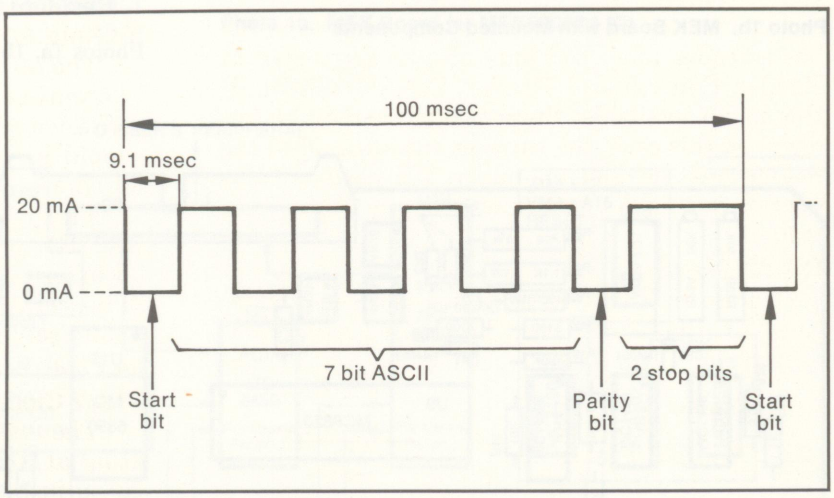


Figure 2. TTY Data Format (10 ch/s)

Table 2 — EQUIPMENT REQUIRED

EQUIPMENT	CHARACTERISTICS
Data Terminal	110 Baud 1. ASR33 Teletypewriter (or equivalent) with automatic reader and punch options, 20mA current loop option, configured for full-duplex operation. 110 or 300 Baud 2. Texas Instruments 733 ASR Twin Tape Cassette Terminal (or equivalent) with Automatic Device Control (ADC).

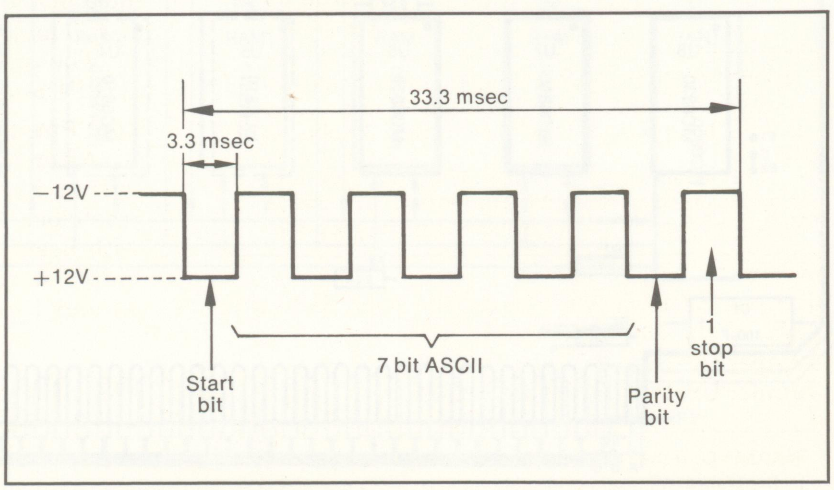


Figure 3. RS-232C Format (30 ch/s)

3 USER'S INFORMATION

The complete system requires three power supplies: +5V, −12V, +12V. The +5V is for all logic on the card. The ±12V is for the RS-232 and the Teletype† interface. The +5V supply requires approximately 1 Amp. The +12V and −12V require approximately 100 mA and 50 mA, respectively.

P1 Connector

This is the connector at the bottom of the MEK Board. To use the Board, power must be connected to the appropriate pins on the connector.

P2 Connector

The PIA at location 8008 and the ACIA at location 8010 are connected to this connector. It is not necessary that this connector be connected for the Board to operate.

P3 Connector

Teletype Terminal:

This connector interfaces the Teletype or RS232 type terminal to the Evaluation Kit. The Kit must be connected with a cable for interfacing to a Teletype (TTY), 20 ma current loop interface. The Serial Output and Serial Input should be connected to the appropriate points on the TTY. Refer to Figure 4.

The return line is TTY Com. If the terminal has a reader relay this may also be connected to Reader Control and Reader Control return. If this is not available, these two wires may be left open. In this case, the turn-on and turn-off of the reader must be done manually. The card will also operate with a TTY that has remote device control.

RS232 Terminal:

To use an RS-232 type device with the Evaluation Kit, the header must be rewired as follows:

Evaluation Kit Connector	RS-232 Connector (DB-25S) 25 pin female
RS-232 Input P3-Pin 2	Pin 2
RS-232 Sig. Gnd. P3-Pin 10	Pin 7
RS-232 Output P3-Pin 3	Pin 3
	Jumper Pins 5, 6, 8 and 20

NOTE: The ±12V power supply must be floating with respect to the MPU ground and the 5V power supply. It must be applied through connector P3 (pins 9, 10, 12).

This is important since the MPU system is isolated through optoelectronic devices between the system 5 Volts and the ±12 Volts required for the terminal.

Jumper:

Jumper E2 to C when using the MIKBUG program of the MCM6830L7 ROM.
Jumper U9-16 to +5V for 10 CPS.
Jumper U9-16 to GND for 30 CPS.

WARNING: If the +5V and Gnd are connected backwards to the Board, all semiconductor devices will be destroyed.

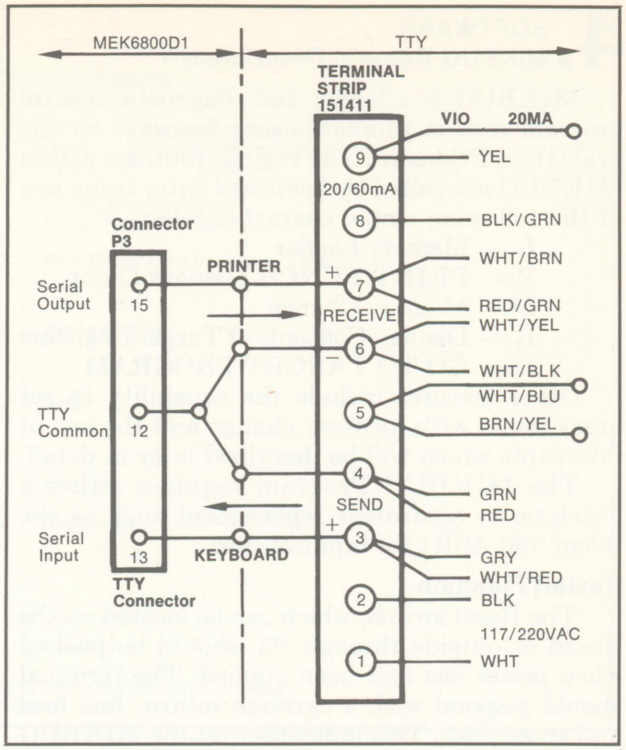


Figure 4. TTY 20ma Full Duplex Wiring Diagram

Bit Rate Adjust

The bit rate must be set up for TTY operation (Potentiometer R23A). For 10 ch/s operation the zero level of the clock measured at test point TP4 should be adjusted to 9.1 msec for 10 ch/s. For 30 ch/s operation it should be set for 3.3 msec.

The bit rate is adjusted by the 50k pot (R23B). It is possible to select the speed putting an external switch through pin 4, 6 and 8 of P3. If no switch is used, connect 4 with 8 for 30 characters/second or 6 with 8 for 10 character/second on P3 according to the speed selected.

$\phi 1$, $\phi 2$ Clocks

The clock frequency that the system is operating at, may be set by adjusting R9 and R13. R9 adjusts the $\phi 1$ pulse width and R13 adjusts the $\phi 2$ pulse width. $\phi 1$ should be set for a minimum of 450 nsec at the 90% point of VDD and $\phi 2$ at 470 nsec minimum at the 90% points of VDD.

The pulse widths of $\phi 1$, $\phi 2$ may be measured at TP1 and TP2.

Turn Power On

After turning power on first depress the reset button. Set MPU clocks for proper speed. Set bit rate clock for proper speed. Connect I/O device with header to Connector P3. (Reset button can be put on the Board or outside through pin 16, 14 and 1 of P3 connector.)

†Teletype Corporation trademark.

4. SOFTWARE
MIKBUG General Description††

MIKBUG is a loader and diagnostic control program that is supplied using firmware on the evaluation PC board. The various routines within MIKBUG are called by keyboard entry using one of the following single character calls:

L — Memory Loader
P — PRINT/PUNCH Memory Dump
M— Memory Change
R — Display Contents of Target Registers
G — GO TO TARGET PROGRAM

Other features include the capability to set breakpoints with memory change and the use of interrupts which will be described later in detail.

The MIKBUG Program requires either a Teletype or equivalent replacement such as the Silent 700 ASR† for input/output.

Restart Function

The Reset switch, which can be located on the Board or outside through P3, should be pushed when power has first been applied. The terminal should respond with a carriage return, line feed and an asterisk. This indicates that the MIKBUG control program is now ready for input. Should control be lost during the execution of a user's program, control may be regained by pushing the reset.

Memory Loader

(To load Type L on the terminal)

The Memory Load function (L) of the MIKBUG loads formatted object tape information in the MEK6800D1 system. The tape format can be generated by the assembler or the punch dump function (P) of MIKBUG.

The paper tape format is shown in Figure 5a.

When loading from paper tape, all characters prior to the Start-of-Record character are ignored. Each record has a checksum which is verified as the data associated with the record is stored to memory. If the checksum is in error, loading is stopped, and a question mark is printed. The operator now has the options to: abort the load, repositioning and rereading the record (entering an L), or continuing the load (entering an L), ignoring the checksum error. The continue option is not recommended since if the error is in the address field the data may destroy data in another area of memory.

Example of Load: (underline indicates user Input)

```
*L
S11300000202020202020286AA02023F86FC881CCC7B
S1040010B635
S9
*
```

Here a short tape has been loaded using the load (L) command. The "S9" has been entered from the keyboard to exit the loader.

†Texas Instruments Inc. trademark.
††See Motorola Engineering Note 100

Print/Punch Memory Dump

(To print or punch Type P on the terminal)

The PRINT/PUNCH Memory Dump (P) function of MIKBUG punches an absolute formatted object tape. The formatted object tape format is shown in Figure 5b. The beginning and ending address must be entered using memory change. The beginning address is to be entered at location A002 and A003 and the ending address at locations A004 and A005. All address and data will be in hexadecimal. The beginning address is the address of the first byte to be punched and the ending address is the address of the last byte to be punched.

PRINT/PUNCH Example:

(Underline indicates user input)

```
M A002
*A002 FF 00
*A003 E8 00
*A004 EA 00
*A005 OF 10
*A006 A0 (SP)(CR)
*P
S11300000202020202038EAA02023F86FC881CCC7B
S1040010B635
```

Memory Change

(To open a memory location Type M on the terminal and type the memory location in hexadecimal)

The memory is changed by using the following three step sequence: 1. Opening the location which displays the contents, 2. changing the location, and 3. closing the location. An open location is one whose contents MIKBUG has printed on the teletype for examination, and whose contents are available for change. A closed location is one whose contents are no longer available for change. Any 8 bit byte may be opened. The contents of an open location may be changed by typing a space followed by two hex characters (new contents). MIKBUG will then open the next location. To end the sequence, and close a location without any change, a space should be typed followed by a Carriage Return (CR). Once a location has been opened, the next location may be opened by typing any character except a space. To open a location, respond to the asterisk with an "M." MIKBUG will then type a space. Next, type the hex address of the location to be opened; it must be 4 hex characters in length.

```
*M 1000
1000 A5 FC
1001
```

In the example above, location 1000 now contains FC and location 1001 is now open.

NOTE: Tapes punched by MIKBUG do not have the End-of-File Record therefore an "S9" must be typed on the keyboard to exit the loader.

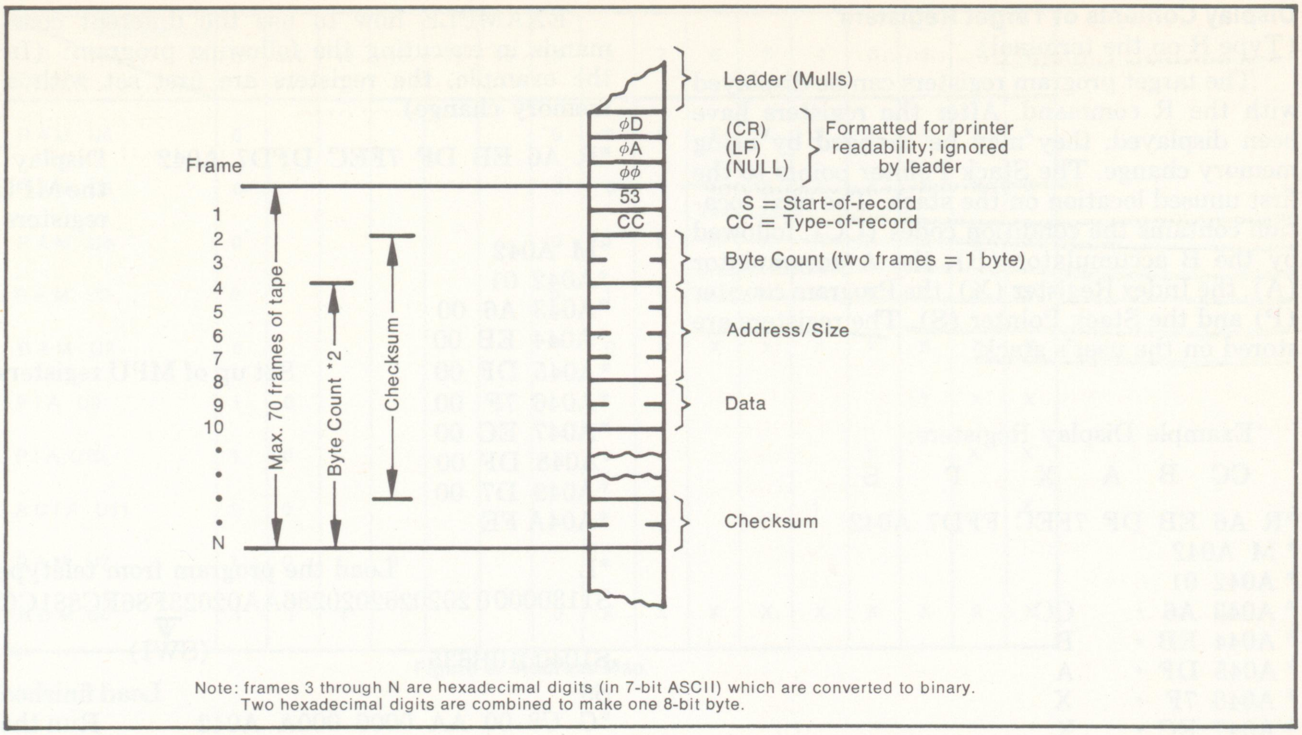


Figure 5a. Paper Tape Format

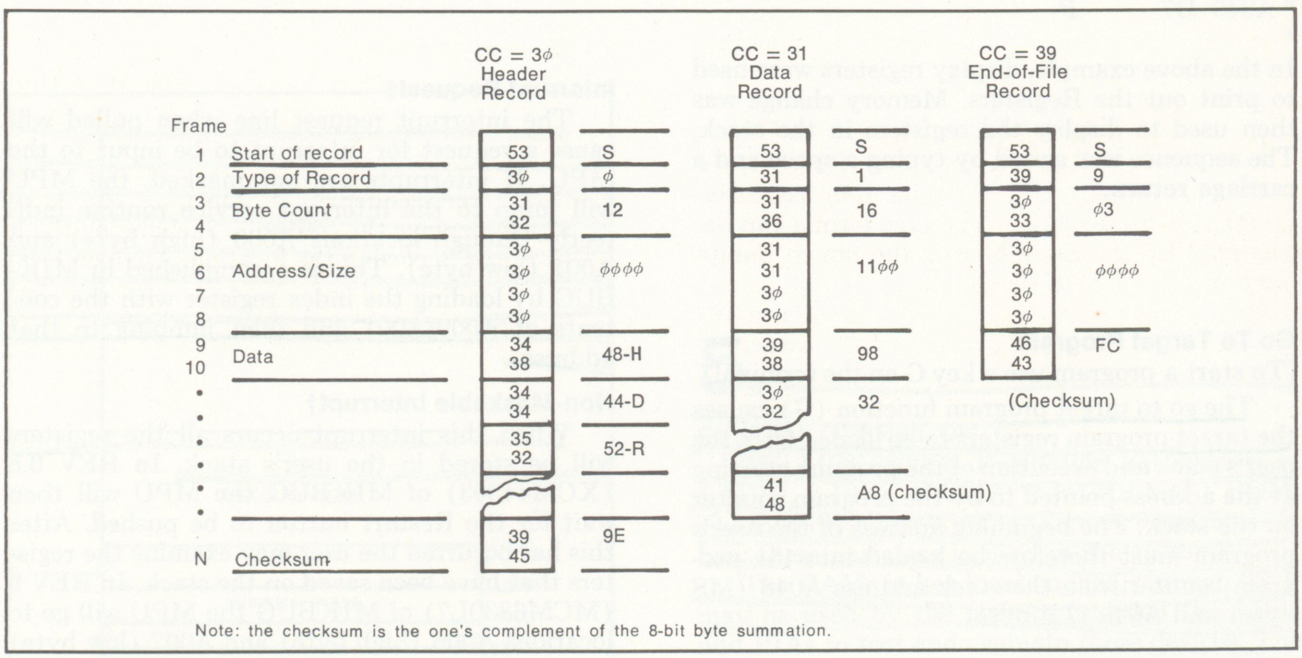


Figure 5b. Object Tape Format

Example: Memory change

```
*M 0000
*0000 02 •
*0001 02 •
*0002 02 •
*0003 02 •
*0004 02 •
*0005 02 •
*0006 86 •
*
```

In the example, location 0000 was opened followed by 0001-0005. No location or data was changed. If memory change does not cause the contents of a register to change, a question mark will be printed.

Display Contents of Target Registers

(Type R on the terminal)

The target program registers can be displayed with the R command. After the registers have been displayed, they may be changed by using memory change. The Stack Pointer points to the first unused location on the stack. The next location contains the condition codes (CC), followed by the B accumulator (B), the A accumulator (A), the Index Register (X), the Program counter (P) and the Stack Pointer (S). The registers are stored on the user's stack.

Example Display Registers:

```
   CC  B   A   X     P     S
*R A6 EB  DF  7FEC  FFD7  A042
* M A042
* A042 01
* A043 A6 •    CC
* A044 EB •    B
* A045 DF •    A
* A046 7F •    X
* A047 EC •    X
* A048 FF •    P
* A049 D7      P
*
```

In the above example, display registers were used to print out the Registers. Memory change was then used to display the registers in the stack. The sequence was ended by typing a space and a carriage return.

Go To Target Program

(To start a program use a key G on the terminal)

The go to target program function (G) causes the target program registers to be loaded from the user's stack and execution of the program starting at the address pointed to by the program counter on the stack. The beginning address of the user's program must therefore be loaded into the program counter from the stack address A048 (MS byte) and A049 (LS byte).

The target program will free run until:

A. A wait (WAI) instruction is encountered where the program will wait for a non-maskable interrupt or an interrupt request.

B. A Software interrupt (SWI) is encountered at which time control will jump back into MIKBUG. The user's register will be stored on the user's stack and they will also be displayed by MIKBUG. Therefore, SWI along with memory change may be used to set breakpoints.

C. The target program blows (recovery is by Restart push-button).

EXAMPLE how to use the different commands in executing the following program: (In the example, the registers are first set with a memory change)

```
*R A6 EB DF 7FEC DFD7 A042          Display
                                    the MPU
                                    registers
*M A042
*A042 01
*A043 A6 00
*A044 EB 00
*A045 DF 00                         Set up of MPU registers
*A046 7F 00
*A047 EC 00
*A048 DF 00
*A049 D7 00
*A04A FE
*L                  Load the program from teletype
S11300000 2020202020286AA02023F86FC881CCC7B
                              (SWI)
S1040010B635
S9                                  Load finished
*G C8 00 AA 0000 000A A042          Run the
*                                   program
```

Interrupt Request†

The interrupt request line when pulled will cause a request for interrupt to be input to the MPU. If interrupts are not masked, the MPU will jump to the interrupt service routine indirectly through locations A000 (high byte) and A001 (low byte). This is accomplished in MIKBUG by loading the index register with the contents of A000/A001 and then jumping to that address.

Non-Maskable Interrupt†

When this interrupt occurs all the registers will be stored in the user's stack. In REV 0.8 (XCM44303) of MIKBUG the MPU will then wait for the Restart button to be pushed. After this has occurred the user may examine the registers that have been saved on the stack. In REV 9 (MCM6830L7) of MIKBUG the MPU will go to locations A006 (high byte) and A007 (low byte) for the address of the NMI service routine.

Allowable Characters

At anytime during the execution of MIKBUG, only hex characters and control characters (P, M, L, G, R, a space, and a Carriage Return) are allowed. If an invalid character is input, it will be echoed back to the terminal and the current function will be terminated.

†See Motorola Engineering Note 100

Address line A / DEVICE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAM U6	0						0	0	0	X	X	X	X	X	X	X
RAM U5	0						0	0	1	X	X	X	X	X	X	X
RAM U4	0						0	1	0	X	X	X	X	X	X	X
RAM U3	0						0	1	1	X	X	X	X	X	X	X
RAM U2	0						1	0	0	X	X	X	X	X	X	X
PIA U9	1	0												1	X	X
PIA U10	1	0											1		X	X
ACIA U11	1	0										1				X
RAM U7	1	0	1	0												
ROM U8	1	1	1				0	X	X	X	X	X	X	X	X	X

Figure 6. Address Map

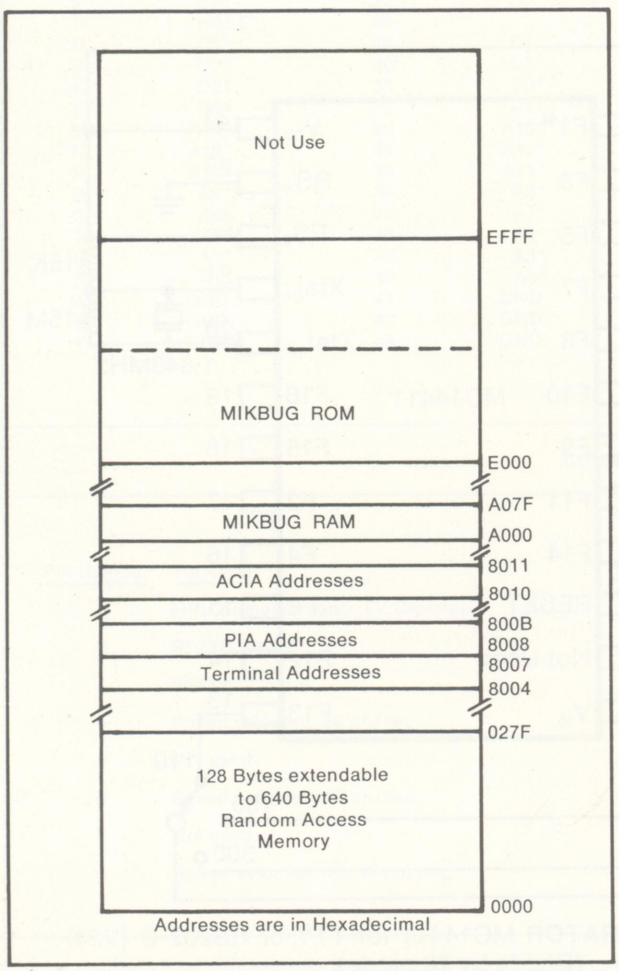


Figure 7. Evaluation Kit Memory Map

5.

SYSTEM OPERATION

The MEK6800D1 Evaluation Kit consists of 1 MPU, 2 PIAs, 1 ACIA, 2 RAMS and 1 ROM. The memory address maps (Figures 6 and 7) show how these parts are located in the system.

The RAMs in hex location 0000 through 027F may be used by the user to program the MPU and PIAs to test and evaluate these devices. The RAM at location A000 is used by the MIKBUG control program for temporary storage and stack area.

The ROM is located at location E000 and it is assumed that there are no devices in the system at a higher address.

The PIAs are located at 8004 and 8008. The PIA at location 8004 is used by the MIKBUG control program to interface the TTY and RS232 device. It is also used with the Programmable Timer (MC14536) for the serial interface timing. The PIA at location 8008 is available for use by the user for evaluation of PIA through the connector P2.

The ACIA is located at address 8010 and is available to the user's peripheral device through connector P2. To provide the proper timing for serial data transmission or reception an external baud rate generator must be connected through connector P2 to pin 4 and 3 of the ACIA. The MC14411 bit rate generator can be used with a crystal of 1.843 MHz to obtain precise bit rates from 110-9600 bits/sec. Refer to Figure 8.

The interconnecting schematic provides for both serial and parallel data transmission with additional user's peripherals; this can be arranged using 1 ACIA and 1 PIA device. The second PIA is used to interface with the TTY or the TI733, providing access to the MIKBUG program.

The MINIBUG program, also contained in the ROM MCM6830L7 and described in Motorola Engineering Note 100, may be used in conjunction with the ACIA to interface with the TTY instead of the MIKBUG program,using the PIA to interface with the TTY. However, use of the MINIBUG program with the Evaluation Kit is not recommended for the following reasons:

- the MINIBUG program does not offer the possibility of PRINT/PUNCH;
- the RAM used in conjunction with the MINIBUG must be located at FF00-FFFF, which means the implementation of a special address decoder to select this RAM. If a TTL circuit is used for this decoder, care must be taken to not overload the address bus;
- to use the Restart vectoring address (FFFF and FFFE) of the MPU, a special hardware address converter (or ROM) must be used to load the Program Counter with the address of the first instruction (FED6).

To use MINIBUG, or some of its subroutines with this Evaluation Kit without using special decoders, the MINIBUG program must be relocated in RAMs U2 through U6 and the scratch pad RAM must be transferred from FF00 to A000.

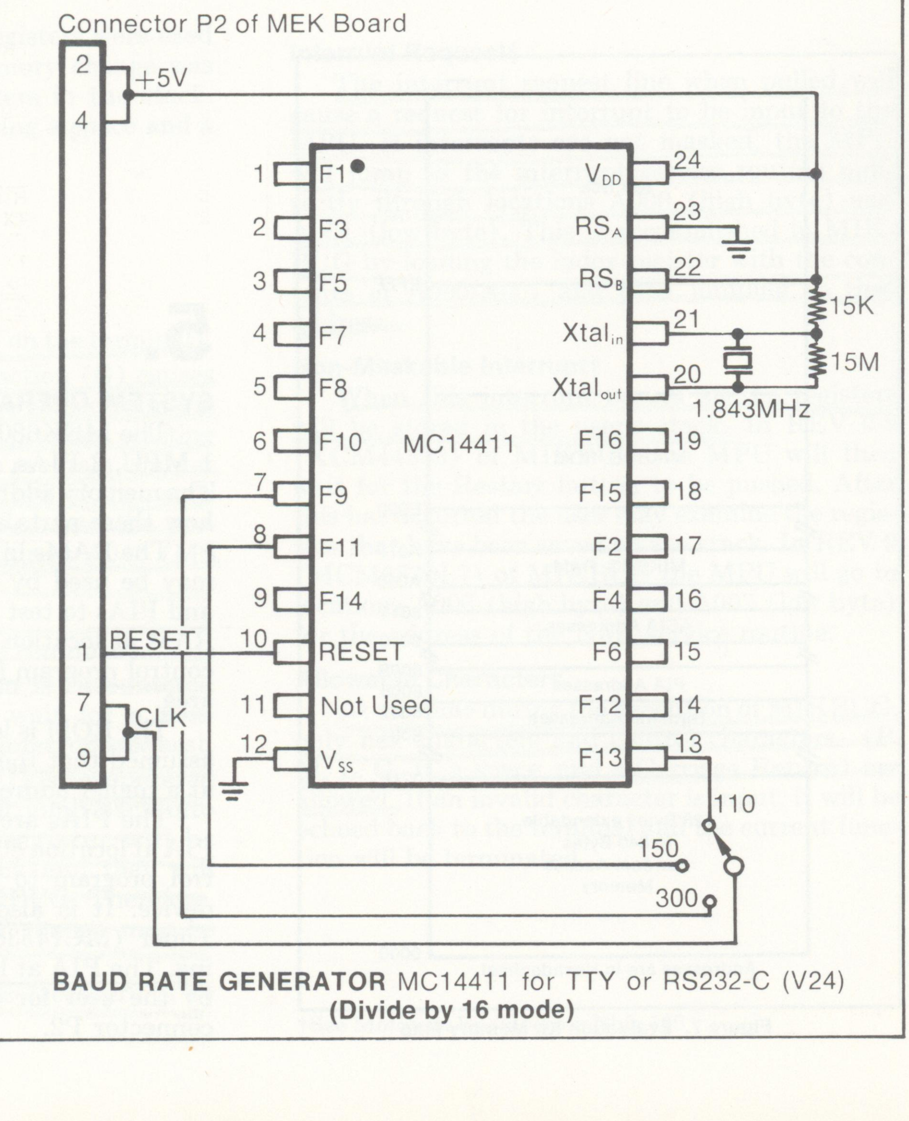


Figure 8. Use of the MOTOROLA I/C MC14411 for external generation of baud rate when using the ACIA.

BAUD RATE GENERATOR MC14411 for TTY or RS232-C (V24)
(Divide by 16 mode)

EVALUATION KIT CONNECTOR PIN ASSIGNMENTS

CONNECTOR P1

Component Side Pin Number	Function	Back side of the board	Function
A	+5VDC	1	+5VDC
B	+5VDC	2	+5VDC
C	+5VDC	3	+5VDC
D	$\overline{IRQ}$	4	
E	$\overline{NMI}$	5	$\overline{RESET}$
F	VMA	6	R/W
H	GND	7	$\phi 1$
J	$\phi 2$	8	GND
K	GND	9	GND
L		10	VUA**
M	−12VDC	11	−12VDC
		12	
P		13	
R		14	
S		15	
T	+12VDC	16	+12VDC
U		17	
V		18	TSC
W		19	
X		20	
Y		21	
Z		22	
A		23	
B		24	
C		25	
D		26	
E	BA	27	
F	$\overline{HALT}$	28	
H	D3*	29	D1*
J	D7*	30	D5*
K	D2*	31	D0*
L	D6*	32	D4*
M	A14*	33	A15*
N	A13*	34	A12*
P	A10*	35	A11*
R	A9*	36	A8*
S	A6*	37	A7*
T	A5*	38	A4*
U	A2*	39	A3*
V	A1*	40	A0*
W	GND	41	GND
X	GND	42	GND
Y	GND	43	GND

CONNECTOR P2

Pin Number	Function		
8, 10, 12, 14, 16, 6, 17, 18, 39, 40,	GND		
2, 4,	+5V		
29	PA7	A Peripheral Data Bus	PIA
30	PA6	A Peripheral Data Bus	PIA
31	PA5	A Peripheral Data Bus	PIA
32	PA4	A Peripheral Data Bus	PIA
33	PA3	A Peripheral Data Bus	PIA
34	PA2	A Peripheral Data Bus	PIA
35	PA1	A Peripheral Data Bus	PIA
36	PA0	A Peripheral Data Bus	PIA
38	CA1	A Peripheral Control	PIA
37	CA2	A Peripheral Control	PIA
21	PB7	B Peripheral Data Bus	PIA
22	PB6	B Peripheral Data Bus	PIA
23	PB5	B Peripheral Data Bus	PIA
24	PB4	B Peripheral Data Bus	PIA
25	PB3	B Peripheral Data Bus	PIA
26	PB2	B Peripheral Data Bus	PIA
27	PB1	B Peripheral Data Bus	PIA
28	PB0	B Peripheral Data Bus	PIA
20	CB1	B Peripheral Control	PIA
19	CB2	B Peripheral Control	PIA
5	$\overline{RTS}$		ACIA
3	TX DATA		ACIA
11	RX DATA		ACIA
7	TX CLOCK		ACIA
9	RX CLOCK		ACIA
15	$\overline{CTS}$		ACIA
13	$\overline{DCD}$		ACIA

CONNECTOR P3

Pin Number	Function	Pin Number	Function
1	MPU Signal Ground (5V Ground)	9	+12 VDC
2	RS232C Input	10	RS232C common, Reader Control Return OV (±12V)
3	RS232C Output	11	Reader Control
4	Speed selected: 30 ch/sec.	12	−12VDC, used as TTY common
5	Not used	13	TTY Serial In
6	Speed selected: 10 ch/sec.	14	External Reset Switch; N.O. contacts
7	Not used	15	TTY Serial Out
8	Speed selected: 10/30 ch/sec.	16	External Reset Switch; N.C. contacts

*Note: the Data Bus and Address Bus are directly connected to the P1 connector.
**VUA is used to interface with external memory from EXORciser.

OPTION A

MEMORY EXPANSION

The MEK Board memory may be expanded using either the MEX6812-1 2K static RAM module or the MEX6815-1 8K dynamic RAM module. When adding memory external to the Board, two precautions must be remembered. One, the Board has no buffering and the user must be careful not to exceed the MC6800 drive capability (130 pf and one TTL load). Two, MPU addressing external to the Board must not conflict with on Board addresses. This contention problem may be avoided by removing U2-U6 from the Board and constraining external memory between 0000_{16} and $7FFF_{16}$.

The MEK Board is compatible with the EXORciser mother board and is therefore easily adaptable to the 2K and 8K memory boards available with the EXORciser. Following is a description of memory expansion requirements using the MEX6812-1 2Kx8 static RAM and the MEX-6815-1 8Kx8 dynamic RAM memory boards.

Memory Expansion with MEX6812-1

The MEX6812-1 is a static memory board having the same functional pin out as the MEK Board. No alterations are required to the MEK Board.

MEMORY EXPANSION WITH MEX6815-1

The MEK PC Board must be modified to allow periodic refreshing of the 8K Dynamic Memory Module. This may be accomplished by stretching the $\phi 1$ and $\phi 2$ clocks into the MPU in order to place it in a wait state (see Chapter 4 of the M6800 Applications Manual for additional detail).

Figure 9 shows the required interface and associated timing. The effect of a $\overline{REF}$ $\overline{REQ}$ from the memory module will be to stretch the MPU's $\phi 1$ and $\phi 2$ clocks while allowing the BUS $\phi 2$ (Pin J) to continue running. The REF GRANT signal allows the memory to do a refresh cycle. Note that an MC7479 flip-flop and an MC7400 Nand Gate must be added to the MEK PC Board. Pin J on the MEK PC Board must be jumpered to Pin L in order to provide a memory clock for the memory module.

Figure 10 shows a portion of the MEK PC Board circuit schematic diagram highlighting this modification for operation with MEX6815-1.

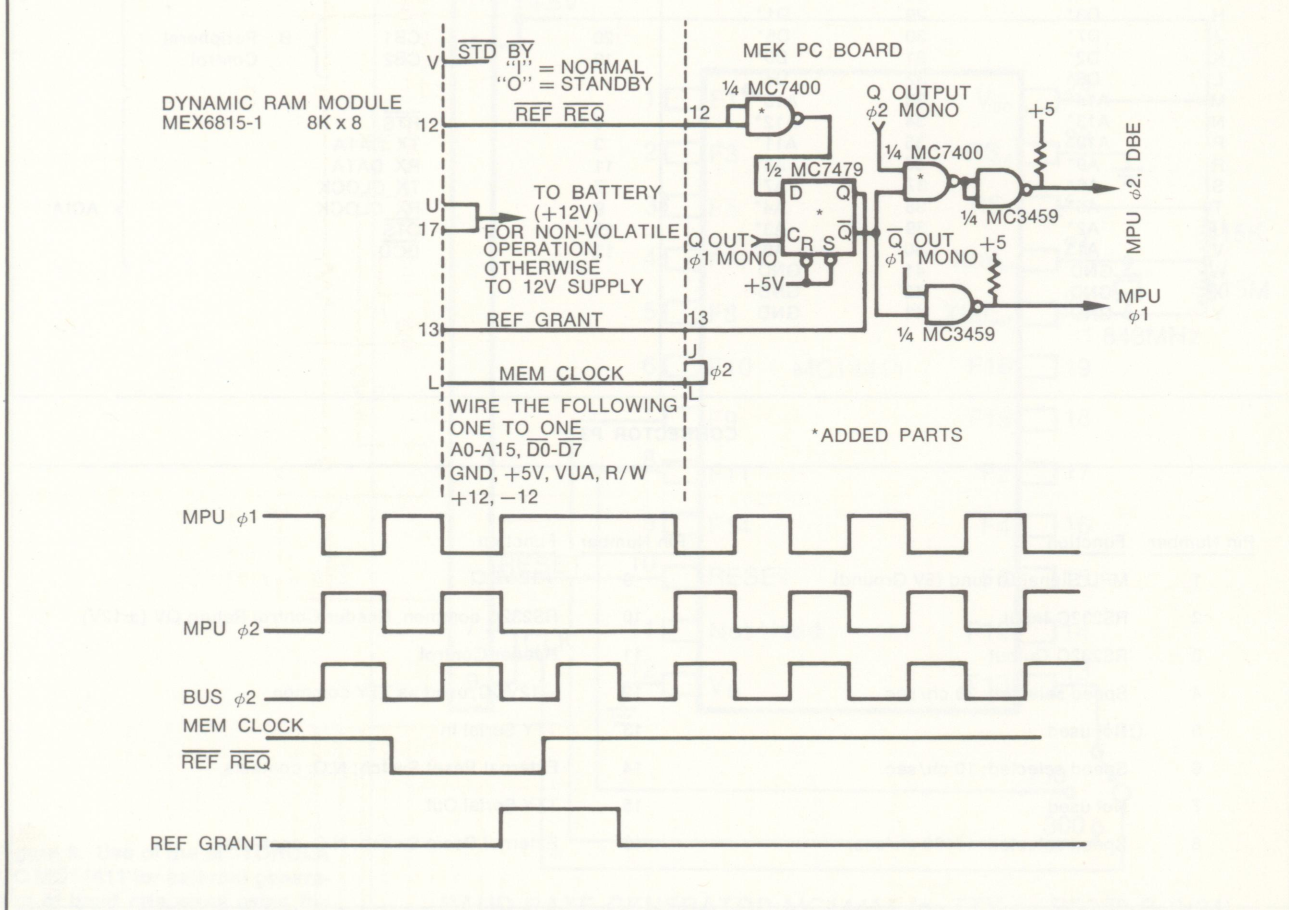


Figure 9. Interface of MEK PC Board to 8K x 8 Dynamic Ram

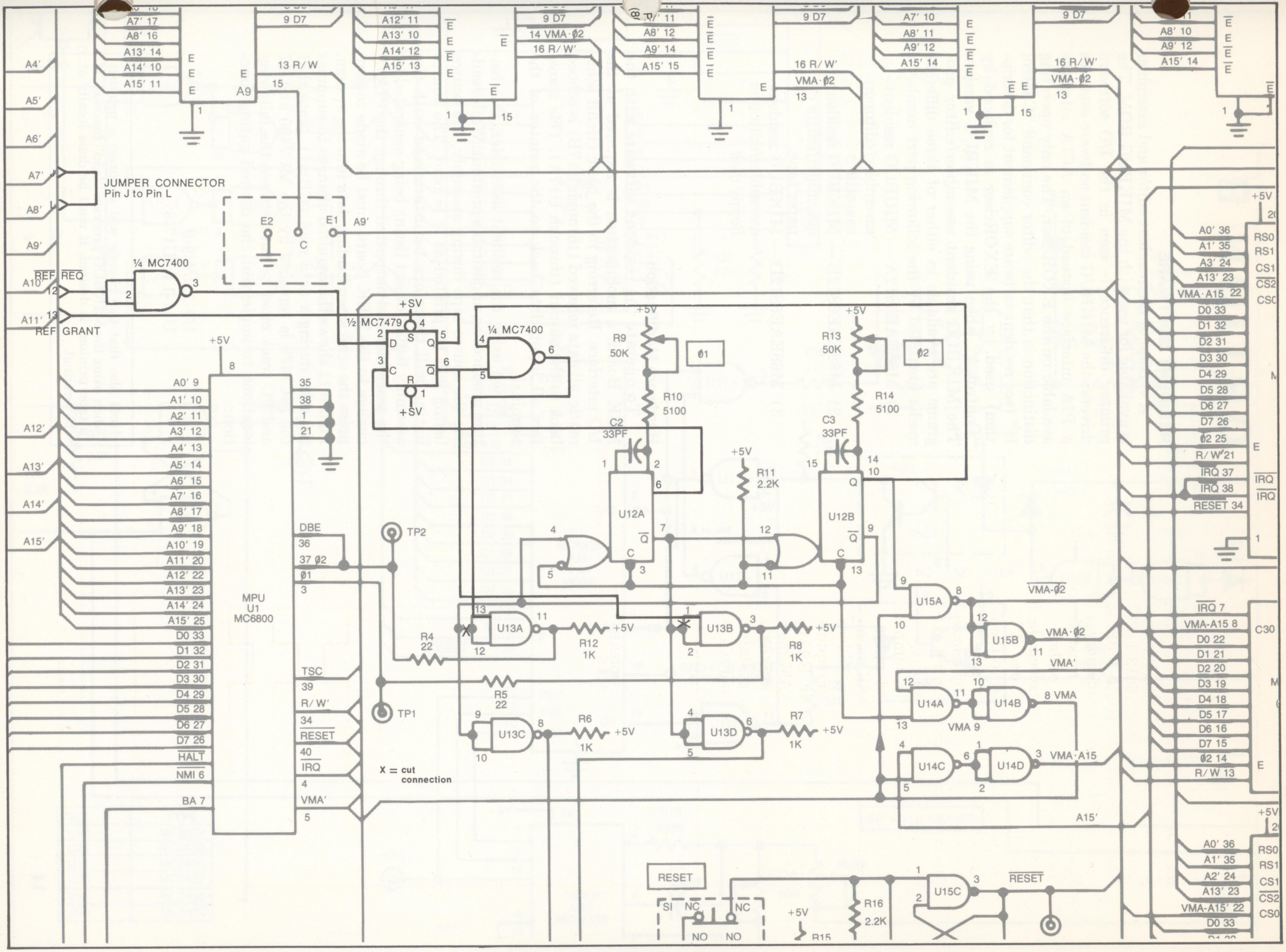


Figure 10. Modifications Required

OPTION B

RESIDENT SOFTWARE

A version of the resident editor and assembler is available for use with the MIKBUG ROM. The primary difference is seen in the I/O software drivers: the MIKBUG resident software assumes a PIA interface instead of an ACIA interface available on the EXORciser. The only operating distinction is that the editor command "Control H" (see resident software supplement for description) used by the EXORciser is replaced by "Control A" when using the MIKBUG version. The MIKBUG resident assembler/editor programs are available in either of three different media identified by the following part numbers:

1) M68EAE6812A — MIGBUG resident assembler/editor on cassette
2) M68EAE6812B — MIKBUG resident assembler/editor on paper tape.
3) M68EAE6812D — MIKBUG resident assembler/editor on floppy disk

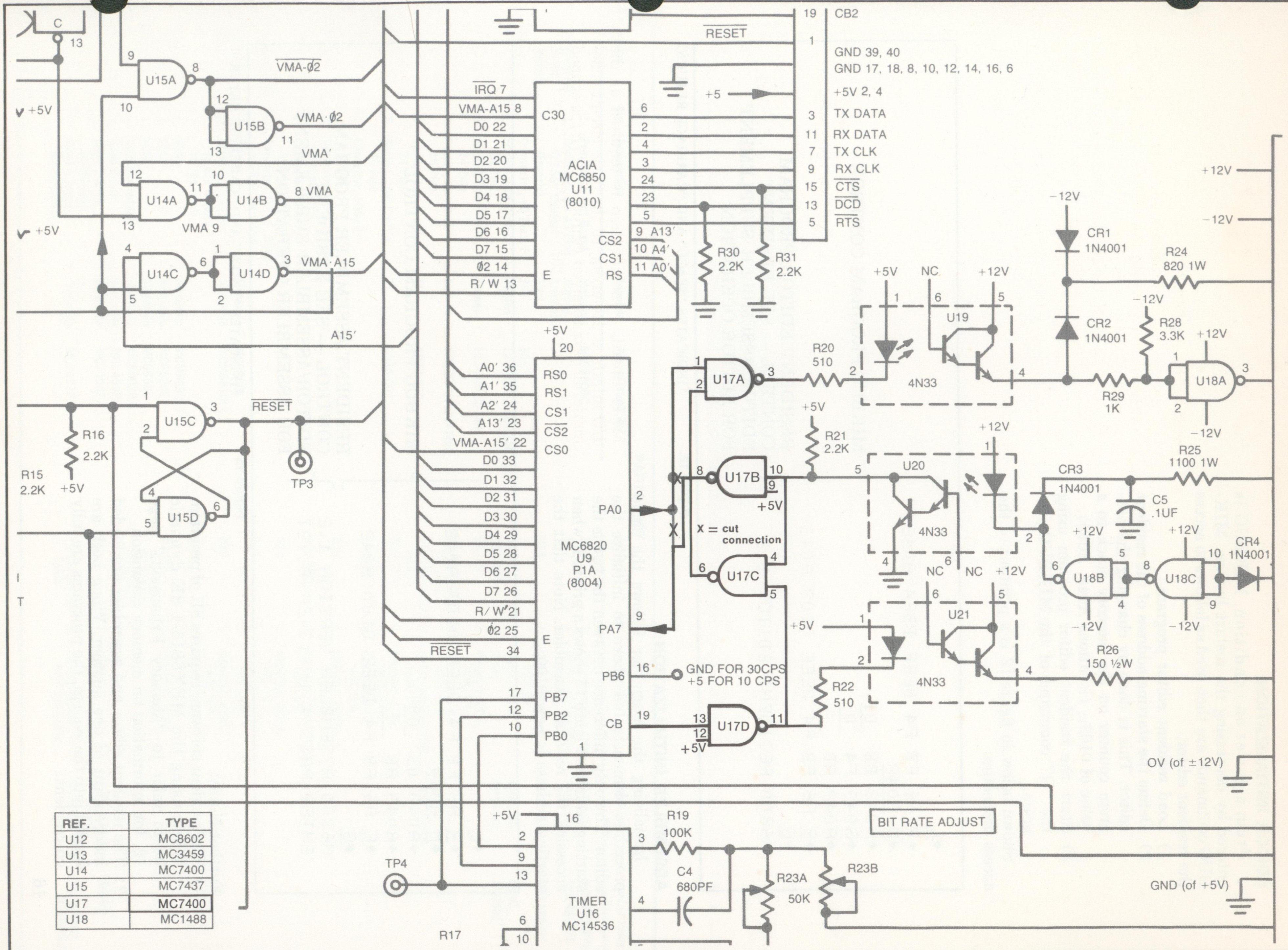
Hardware Modification

To operate the resident software with the MEK Board, a modification is required at the I/O interface. Referring to the basic circuit schematic, all data received (through U17B) is echoed back to the output (through U17A). This transfers all input data back to the printer of the terminal.

When using the resident assembler, this feature produces an undesirable printout. While reading the source tape during assembly, the source listing will be printed as it is read. The source listing will therefore be intermixed with the assembly listing/object listing being generated by the assembler and outputted through the PIA. A change to the MEK Board must be made to suppress the echo when loading data into the system. Figure 11 shows the required changes to suppress the echo during a data read (CB=1). Note that the MC1489 is replaced by the MC7400 at U17 and PC track must be cut between the following locations to implement the desired gating function:

U9-2 to U9-9
U17-1 to U17-8

Note also that this change will suppress a printout when using the MIKBUG program "load" function. If a tape printout is desired, it must be done locally at the terminal.



REF.	TYPE
U12	MC8602
U13	MC3459
U14	MC7400
U15	MC7437
U17	MC7400
U18	MC1488

Figure 11

EDITOR INITIALIZATION:

From a power on condition, MIKBUG is entered by depressing the restart button. MIKBUG commands are then used as follows to access the resident editor.

1) Load resident editor program.
2) Define the starting address of the resident editor. This is done by changing the program counter on the memory stack to a value of 0103_{16} (locations A048, A049).
3) Start the resident editor program using the "G" command of the MIKBUG program.

Shown below in figure 12 is a printout of the access procedure.

```
*L
*R A6 FB F4 DEEE BBF4 A042
*M A048
*A048 BB  01
*A049 F4  03
*A04A AB  †
*R A6 FB F4 DEEE 0103 A042
*G
M6800 RESIDENT EDITOR 1.2
```

MIKBUG PROGRAM CONTROL

RESIDENT EDITOR PROGRAM CONTROL — SEE RESIDENT EDITOR/ASSEMBLER SUPPLEMENT FOR EDITOR OPERATION

Figure 12. †INDICATES "SPACE" AND "CARRIAGE RETURN"

ASSEMBLER INITIALIZATION:

Initializing the assembler follows the same procedure demonstrated above to initialize the editor. The only difference between the two is the starting address. Figure 13 shows a printout when accessing the resident assembler. Note that the starting address is now 0100_{16}.

```
*L
*R A6 FB F4 DEEE 0103 A042
*M A049
*A049 03  00
*A04A AB  †
*R A6 FB F4 DEEE 0100 A042
*G
M6800 ASSEMBLER VERSION 1.2
ENTER PASS : 1P,1S,2P,2L,2T
```

MIKBUG PROGRAM CONTROL

RESIDENT ASSEMBLER PROGRAM CONTROL — SEE RESIDENT EDITOR/ASSEMBLER SUPPLEMENT FOR ASSEMBLER OPERATION

Figure 13 †INDICATES "SPACE" AND "CARRIAGE RETURN"

Further Notes:

1) The resident software requires 8K of program. When using the MEX6815-1 8K dynamic board refer to "Memory Expansion," page 12, for information on memory expansion.
2) The resident editor and assembler are loaded independent of one another. When both are loaded into the system, the programs are fully co-resident.

PARTS LIST

Symbol	Qty	Description	Supplier	Part or Stock No.
	1	Printed Circuit Board	Motorola	MEC6800C
P1 (External to PC Board)	1	Connector 2 x 43 Pins, 0.156″ Centers (Female)	Burndy Amphenol Alco	
P2	1	Connector, 40 Pins, P2	3M	3432
P3	1	Socket, Low Profile, 16 Pins	Barnes	121-20012-164FOR
	5	Socket, Low Profile, 14 Pins	Barnes	121-20012-144
	2	Socket, Low Profile, 16 Pins	Barnes	121-20012-164FOR
	4	Socket, Low Profile, 24 Pins	TI	C932-402
	3	Socket, Low Profile, 40 Pins	TI	C934-002
C1	1	Capacitor, 100μF, 16VDC		
C2, C3	2	Capacitor, 33PF, 500V, Dipped Mica		
C4	1	Capacitor, 680PF, 500V, Dipped Mica		
C5 thru C15	11	Capacitor, 0.1μF, 50V, ±20% Tantalum		
U1	1	MPU	Motorola	MC6800L
U6-U7	2	RAMs (possibility to add 4 more on the card)	Motorola	MCM6810L1
U8	1	ROM	Motorola	MCM6830L7
U9-U10	2	PIAs	Motorola	MC6820L
U11	1	ACIA	Motorola	MC6850L
U12	1	Dual Monostable	Motorola	MC8602P/L
U13	1	Quad Driver	Motorola	MC3459P/L
U14	1	Quad 2 Input NAND Gate	Motorola	MC7400P/L
U15	1	Quad 2 Input NAND Buffers	Motorola	MC7437P/L

Symbol	Qty	Description	Supplier	Part or Stock No.
U16	1	Programmable Timer	Motorola	MC14536P/L
U17	1	Quad Line Receiver RS232C	Motorola	MC1489P/L
U18	1	Quad Line Driver RS232C	Motorola	MC1488P/L
U19-U21	3	Opto-Coupler	Motorola	4N33
CR1-CR4	4	Diode	Motorola	1N4001
R4, R5	2	Resistor, 22Ω, ¼W, 5%		
R8, R12, R6, R7, R29	5	Resistor, 1K, ¼W, 5%		
R9, R13, R23A, R23B	4	Potentiometer, 50K, ¼W, 5%	TRW. Bourns.	
R10, R14	2	Resistor, 5.1K, ¼W, 5%		
R18, R2	2	Resistor, 10K, ¼W, 5%		
R19	1	Resistor, 100K, ¼W, 5%		
R20, R22	2	Resistor, 510Ω, ¼W, 5%		
R24	1	Resistor, 820Ω, 1W, 10%		
R25	1	Resistor, 1.1K, 1W, 10%		
R26	1	Resistor, 150Ω, ½W, 5%		
R28	1	Resistor, 3.3K, ¼W, 5%		
R1, R3, R11, R15, R16, R17, R21, R27, R30, R31	10	Resistor, 2.2K, ¼W, 5%		
S1	1	Switch	Control Switch Circuit	B8600
P3 Connector/ Cable	1	16 Pin Header with 10′ Cable	Assembly Corp.	CA-S16P02-281-A0-120

REF.	TYPE
U12	MC8602
U13	MC3459
U14	MC7400
U15	MC7437
U17	MC1489
U18	MC1488

INTERCONNECT SCHEMATIC FOR MOTOROLA MEK6800DI KIT

USING THE MOTOROLA MEK BOARD MEC6800C

MOTOROLA Semiconductor Products Inc.

BOX 20912 • PHOENIX, ARIZONA 85036 • A SUBSIDIARY OF MOTOROLA INC.

9697-4 PRINTED IN USA 1-76 IMPERIAL LITHO B54973