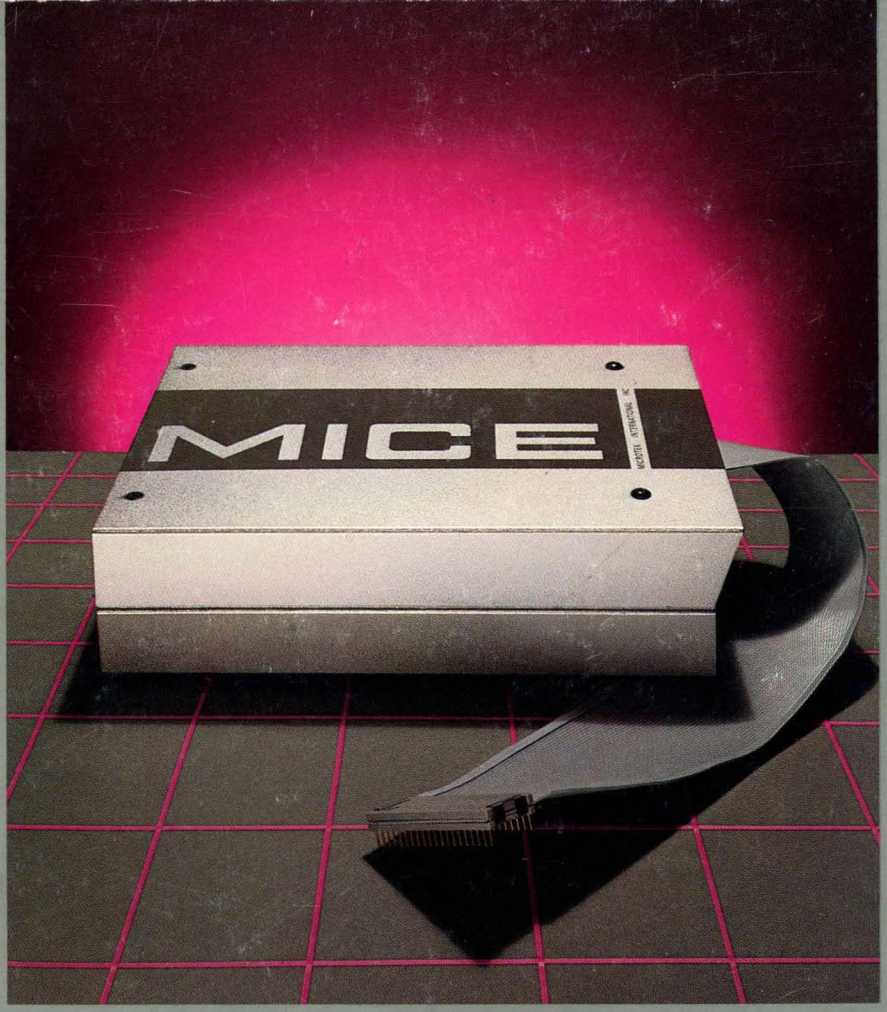


MICE USER'S GUIDE

PREFACE

This document provides information on the MICE products from MicroTek International, Inc. Installation and operation instructions are described.

The information presented in this document is believed to be accurate and complete, but no responsibility is assumed for any errors or omissions.

Additional copies of this document or other literature and product information may be obtained from:

MicroTek International, Inc.
2-1, Science Road 1
Hsinchu Science-based Industrial Park
Hsinchu, Taiwan, 300, R.O.C
Telephone: (035) 772155-7
Telex: 32169 MICROTEK

MicroTek Lab., Inc.
17221 South Western Avenue
Gardena, California 90247
Telephone: (213) 538-5369
Telex: 696334 BENNY GDNA

Sales Representative Office
GYC, Inc.
100 Clement Street
San Francisco, California 94118
Telephone: (415) 751-7150

Printed in U.S.A.
July 1982

CONTENTS

CHAPTER 1
INTRODUCTION TO MICE

MicroTek's Micro In-Circuit Emulator, MICE, is a low-cost development tool that emulates most of the industry-standard microprocessors and is setting a new standard for universal, high-performance emulation, at an exceptionally low cost-per-function.

Traditionally, microprocessor development has been done on a dedicated system. This computer system would have its complement of peripheral devices and sufficient mass storage to accommodate the user's program and a variety of support software such as editors, debuggers, file management programs, etc. Attached to this sytem would be a dedicated emulator which allowed the designer to examine the system to assure that all components, both hardware and software, were functioning properly.

More recently, a variety of general purpose computers have been designed for use in microprocessor development. Through the use of cross assemblers and cross compilers, code can be generated for the target system. However, the emulator products are again dedicated to that computer.

With MICE, a third approach to microprocessor development is available; one that uses fewer resources, performs most of the same functions, and yet cost a fraction of its predecessors. Consisting of a control card and a separate personality card for the microprocessor to be emulated, the MICE module is controlled via an RS-232C compatible interface. All software necessary to operate the MICE is contained in EPROMs within the module. The MICE can be operated using only a display terminal or in conjunction with a computer system. Different processors can be emulated by merely changing the personality card and associated EPROMs.

Some of the key features of MICE are:

- Operation at speeds up to the maximum rated frequency of the specified microprocessor.
- Target processor retains its entire memory and I/O spaces.
- Emulation memory which can be mapped into any 8K block of target processor memory.
- Enabling and disabling of hardware control signals to the processor with console commands.
- Single address breakpoint with loop count.
- Forward and backward tracing with 256 cycles of trace memory.
- Single character command set with help command to list all commands and the proper syntax.
- Using a single input command, any port can be sampled up to 256 times at intervals from 1 to 256 milliseconds; and multiple data output with a single port output command.
- Resident assembler and two-pass disassembler which assigns labels to subroutine and branch addresses.
- Built-in memory diagnostics and block memory transfer for the target processor's memory.
- Downloading and uploading of target program between MICE and host computer systems.

Some of the advantages of MICE over the other emulators are:

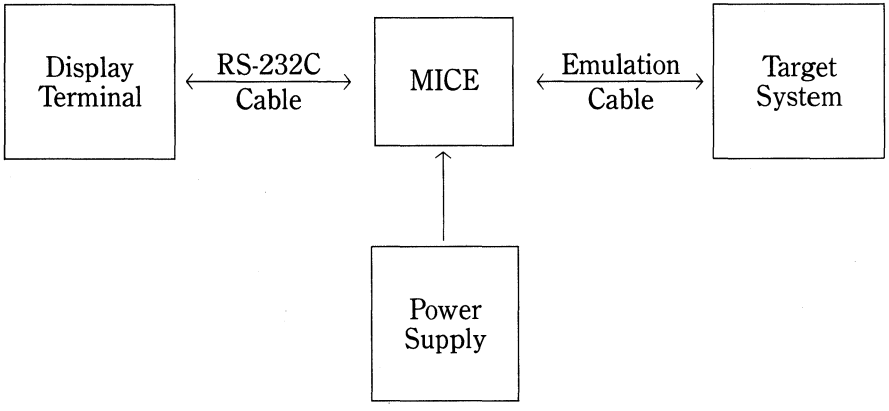
universality A wide variety of microprocessors can be emulated using this single, low-cost development tool. You can avoid both the inflexibility of dedicated systems and the heavy capital investment required for the general purpose systems.

versatility The emulation of a different processor requires only the changing of a personality card and the control EPROMs. Multiple design projects can be carried out concurrently, and the control card can be reused on different projects.

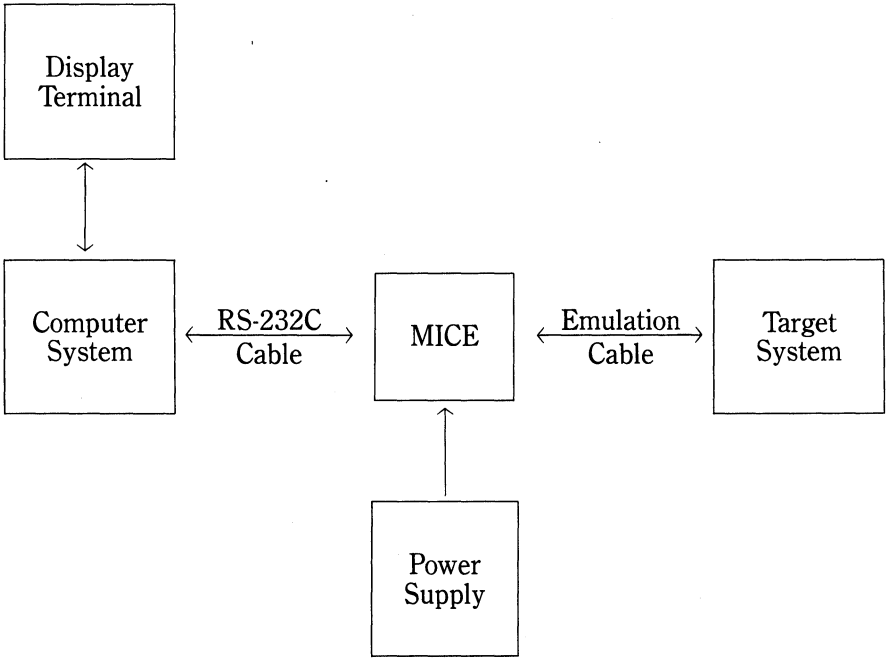
flexibility With MICE providing its own resident assembler and disassembler, it can be minimally configued with just a display terminal and a power supply. After the target program has been downloaded from the host computer, testing can be done off-line, thereby freeing the computer.

MICE Operating Configurations

The MICE module can be used in either of two possible configurations. The simplest configuration only requires an RS-232C compatible terminal as the display and command entry device.



If a computer system is used as the controlling device, the configuration is similar to the one above; the computer with its display terminal now replaces the lone terminal.



When a computer system is interfaced to the MICE module, a driver program must be resident in the computer system. Driver programs for various systems have already been written, including

- the Apple computers
- the Intel development systems
- the Digital Equipment mini- and main-frame computers

For an updated list on the computers supported and power supply vendors, contact your local MICE representative.

MICE Applications

Because of its unique design, MICE offers new applications which include:

1. Inexpensive evaluation of new microprocessors is made possible since there is no longer a need to purchase either an evaluation board or an expensive development system.

2. Several designers can now share the use of a single development system, eliminating the difficult problem of allocating a single resource and the expensive need of multiple work-stations. Large programs can be edited, assembled or compiled, and then down-loaded using MICE into the target system. Since MICE has its own assembler and disassembler, the programs can then be tested using only a display terminal.

3. In-field or service center testing is ideal. Rather than sending the suspected bad board back to the factory for service, MICE's compact size makes it portable and allows it to be brought to remote sites for diagnosing problems; thus saving time and customer inconvenience. With its RS-232C interface, MICE can be quickly interfaced to any compatible display terminal. Diagnostic programs can be generated using the resident assembler.

4. Personal computers can be upgraded to a development system at a fraction of the typical costs. Driver programs for various computers have already been written, allowing programs assembled or compiled to be downloaded.

CHAPTER 2
MICE INSTALLATION PROCEDURES

All hardware and software items necessary for operation are contained in the MICE shipping package. The package includes a MICE module, containing a control card and a personality card for the microprocessor being emulated; an emulator cable for connection to the target system; a DC power cable; and a user's guide. After unpacking, check that all the items are complete and undamaged. If any item in the package is damaged or missing, contact your local MICE representative immediately.

MICE Specifications

The MICE module comes in either an aluminum or a plastic case with the following dimensions:

Width —6.89 in. (17.5 cm)
Height—2.76 in. (7.0 cm)
Depth —9.06 in. (23.0 cm)

and weighs approximately 32 oz (900 g). If the module comes in an aluminum case, its dimensions are a bit smaller; also, be careful where it is placed since aluminum conducts electricity.

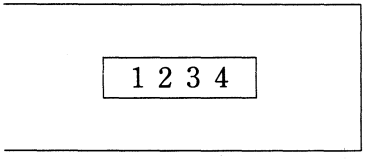
The minimum and maximum operating and storage limits for temperature and humidity are:

	operating		storage	
	min	max	min	max
Temperature in degrees Fahrenheit/Celsius	32/0	122/50	14/−10	149/65
Relative humidity (without condensation)	20%	80%	20%	90%

In order to use MICE, the MICE module requires three power supply voltages; each of which must be regulated to within 5 percent of nominal:

+ 5 VDC at 1.5A (maximum)
+12 VDC at 0.2A (maximum)
−12 VDC at 0.2A (maximum)

Using the supplied power cable, the mating female connector is to be connected to the pins at the rear of the MICE module with the locking tab pointing up. The power connector pinouts are as follows:



Pin	Description
1	+ 5 VDC
2	Ground
3	+12 VDC
4	−12 VDC

It is very important to make sure that the correct voltages are connected to the proper pins, otherwise, MICE may suffer severe damages to its circuitry; also, when turning off the power to the MICE module, always wait a few seconds to allow the capacitors to fully discharge before turning on the power again.

Communicating with MICE

Whether the MICE module is connected to a terminal or to a computer system, the connection between the controlling device and MICE is across a programmable RS-232C compatible interface. To gain access to the controls, the cover to the MICE module must first be removed. Next, carefully lift the top, personality card out to gain access to the bottom, control card. Locate the switch designated as DSW7 on the control card. This switch sets the following power-up and reset modes:

Switch Section	Description
1-3	Baud Rate
4	7/8 Data Bits
5	Disable/Enable Parity
6	Odd/Even Parity

The number of stop bits is permanently set to one; and the communication is full duplex.

MICE modules are shipped from the factory with the switch preset to 2400 baud, seven data bits, and parity enabled with even parity. The switch position for this default selection is shown below.

	Switch Section					
Mode	**1**	**2**	**3**	**4**	**5**	**6**
2400 Baud	ON	OFF	ON			
7 Data Bits				ON		
Parity Enable					OFF	
Even Parity						OFF

The baud rate is initialized or updated during power-up or reset. Any of 8 different baud rates (110-9600) can be selected. To do this, place sections 1, 2, and 3 of switch DSW7 to the proper positions as shown below.

Baud Rate	Switch Section 1	2	3
110	OFF	OFF	OFF
150	ON	OFF	OFF
300	OFF	ON	OFF
600	ON	ON	OFF
1200	OFF	OFF	ON
2400	ON	OFF	ON
4800	OFF	ON	ON
9600	ON	ON	ON

The number of data bits and the parity selection are also initialized or updated during a power-up or reset. To change the selection settings, place sections 4, 5, and 6 of the switch to the proper positions as shown below.

Mode	Switch Section 4	5	6
8 Data Bits	OFF		
7 Data Bits	ON		
Parity Enable		OFF	
Parity Disable		ON	
Even Parity			OFF
Odd Parity			ON

When parity is disabled, the odd/even parity switch section can be set to either position since it is ignored.

DTE or DCE

Having selected the proper data rate and characteristics, we must now determine whether our controlling device has a data terminal equipment (DTE) interface or a data communication equipment (DCE) interface. Display terminals are usually equipped with DTE interfaces; computer systems are usually equipped with both. If by chance the MICE module is to be interfaced to a modem for remote use, modems are usually equipped with DCE interfaces.

There are three methods for determining the type of interface on the controlling device, listed here in preferential order:

1. Data are transmitted on pin 2 and received on pin 3 of the D-connector for DTE devices; the reverse is true for DCE devices.
2. The voltage on pin 2 with respect to pin 7 of the D-connector is approximately −12 VDC for DTE devices, and on pin 3 for DCE devices.
3. Connect the two devices together. If it works, it must be correct; otherwise, it might be reversed.

The header designated as HDR2 on the control card is used to configure the interface between MICE and the controlling device. The pin definitions on the header are shown below.

D-connector			MICE Interface Logic
pin 8...1		14..(O)	CF — Received Line Signal Detector
pin 20...2		13..(I)	CD — Data Terminal Ready
pin 6...3		12..(O)	CC — Data Set Ready
pin 5...4		11..(O)	CB — Clear to Send
pin 4...5		10..(I)	CA — Request to Send
pin 3...6		9..(O)	BB — Received Data
pin 2...7		8..(I)	BA — Transmitted Data

Pins 1 through 7 on the header are connected to the female D-connector while pins 8 through 14 are connected to the MICE RS-232C interface logic. The symbols "I" and "O" within the parenthesis indicate whether the signals are incoming or outgoing with respect to the logic.

MICE modules are shipped from the factory with the header wired straight across, pins 1-to-14, 2-to-13, etc. In this configuration, MICE is ready for most DTE interfaces; which means that if a display terminal is connected to the MICE module using a straight-wire cable with male D-connectors, the MICE module should respond when power is applied. The display terminal should always be used to check new MICE units to insure that units shipped have not been damaged.

If the controlling device has a DCE interface, the header must be rewired because the incoming signals are now outgoing, and vice versa, on the same pins.

DTE interface	DCE interface	
8 to 7	8 to 6	(I) Transmitted Data
9 to 6	9 to 7	(O) Received Data
10 to 5	10 to 4	(I) Request to Send
11 to 4	11 to 5	(O) Clear to Send
12 to 3	12 to 2	(O) Data Set Ready
13 to 2	13 to 3	(I) Data Terminal Ready
14 to 1	no connect	(O) Received Line Signal Detector

Note that the first six signals are paired, where one is the complement of the other.

Finally, the MICE module requires that both the Request to Send and Data Terminal Ready inputs, pins 10 and 13 respectively on the logic side of the header, to be at +12 VDC before it transmits any data. If the controlling device does not supply the necessary voltage, it may be obtained by not connecting the pins to the designated pins as indicated in the above table, but connecting them to pin 14 (Received Line Signal Detector) of the header which is always at +12 VDC. To accommodate computer systems which are sending commands and data too fast for MICE, the Data Set Ready output, normally at +12 VDC, is pulled low by MICE to −12 VDC; the signal is restored to the +12 VDC when MICE is again ready. To accommodate display terminals with slow carriage-returns or line-feeds, six null characters are always transmitted after a new line is issued.

Applying Power to MICE

We are now ready to do a preliminary test on our MICE. First, replace the personality card to its original position, reconnecting the ribbon cable between the two boards if it had been removed earlier; the arrows indicate pin 1 and must be aligned when attaching the cable to the connector. Next, connect the RS-232C cable from the controlling device to the female D-connector on the rear of the MICE module. Finally, with the controlling device ready, connect the power cable with the locking tab pointing up, and apply power. Within a few seconds, the start-up message should be displayed.

***** MICE-type V#. # *****

>

---

type identifies the target processor being emulated by the personality card.

#. # indicates the revision level of the controlling program on the control card.

> is the prompt character indicating that MICE is ready for a command.

During the delay, the RAMs and the EPROMs on the control card are tested. Component failures are listed as detected in the format:

U## – FAILURE

where:

is the component number on the control card

If there is no response from MICE, here are some things to try:

1. Check the RS-232C cable connection at both ends.
2. Check the power supply connections and voltages.
3. Check that the header has the proper interface.
4. Check that both Request to Send and Data Terminal Ready, pins 10 and 13 respectively on the logic side of the header, are at +12 VDC.
5. Check that the controlling device has the proper voltage level requirements on its RS-232C inputs for transmission and reception.
6. Check that the baud rates of the controlling device and the MICE modules are the same, resetting if necessary. If a message does appear but is garbled, any combination of baud rate, data length, or parity could be incorrectly set.
7. Check the RS-232C cable for incorrectly wired or loose pins.
8. If a computer system is the controlling device, check that the driver progam is running and the RS-232C cable is connected to the correct port and that the port is working.

Remember that the power must be turned off then on again since the configuration switch is only read during a power-up or a reset. It is also important to wait a few seconds before the power is turned on to allow the capacitors to fully discharge.

If the MICE module still does not respond, contact your local MICE representative for further directions.

On the other hand, a proper message displayed means that we are almost done. We must next double check that the number of data bits and the parity have been set correctly. In response to the command prompt character, ">", enter a question mark, "?", immediately followed by a carriage-return. If the data bits and parity are correctly matched, the MICE command summary is listed, otherwise, the error message "WHAT?" is printed. Reset the switch sections as necessary and remember to wait a few seconds before power is turned on again.

Finally, alphabetic characters must be entered in upper-case. Aside from the character "r", which is the command to reset the MICE module, all other lower-case characters are not recognized.

Interfacing to Target

MICE modules are shipped from the factory configured to run in the stand-alone mode—running without a target system. The mappable emulation memory is enabled and preset to begin from the program memory address 0000H. With the exception of the 8048 family, the personality card has a six or eight section switch; five sections affect the mappable memory as shown below.

	Switch Section				
Mode	**2**	**3**	**4**	**5**	**6**
0000H - 1FFFH	ON	ON	ON		
2000H - 3FFFH	ON	ON	OFF		
4000H - 5FFFH	ON	OFF	ON		
6000H - 7FFFH	ON	OFF	OFF		
8000H - 9FFFH	OFF	ON	ON		
A000H- BFFFH	OFF	ON	OFF		
C000H - DFFFH	OFF	OFF	ON		
E000H - FFFFH	OFF	OFF	OFF		
Memory Enable				ON	
Memory Disable				OFF	
Write Enable					ON
Write Protect					OFF

When MICE is used to replace the processor in the target system, the 40-pin ribbon cable is used. With the power off, one end of the cable mates with the 40-pin male connector on the personality card; the arrow indicates pin 1 and must be aligned when attaching the cable. Next, carefully extract the processor from the target system and insert the male plug on the end of the cable into the 40-pin socket; pin numbers are marked on the plug and must be inserted as indicated. A number of jumpers and switch sections on the personality card select the clock source to the processor and other emulation parameters. Since these jumpers and switch settings vary for different types of processors, consult the corresponding guide in the appendices on their meaning. Also consult the appendices if a different personality card is to be installed.

Finally, care must be taken when using the ribbon cable; it is not the same for all processors. If your cable is not identified, mark it yourself so that you can distinguish it from the others. Also, the 40-pin plug that is inserted in place of the target processor has pins which can be easily damaged. If possible, protect those pins by plugging them into a 40-pin socket, using the unit together for insertions and removals. Lastly, the cable and the personality card add additional loads to the target processor and drivers which may cause problems if the target system design is marginal.

CHAPTER 3
MICE COMMAND LANGUAGE

All MICE products have a common set of commands that are identified by a single character. Regardless of the target processor being emulated, only a different personality card is changed with no time being wasted in learning a new command language.

In this chapter, the commands are described with all the possible options that are available although not all options are applicable to the different types of processors. For the particular processor being emulated, either consult the help (?) command or the corresponding guide in the appendices. The commands described in this chapter are grouped as follows:

MICE Utility Commands
- ? Help Command
- ! Attention Command

Memory and Port Contents Commands
- M Memory Display/Change Command
- T Memory Test/Transfer Command
- A Assemble into Memory Command
- Z Disassemble Memory Command
- I Port Input Command
- O Port Output Command

Registers and Control Signals Commands
- R Register Display/Change Command
- J Program Counter Change Command
- D Disable Control Signal Command
- E Enable Control Signal Command
- X Reset Target Processor Command

Emulation and Trace Control Commands
- G Begin Emulation Command
- H Halt Emulation Command
- C Cycle Step Command
- S Instruction Step Command
- F Forward Trace Command
- B Backward Trace Command
- L List Trace Command

Utility Commands Involving a System
- : Download Command
- / Download Command
- U Upload Command

Command Syntax

MICE indicates that it is ready to accept a command line by printing a greater-than character, ">", on a new line. A command may then be entered which must be terminated by a carriage return. The general syntax of MICE commands is:

command [**parameters**]<**cr**>

where:

command is the single character for the command.

parameters are one or more variable data supplied with the command. Parameters are alpha-numeric; and when a numeric parameter is called for, it must be entered in hexadecimal form.

Where a comma is shown in the syntax, either a comma or a space can be used.

Notations and Conventions

A set of conventions are used to describe the structure of the commands. The notation and the rules are as follows:

1. An upper-case entry must be entered as shown.

2. A lower-case entry in the description of a command is the class-name for a class of parameters. To create an actual operable command, a particular member of this class must be entered. A class-name never appears in a actual operable command. For example, the lower-case entry:

start-address

means that the MICE will only accept a hexadecimal value as an address in the target processor's memory space. The maximum value accepted depends on the target processor being emulated.

3. A required entry is shown without any enclosures, whereas an optional entry is denoted by enclosing in brackets. For example, in the command description

G [**address**]

the command "G" is required; and the brackets around the entry "address" means that its selection is optional in this command. Where brackets are within another set of brackets, the entry enclosed by the inner brackets may only be entered if the items outside those inner brackets have already been entered. For example, in the command description

H [**address** [, **count**]]

the command "H" is required; and the brackets means that the selection of "address" and ", count" is optional in the command. However, an address must first be entered if a count value is to be specified.

4. Where an entry must be selected from a choice of two or more entries, the choices for the required entry are enclosed in braces with the entries separated by vertical bars. For example,

{**I**|**T**|**A**}

indicates that either "I" or "T" or "A" must be entered.

5. Where a choice exists for an optional entry, the choices are enclosed in brackets with the entries separated by vertical bars. For example,

[**P**|**I**|**X**]

indicates that either "P" or "I" or "X" may be entered.

6. Where an entry can be optionally repeated, the repeatable entry is enclosed in brackets followed by an ellipsis, "...". For example,

[, qualifier]...

indicates that ", qualifier" can be repeated as many times as required.

7. A decimal number in the text of this manual is without a suffix; whereas, a hexadecimal number has an "H" for a suffix. For example,

1234
1234H

the first value of "1234" is a decimal number, but "1234H" is a hexadecimal number having a decimal value of 4660.

8. In the examples, input normally entered by the user are differentiated from MICE output by being underlined.

Editing Characters

Each character entered on the keyboard is stored in a line editing buffer until the RETURN key is pressed. If too many characters are entered without a RETURN, an error message is printed, and the command is ignored. The line editing buffer can be edited or entirely deleted by using special non-printable editing characters. Control characters are entered by holding down the control key (CONTROL or CTRL) while the character is typed.

BACKSPACE deletes the preceding character from the line buffer and from the display. Repeated usage is allowed. Control-H performs the same function. When a hardcopy terminal is used instead of a display screen, RUBOUT should be used.

RUBOUT deletes the preceding character from the line buffer and echoes the deleted character on the display preceded by a backslash character, "\ ". Repeated usage is allowed. On some terminals, this key may be labeled as DELETE or just DEL.

ESCAPE ignores the current contents of the line buffer and prompts (">") for a new command on the next line. This key is also used to terminate commands in process and to return to the prompt state. On some terminals, this key may be labeled as ESC; CONTROL-Y performs the same function.

CONTROL-R causes a carriage-return/line-feed, followed by the redisplaying of the current undeleted contents of the line buffer. This is useful to see a clean copy of the command line after RUBOUT has been used.

CONTROL-X ignores the current contents of the line buffer and prompts (">") for a new command on the next line.

Operator-Controlled Pauses

The following control characters can be used to suspend the MICE output, allowing inspection of the display, and then resume the output.

CONTROL-S suspends MICE output.

CONTROL-Q resumes MICE output after the CONTROL-S is given.

MICE Utility Commands

These commands allow the user to query MICE for a summary of the commands and syntax that are available for the target under emulation. In addition, the user can query to have displayed the processor type being used.

Help Command—?

?

? is the command keyword.

The command summary for the target processor currently emulated is displayed on the terminal. Although the commands are common regardless of the target processor being emulated, some of the parameters may differ for different processors.

During the display, CONTROL-S may be used to stop the display, and CONTROL-Q used to resume the display. ESCAPE or CONTROL-Y may be used to abort the display before the entire summary is displayed.

Example: Display the command summary for the target processor currently being emulated.

```
>?<cr>
ASSEMBLE                A loc
BACKWARD TRACE          B tri[a1[a2[q1..qn]]]
CYCLE STEP              C
DISABLE                 D {I|T|A}
ENABLE                  E {I|T|A}
FORWARD TRACE           F tri[a1[a2[q1..qn]]]
EXECUTION               G [a]
BREAKPOINT              H [a[c]]
INPUT                   I port[c[time]]
JUMP                    J a
LIST TRACE              L [c1[c2]]
MEMORY                  M [P|I|X] a[a[d]]
OUTPUT                  O port d1[d2[d3[d4]]]
REGISTER                R [B0|B1] [A|0|1|2|3|4|5|6|7|T|P]
INSTRUCTION STEP        S [c]
TEST/TRANSFER           T [P|I|X] a1 a2 {S|M|a3}
UPLOAD                  U a1 a2 [I|T]
RESET                   X [a]
DISASSEMBLE             Z a1 [a2]
DOWNLOAD (INT)          :
DOWNLOAD (TEK)          /
HELP                    ?
ATTENTION               !

>
```

The notation of the summary does not fully follow the notation and conventions previously described to avoid a lengthy display.

Attention Command—!

!

! is the command keyword.

The target processor type currently being emulated by the personality card in the MICE module is displayed on the terminal. Up to six characters are used for the identification.

Example: Display the processor type currently being emulated.

```
>!<cr>
EP8048
>
```

The response for other processor types supported are:

6502	(MOS Technology, Inc.):	EP6502
8085	(Intel Corporation):	EP8085
NSC800	(National Semiconductor Corporation):	NSC800
Z80	(Zilog Corporation):	EP-Z80
Z80R	(Zilog Corporation):	EPZ80R

Memory and Port Contents Commands

These commands give access to the contents or current values stored in the designated memory locations or input/output ports. The type of memory, maximum amount of memory, and total number of ports differ for the various processors. For the particular processor being emulated, consult the corresponding guide in the appendices.

Commands with no memory type requirements always operate on the target's program memory. Targets with only one type of memory have no distinction between program and data memory.

MICE always checks that the memory type and the memory and port addresses are valid before an operation is performed. References to an invalid type or invalid address result in having an error message being printed and command ended. MICE, however, does not verify that there is memory or anything connect to the port when completing an operation. In these cases, data written are lost, and data read are random.

Memory Display/Change Command—M

M [type,] start-address [, end-address [, value]]

M	is the command keyword (Memory).
type	is a single alphabetic character indicating the type of memory to be used for processor with more than one type of memory.
start-address	is the hexadecimal address of the target processor from which to begin the display of memory content.
end-address	is the hexadecimal address of the target processor indicating the last memory location of the range to be either displayed or filled.
value	is the hexadecimal value to be written into the specified memory range.

If only the start-address is specified, the content for that memory location is first displayed. MICE then waits for an input from the user. To advance to the next memory address, a line-feed should be entered; the next memory location's content is then displayed, and MICE again waits. Entering a line-feed repeats the entire sequence. If either a carriage-return or an escape is entered, the command is ended. To change the memory content displayed, enter the new hexadecimal value before the line-feed or carriage-return. For target processors with more than one type of memory, the program memory is always selected when the type specification is omitted. To determine whether the target processor being emulated has more than one type of memory, either consult the help (?) command or the corresponding guide in the appendices

Example: Display and change the memory contents from address location 0210H of our 8048 target. The program memory type "P" is assumed.

```
>M210<cr>
0210 38<lf>
0211 79 59<lf>
0212 0C<cr>
>
```

On the other hand, an end-address specification causes the memory contents in the defined memory range to be displayed. The end-address must be greater-than or equal-to the start-address or an error message is printed and the command ended. The display can be aborted by entering an escape. Again the type specification can be used to indicate the memory type is to be displayed.

Example: Display the program memory contents 0209H to 0217H for our 8048 target processor overlapping the previously altered area:

```
>MP 209 217 <cr>
      00  01  02  03  04  05  06  07  08  09 0A  0B  0C 0D  0E  0F
0200                                      8D  15  47  92 0C  7F  93
0210  38  59  0C  83  01  28  38  48
>
```

Finally, a value specified causes that value to be written into the defined range. MICE does not verify the content change; and the command can be aborted by entering an escape although some of the memory locations may be altered. The type specification can be used to indicate the memory type to be changed.

Example: Write the value 28 into the external memory range 0024H to 0053H of our 8048 target processor.

```
>M, X, 0024, 53 28<cr>
>
```

Memory Test/Transfer Command—T

T [type,] start-address, end-address {S|M|dest-address}

T	is the command keyword (Test/Transfer).
type	is a single alphabetic character indicating the type of memory to be used for processor with more than one type of memory.
start-address	is the hexadecimal address of the target processor from which to begin the test or transfer operation.
end-address	is the hexadecimal address of the target processor indicating the last memory location of the range to be either tested or transferred.
S	is the alphabetic character indicating that the sum of the contents in the specified range is to be displayed.
M	is the alphabetic character indicating that a memory test is to be performed for the specified range.
dest-address	is the hexadecimal address of the target processor to which the data is to be transferred.

Start-address and end-address defines a range in the target processor's memory whose memory contents are either to be tested or transferred. The end-address must be greater-than or equal-to the start-address or an error message is printed and the command ended. For target processors with more than one type of memory, the program memory is always selected when the type specification is omitted. To determine whether the target processor being emulated has more than one type of memory, either consult the help (?) command or the corresponding guide in the appendices.

If the alphabetic character "S" follows the range specification, the hexadecimal sum of the contents for the indicated range, with carry added back, modulo 256, is displayed.

Because our 8048 does not support this command, these examples assume that one of the other processors is being emulated.

Example: First display the program memory contents for the range 0045H to 004DH, and then perform a sum for the same range.

```
>M 45, 4D<cr>
      00  01  02  03  04  05  06  07  08  09 0A 0B 0C 0D 0E 0F
0040                      39  78 D2  F1 CB  69  03 2C  50
>T45, 4D S<cr>
THE CHECKSUM IS 2A
>
```

On the other hand, an "M" following the range specification causes a memory test to be performed on the indicated range. The upper-byte and lower-byte of the address whose memory is to be tested are exclusive-or'ed and written into the memory for the entire range. The entire range is then verified and the complement value is next written. The entire range is again checked. If the comparison fails in either pass, the test is stopped at the failed address and that address is displayed. The original contents in the memory are destroyed in this test. Testing in progress may also be aborted by entering an escape.

Example: Test the memory range 0024H through 0039H of our target processor.

```
> T, 024 0039, M <cr>
RAM OK!
>
```

Finally, a dest-address specification indicates that the memory contents in the defined range is to be transferred into the memory beginning at dest-address. Data are transferred one location at a time from either the starting or the ending address of the range depending on whether there is any overlap.

Example: Transfer the memory contents in the range 0046H through 005FH of our target processor to the memory beginning at 0027H.

```
> T 46 5F 27 <cr>
>
```

Assemble into Memory Command—A

A start-address

A is the command keyword (Assemble).

start-address is the hexadecimal address of the target processor's program memory into which MICE begins storing the assembly language program entered.

Rather than entering programs or program changes in machine code using the previously described memory change (M) command, the MICE resident assembler accepts and converts mnemonic inputs into machine code. The converted code is then stored into the target processor's program memory starting from the indicated start-address. The assembler does not recognize symbolic labels; constants and addresses are entered as hexadecimal values. The instruction mnemonics accepted are those that had been adopted by the original manufacturer for the processor being emulated.

After the command keyword and start-address are received, MICE displays the following column headings:

```
LINE    LABEL    SOURCE CODE    LOC    OBJ
0001             *
```

where the hexadecimal value of 0001H under the column LINE indicates the line number being entered, and the asterisk, "*", indicates the new cursor position. MICE then waits for an assembly language line input from the user. The line entered can not cross into the next column, LOC, and must be terminated with either a line-feed or a carriage-return. MICE assembles the line and stores the machine code from the indicated start-address. The first program memory location to be stored is printed under the column LOC, and the data stored under the column OBJ. If a line-feed had been entered, the next line number is printed, and MICE again waits. If a carriage-return had been entered instead, the command is ended with the total number of lines entered reported as a hexadecimal value.

Example: Enter a 5 line 8048 program from 0000H with an error on the third input line.

```
>A 0<cr>

LINE   LABEL   SOURCE CODE          LOC   OBJ
0001           MOV A, #0<lf>        0000  2300
0002           MOV  R0,  A<lf>      0002  A8
0003           MVO R1, A <lf>
ERROR CODE, TRY AGAIN!

0003           MOV R1,A  <lf>       0003  A9
0004            INC A<lf>           0004  17
0005           JMP 2<lf>            0005  0402
0006           <cr>

YOUR TOTAL LINES IS 0005
LINE-ASSEMBLER COMPLETED
>
```

When making program changes into an existing program, it is advisable to check the program around the modifying area using the disassemble memory (Z) command before and after the change. Rechecking the area is important to assure that the new program changes do not affect the surrounding program. The disassemble memory command is described in the following pages.

Disassemble Memory Command—Z

Z start-address [, end address]

Z	is the command keyword.
start-address	is the hexadecimal address of the target processor's program memory from which to begin the display of disassembled memory content.
end-address	is the hexadecimal address of the target processor's program memory indicating the last memory location of the range to be disassembled for display.

If only the start-address is specified, the content for that memory location is disassembled and displayed. More data are read from subsequent locations to complete the instruction if necessary. On the other hand, an end-address specification causes the memory contents in the defined memory range to be disassembled and displayed. The end-address must be greater-than or equal-to the start-address or an error message is printed and the command ended. If an illegal machine code is encountered, the disassembler terminates at the illegal code's address. Entering an escape while MICE is displaying the disassembled data aborts the display.

MICE uses a two-pass disassembler with all branch and subroutine call addresses first identified and converted to labels; only a total of 450 labels are remembered for the disassembly. Depending on the range to be disassembled, there may be a pause before any line is displayed. Thus, to end the command before the display begins, enter and escape.

Example: Disassemble our previously entered 8048 program. Note that the byte following the end-address is also read in order to complete the instruction.

```
>Z0 5<cr>
LINE   LABEL   SOURCE CODE     LOC    OBJ
0001           MOV A, #0       0000   2300
0002   B0002   MOV R0, A       0002   A8
0003           MOV R1, A       0003   A9
0004           INC  A          0004   17
0005           JMP  0002       0005   0402

  DISASSEMBLY COMPLETED
>
```

Port Input Command—I

I port [, count [, duration]]

I	is the command keyword (Input).
port	is the hexadecimal address of the target processor's input port whose content is to be read and displayed.
count	is a hexadecimal value from 00H to FFH specifying the number of times the input port is to be read with 00H indicating that it is to be read 256 times.
duration	is a hexadecimal value from 00H to FFH specifying the interval in milliseconds between each read with 00H indicating that the interval is to be 256 milliseconds.

The input port command has two modes of operation. If neither the count nor the duration is specified, a range of port contents can be read and displayed beginning with the port address indicated.

MICE first reads and displays the contents for the port specified and waits for an input from the user. To advance to the next port, a line-feed should be entered; the next port's content is then read and displayed, and MICE again waits. Entering a line-feed repeats the entire sequence. If either a carriage-return or an escape is entered, the command is ended.

Example: Read and display the contents of ports 4, 5, and 6 of our 8048 target processor.

```
>I4<cr>
 04 DC <lf>
 05 87 <lf>
 06 FF <cr>
>
```

On the other hand, specifying a count with or without a duration specification only operates on the port indicated. MICE first reads the specified input port the number of times indicated; the contents are then displayed. Entering an escape while MICE is displaying the port contents aborts the display.

The duration specification allows the user to select the interval between each read to compensate for data inputs which are time critical; and if no interval is specified, the data are read in 1 millisecond intervals.

Example: Read and display the next 64 (40H) values of port 5 of our 8084 target processor at 10 (0AH) millisecond intervals. The display is aborted before the entire 64 values are listed.

```
>I, 05, 40 A
PORT  00  01  02  03  04  05  06  07  08  09  0A  0B  0C  0D  0E  0F
   05 43  23  56 AF  85 9F D7  E3 B6 AB  14  C5 B7 6D  07  16
      38  70  01  46 A4 C6 3B  78  93 FF FF FF FF FF FF FF
      FF FF FF <esc>
>
```

Note that the interval can be up to 256 milliseconds, which is approximately a quarter of a second; if the number of times the port is to be read is also 256, the elapse time before any data is displayed is approximately 65 seconds! Thus, to end the command even before the display begins, enter an escape.

Port Output Command—O

O port,value-1 [, value-2 [, value-3 [, value-4]]]

O is the command keyword (Output).

port is the hexadecimal address of the target processor's output port to be written.

value-1 is the hexadecimal value to be written into the specified output port of the target processor.

value-2 is the hexadecimal value of the second value to be written into the specified ouput port of the target processor.

value-3 is the hexadecimal value of the third value to be written into the specified output port of the target processor.

value-4 is the hexadecimal value of the fourth value to be written into the specified output port of the target processor.

If only one value is specified, it is written to the output port indicated. If more than one value is specified (up to a maximum of four), each value is written to the indicated port at 1 millisecond intervals with the first value going out immediately.

Example: Write the value 48H to port 01H of our 8048 target processor.

```
>O 1 48<cr>
>
```

Example: Write the values 1, 2, and 3 to port 05H of our 8048 target processor. The value 1 is written immediately followed by the value 2 after 1 millisecond, and the value 3 after another millisecond.

```
>O 05, 1 , 2,3<cr>
>
```

Registers and Control Signals Commands

These commands display and permit the user to change the register contents of the target processor being emulated. Other commands in this group enables or disables some of the control signals going to the processor.

Register mnemonics and register sizes depend on the target being emulated; rarely are they the same for the different types of processors. For target processors with more than one set of registers, some registers may cross over, being in both sets. For more detailed information on the registers of the target being emulated, consult the corresponding guide in the appendices.

The controllable signals available also depend on the target processor being emulated. The possible signals are:

A—external memory access
B—bus request and acknowledge
H—bus hold request and acknowledge
I—interrupt request
N—non-maskable interrupt request
T—trap request/timer interrupt

The available signal is either enabled or disabled at the input pin on the target processor. Only the sense input to the processor is affected; the circuits which generated the control signal in the target system are unaffected. For the target being emulated, consult the corresponding guide in the appendices on the control signals that are available.

Register Display/Change Command—R

R [bank,] [register]

R is the command keyword (Register).

bank is one or two characters indicating the group of registers to be used when processors have more than one set of registers.

register is a single character indicating the target register whose content is to be displayed.

All register contents are displayed if no specific register is specified. For target processors with more than one set of registers, the currently selected set is always used when the bank selection is omitted. To determine whether the target processor being emulated has more than one set of registers, either consult the help (?) command or the corresponding guide in the appendices.

Example: Display the default (current) set of register contents for our 8048 target processor.

```
>R<cr>
CURRENT BANK = 0
 A  0  1  2  3  4  5  6  7  T  P   PC
BC 48 02 67 89 48 00 85 42 00 08 0003
>
```

Example: Display the main and alternate sets of register contents.

```
>R ,B0<cr>
 A  0  1  2  3  4  5  6  7  T  P   PC
BC 48 02 67 89 48 00 85 42 00 08 0003
>R B1<cr>
 A  0  1  2  3  4  5  6  7  T  P   PC
28 34 09 AC 5D 76 5A EE 81 00 08 0003
>
```

If a register is specified, the content for that register is first displayed. MICE then waits for an input from the user. To advance to the next register, a line-feed should be entered; the next register's content is then displayed, and MICE again waits. Entering a line-feed repeats the entire sequence unless the next register displayed is the PC (program counter.) If either a carriage-return or an escape is entered, the command is ended. If the next register is the PC, the command is also ended. To change the register content for the register displayed, enter the new hexadecimal value before the line-feed or the carriage-return.

Example: Change a few of the register contents of our 8048 target processor.

```
>R<cr>
CURRENT BANK = 0
 A   0   1   2   3   4   5   6   7   T   P    PC
BC  48  02  67  89  48  00  85  42  00  08  0003
>R B0 A<cr>
A BC<lf>
0 48 BC<lf>
1 02 BB<lf>
2 67 02<lf>
3 89 17<lf>
4 48 <lf>
5 00 <cr>
>R    <cr>
CURRENT BANK = 0
 A   0   1   2   3   4   5   6   7   T   P    PC
BC BC BB  02  17  48  00  85  42  00  08  0003
>
```

To change the program counter (PC), the command J is used and is described in the following command description. Also, some registers may be common to both banks for processors with more than one set of registers. For the particular processor being emulated, consult the corresponding guide in the appendices.

Program Counter Change Command—J

J address

J is the command keyword.

address is the hexadecimal address of the target processor's program memory location to which the program counter is to be set.

Changing the target processor's program counter causes subsequent emulation to continue from that address in the program memory space.

Example: Change the program counter of our 8048 program from the current address of 0003H to 0002H.

```
>R<cr>
CURRENT BANK = 0
 A   0   1   2   3   4   5   6   7   T   P   PC
BC BC BB 02 17 48 00 85 42 00 08 0003
>J 02<cr>
>R B0<cr>
 A   0   1   2   3   4   5   6   7   T   P   PC
BC BC BB 02 17 48 00 85 42 00 08 0002
>
```

Disable Control Signal Command—D

D control-signal

D is the command keyword (Disable).

control-signal is a single alphabetic character
{A|B|H|I|N|T}
indicating a control signal to the target processor.
A—external memory access
B—bus request and acknowledge
H—bus hold request and acknowledge
I —interrupt request
N—non-maskable interrupt request
T—trap request/timer interrupt

The control-signal specified is deactivated at the pin on the target processor. This command only affects the sense input to the processor; the circuits which generated the control-signal are unaffected. When resetting the processor using the X command (to be described later), these signals are activated if the MICE emulator is connected to a power-applied target system.

Not all of the control signals are available to a target processor. For the particular processor being emulated, either consult the help (?) command or the corresponding guide in the appendices.

Example: Deactivate the interrupt request control signal to our 8048 target processor, thereby, not processing any interrupts which may occur.

```
>D I<cr>
>
```

Enable Control Signal Command—E

E control-signal

E is the command keyword (Enable).

control-signal is a single alphabetic character
{A|B|H|I|N|T}
indicating a control signal to the target processor.
A—external memory access
B—bus request and acknowledge
H—bus hold request and acknowledge
I—interrupt request
N—non-maskable interrupt request
T—trap request/timer interrupt

The control signal specified is activated at the pin on the target processor. When resetting the processor using the X command (to be described later), these signals are not activated if the MICE emulator is not connected to a power-applied target system.

Not all of the control signals are available to a target processor. For the particular processor being emulated, either consult the help (?) command or the corresponding guide in the appendices.

Example: Activate the interrupt request control signal to our 8048 target processor, thereby, allowing any interrupts which may occur.

```
>E I<cr>
>
```

Reset Target Processor Command—X

X [address]

X is the command keyword.

address is the hexadecimal address of the target processor's program memory to which the program counter is to be set after the reset.

A signal pulse is generated on the appropriate pin of the target processor under emulation to reset it. The processor's program counter, registers, and flags may be altered depending on the type of processor under emulation. Only the target processor is reset, any other circuits in the target system that are also connected to the same pin are not affected by the reset. In addition, control and interrupt signals to the target processor that are selectively controllable by the disable and enable commands are activated if the MICE emulator is connected to a power-applied target system.

An optional start address can be specified; it sets the target processor's program counter to the specified program memory address.

Example: Reset our 8048 target processor, showing the changes in the program counter, registers, and flags before and after.

```
>R<cr>
CURRENT BANK = 0
 A  0  1  2  3  4  5  6  7  T  P   PC
BC BC BB 02 17 48 00 85 42 45 08 0002
>X<cr>
>R B0<cr>
 A  0  1  2  3  4  5  6  7  T  P   PC
BC BC BB 02 17 48 00 85 42 00 C8 0000
>X3<cr>
>R B0<cr>
 A  0  1  2  3  4  5  6  7  T  P   PC
BC BC BB 02 17 48 00 85 42 00 C8 0003
>
```

When targets execute the program halt instructions, the message "PROGRAM HALT!!" is printed. The target processor is disabled; the reset command must be used to enable the target.

Emulation and Trace Control Commands

These commands control the mode of emulation for the target processor. Emulation may be stopped at user defined addresses, instruction-stepped, or cycle-stepped. Additionally, program operation may be recorded either forward from or backward to a user defined trigger address; up to 256 trace steps can be recorded. However, programs running in trace mode execute approximately 2 to 3 orders slower than real time since MICE stops the target processor after each cycle to record the status. Execution is also slower when emulation is to be stopped at a defined address since MICE must stop the target to compare the address after each cycle.

Status information displayed by the commands uses the following column headings:

IFADDR ADDRESS DATA STATUS

Where;

IFADDR is the instruction fetch address.
ADDRESS is the value on the address bus.
DATA is the value on the data bus.
STATUS is the type or processor activity

- A – interrupt acknowledge
- F – program memory fetch
- I – port input
- O – port output
- R – external/data memory read
- S – instruction fetch
- W – external/data memory write

Begin Emulation Command—G

G [**address**]

G is the command keyword (Go).

address is the hexadecimal address of the target processor's memory from which emulation is to begin.

The target processor's program counter is first set to the address indicated. MICE then starts the real-time emulation of the target processor. If no address is specified, program emulation starts at the current program counter. Aside from the help (?) and the attention (!) commands, entering a new MICE command stops the emulation.

Example: Resume program emulation of our 8048 program, whose current program counter is 0003H, from 0002H.

```
>R<cr>
CURRENT BANK = 0
 A   0   1   2   3   4   5   6   7   T   P   PC
BC BC BB  02  17  48  00  85  42  00  08  0003
>G 02<cr>
>R B0<cr>
 A   0   1   2   3   4   5   6   7   T   P   PC
13  13  13  02  17  48  00  85  42  00  08  0004
>
```

Example: Resume program emulation of our 8048 program from the current program counter of 0004H.

```
>R B0<cr>
 A   0   1   2   3   4   5   6   7   T   P   PC
13  13  13  02  17  48  00  85  42  00  08  0004
>G<cr>
>
```

Halt Emulation Command—H

H [address [, count]]

H is the command keyword (Halt).

address is the hexadecimal address of the target processor's memory where emulation is to be stopped.

count is a hexadecimal value from 0000H to FFFFH specifying the number of times the indicated address is to be encountered before emulation is stopped with 0000H indicating that emulation is to be stopped at the 65536th occurrence.

Emulation of the target processor is first stopped. If neither the address nor the count is specified, the current status is displayed. On the other hand, an address specification restarts the emulation from the current address until the address in the bus compares with the indicated halt address. MICE then stops the emulation and displays the new status. If the address encountered is in the mid-cycle of an instruction, that instruction is completed for processors that do not support single-cycle operation.

A count may also be specified to indicate the number of times the address is to be encountered before MICE stops the emulation. If no count is specified, the count is one, and the emulation stops at the first occurrence of the address.

Example: Stop the emulation of our 8048 target processor and display its current status.

```
>H<cr>
IFADDR   ADDRESS   DATA   STATUS
0002       0002     A8      S
>R B0<cr>
 A   0   1   2   3   4   5   6   7   T   P    PC
87  86  86  02  17  48  00  85  42  00  08  0002
>
```

Example: Restart emulation with the halt address in the middle of an instruction in our 8048 program.

```
>H 6
IFADDR   ADDRESS   DATA   STATUS
           0006      02      F
0002       0002     A8      S
>R B0 <cr>
 A   0   1   2   3   4   5   6   7   T   P    PC
88  87  87  02  17  48  00  85  42  00  08  0002
>
```

Because our 8048 can not be stopped in mid-cycle, emulation is stopped at the next instruction in the above example.

If the count value is too large or the address has not yet been encountered, entering an escape stops the emulation and ends the command.

Example: Restart emulation and stop at the 34th (22H) occurrence of address 95H, an address which our 8048 program never reaches.

```
>H,95 22<cr>
<esc>
>R B0<cr>
 A   0   1   2   3   4   5   6   7   T   P    PC
BC  BC  BB  02  17  48  00  85  42  00  08  0003
>
```

Cycle Step Command—C

C

C is the command keyword (Cycle).

MICE stops the processor and displays the current status of the target and waits for an input from the user. To advance the target processor to the next machine cycle, a line-feed should be entered; the new status is then displayed, and MICE again waits. Entering a line-feed repeats the entire sequence. If either a carriage-return or an escape is entered instead, the single-cycle mode of emulation is ended.

Example: Cycle step our 8048 program. Note that the processor does not begin emulation until the current status is displayed and until a line-feed is entered.

```
>R, B0<cr>
 A  0  1  2  3  4  5  6  7  T  P   PC
BC BC BB 02 17 48 00 85 42 00 08 0003
>C<cr>
IFADDR   ADDRESS   DATA   STATUS
0003       0003     A9    S<lf>
0004       0004     17    S<lf>
0005       0005     04    S<lf>
           0006     02    F
0002       0002     A8    S<lf>
0003       0003     A9    S<lf>
0004       0004     17    S<cr>
>R<cr>
CURRENT BANK = 0
 A  0  1  2  3  4  5  6  7  T  P   PC
BD BD BD 02 17 48 00 85 42 00 08 0004
>
```

In the above example, MICE did not pause at the third cycle because our target processor does not support single-cycle operation. Hence, the entire instruction is executed and status displayed before MICE again waits.

Example: The cycle step mode of emulation can be terminated before a single cycle has advanced by entering either a carriage-return or an escape in response to the first status line.

```
>R B0<cr>
 A  0  1  2  3  4  5  6  7  T  P   PC
BD BD BD 02 17 48 00 85 42 00 08 0004
>C<cr>
IFADDR   ADDRESS   DATA   STATUS
0004       0004     17    S<esc>
>RB0<cr>
 A  0  1  2  3  4  5  6  7  T  P   PC
BD BD BD 02 17 48 00 85 42 00 08 0004
>
```

Instruction Step Command—S

S [count]

S is the command keyword (Step).

count is a hexadecimal value from 0000H to FFFFH specifying the step interval between the status display with 0000H indicating that the display occurs every 65536 steps.

If no step count is specified, MICE assumes an instruction step count of one. In this mode of operation, MICE first displays the current status of the target and waits for an input from the user. To cause the target processor to execute the next instruction, a line-feed should be entered; the new status is then displayed, and MICE again waits. Entering a line-feed repeats the entire sequence where the new status is displayed after every instruction. If either a carriage-return or an escape is entered instead, the single step mode of emulation ended.

Example: Single step our 8048 program. Note that the processor does not begin emulation until the current status is displayed and until a line-feed is entered.

```
>R<cr>
CURRENT BANK = 0
 A  0  1  2  3  4  5  6  7  T  P   PC
BD BD BD 02 17 48 00 85 42 00 08 0004
>S<cr>
IFADDR   ADDRESS   DATA   STATUS
0004       0004      17    S<lf>
0005       0005      04    S<lf>
0002       0002      A8    S<lf>
0003       0003      A9    S<lf>
0004       0004      17    S<cr>
>R<cr>
CURRENT BANK = 0
 A  0  1  2  3  4  5  6  7  T  P   PC
BE BE BE 02 17 48 00 85 42 00 08 0004
>
```

In the above example, note that the display only has a status of "S" to indicate an instruction fetch cycle; no other status types are displayed. If a more careful examination of a program is required, first instruction step to the area in question, and then cycle step the trouble area for a more detailed display.

If an instruction step count other than one is entered, MICE immediately begins emulation from the current program counter until the count of the next instruction to be executed equals the number specified. The current status is displayed, and MICE waits for a line-feed before executing the next "count" instructions. Again, if either a carriage-return or escape is entered instead, the multi-step mode of emulation is ended. For large intervals, an escape entered while emulation is in process could be used to cause the emulation to be ended before the specified number of instructions are executed.

Example: The first command of this example single steps our 8048 program to better illustrate the status display for the multi-step command.

```
>R<cr>
CURRENT BANK = 0
 A   0   1   2   3   4   5   6   7   T   P    PC
BE BE BE 02 17 48 00 85 42 00 08 0004
>S 1<cr>
IFADDR   ADDRESS   DATA   STATUS
0004        0004      17    S<lf>
0005        0005      04    S<lf>
0002        0002      A8    S<lf>
0003        0003      A9    S<lf>
0004        0004      17    S<cr>
>R B0<cr>
 A   0   1   2   3   4   5   6   7   T   P   PC
BF BF BF 02 17 48 00 85 42 00 08 0004
>S 3<cr>
IFADDR   ADDRESS   DATA   STATUS
0002        0002      A8    S<lf>
0005        0005      04    S<lf>
0004        0004      17    S<lf>
0003        0003      A9    S<lf>
0002        0002      A8    S<esc>
>R B0<cr>
 A   0   1   2   3   4   5   6   7   T   P   PC
C3 C2 C2 02 17 48 00 85 42 00 08 0002
>
```

In summary, the instruction step command first executes "count−1" instructions before the first status display line, and then executes "count" instructions between subsequent displays.

Forward Trace Command—F

F trigger-address [, start-address [,end-address [, qualifier]...]]

F	is the command keyword (Forward).
trigger-address	is the hexadecimal address of the target processor's memory from which MICE begins recording trace information.
start-address	is the hexadecimal address of the target processor's memory indicating that only trace information whose addresses are greater-than or equal-to it are recorded.
end-address	is the hexadecimal address of the target processor's memory indicating that only trace information whose addresses are less-than or equal-to it are recorded.
qualifier	is a single alphabetic character {A|C|F|I|O|R|S|W} indicating the type of processor activity to be recorded A—interrupt acknowledge only C—all machine cycles F—program memory fetch only I—port input only O—port output only R—external/data memory read only S—instruction fetch only W—external/data memory write only

Forward tracing starts the emulation of the target and only begins the recording of target status information from the trigger-address until the MICE trace buffers are full, recording a total of 256 cycles; the processor is then stopped and the command is ended. If the target processor is in mid-cycle at the 256th recorded cycle for processors that do not support single-cycle operation, the cycles are completed but the status information in excess of the 256 cycles is not recorded. The recorded information can be examined by using the list trace command "L", to be described later.

If only the trigger-address is specified, all machine cycles after that address is encountered are recorded. An address range may be specified using start-address and end-address to record only status information whose addresses are within the defined range. If an end-address is specified, its value must be greater-than or equal-to the start-address or an error message is printed and the command ended. A start-address alone sets the end-address to the last address accessible by the target.

The specific type of information to be recorded is specified by the qualifiers entered. If no qualifiers are specified, all machine cycles are recorded. To enter any qualifyers at all, an address range must first be defined. Then, the qualifiers for the activity to be recorded are entered. Not all of the qualifiers are applicable to a target processor. For the particular processor being emulated, either consult the help (?) command or the corresponding guide in the appendices.

Example: Begin recording our 8048 program status from the trigger address of 0004H and only within the range 0000H through 0065H. The qualifiers entered is equivalent to all machine cycles, "C".

```
>F004,0,65 F  R,S W<cr>
TRACE BUFFER FULL
>
```

Depending on the specification, the elapse time before the buffers become full may be long. Entering an escape terminates the trace, yet retaining what was already recorded.

Example: Specify a trigger address of 0095H which our 8048 program never reaches.

```
>F 95<cr>
<esc>
>
```

Backward Trace Command—B

B trigger-address [, start-address [, end-address [, qualifier]...]]

B	is the command keyword (Backward).
trigger-address	is the hexadecimal address of the target processor's memory where emulation is to be stopped.
start-address	is the hexadecimal address of the target processor's memory indicating that only trace information whose addresses are greater-than or equal-to it are recorded.
end-address	is the hexadecimal address of the target processor's memory indicating that only trace information whose addresses are less-than or equal-to it are recorded.
qualifier	is a single alphabetic character {A|C|F|I||O|R|S|W} indicating the type of processor activity to be recorded A—interrupt acknowledge only C—all machine cycles F—program memory fetch only I—port input only O—port output only R—external/data memory read only S—instruction fetch only W—external/data memory write only

Backward tracing starts the emulation of the target and immediately begins the recording of target status information until the trigger-address is reached; the target processor is then stopped and the command is ended. If the target processor is in mid-cycle at the trigger-address for processors that do not support single-cycle operation, the cycles are completed but the status information after the trigger-address is not recorded. The recorded information can be examined by using the list trace command, "L", to be described later. If more than 256 cycles elapsed before the trigger-address, only the last 256 are recorded.

If only the trigger-address is specified, all machine cycles up to and including that address is encountered are recorded. An address range may be specified using start-address and end-address to record only status information whose addresses are within the defined range. If an end-address is specified, its value must be greater-than or equal-to the start-address or an error message is printed and the command ended. A start-address alone sets the end-address to the last address accessible by the target.

The specific type of information to be recorded is specified by the qualifiers entered. If no qualifiers are specified, all machine cycles are recorded. To enter any qualifiers at all, an address range must first be defined. Then, the qualifiers for the activity to be recorded are entered. Not all of the qualifiers are applicable to the target processor. For the particular processor being emulated, either consult the help (?) command or the corresponding guide in the appendices.

Example: Record our 8048 program status up to and including the trigger address of 0005H and only within the range 0000H through 00FFH.

```
> B 5 0 FF<cr>
BACKWARD TRACE STOP AT STEP 04
>
```

In the above example, the status line following our command line indicates that 04H cycles had been recorded. The counter is a hexadecimal value from 00H to FFH with 00H indicating that 256 cycles had been recorded.

Again, depending on the specification, the elapse time before the trigger address is encountered may be long. Entering an escape terminates the trace, yet retaining what was already recorded.

Example: Specify a trigger address of 0095H which our 8048 program never reaches.

```
> B 95<cr>
<esc>
>
```

List Trace Command—L

L [first-cycle [, last-cycle]]

L is the command keyword (List).

first-cycle is a hexadecimal value from 00H to FFH indicating the index to the first recorded cycle to be listed; the value 00H represents the first recorded cycle.

last-cycle is a hexadecimal value from 00H to FFH defining the last recorded cycle to be displayed.

The status information recorded during a forward or backward trace command is displayed using this command. An optional range can be entered to specify the amount and window of information to be listed. If no range is specified, the entire record is displayed. Entering first-cycle alone begins the display from that cycle to the last cycle recorded. Finally, a last-cycle specification causes only those cycles recorded to be listed. If a last-cycle is specified, its value must be greater-than or equal-to the first-cycle or an error message is printed and command ended. The command is also ended if the trace buffers are empty or if nothing is recorded within the specified range.

Example: List the first eleven cycles recorded in the trace buffer.

```
>L 0 A<cr>
IFADDR   ADDRESS   DATA   STATUS
0002       0002     A8      S
0003       0003     A9      S
0004       0004     17      S
0005       0005     04      S
           0006     02      F
0002       0002     A8      S
0003       0003     A9      S
0004       0004     17      S
0005       0005     04      S
           0006     02      F
0002       0002     A8      S
>
```

Entering an escape while MICE is displaying the trace aborts the display.

Example: List the trace cycles from the 5th cycle to the last cycle. The display is aborted before the listing is completed.

```
>L 4<cr>
IFADDR   ADDRESS   DATA   STATUS
           0006     02      F
0002       0002     A8      S
0003       0003     A9      S
0004       0004     17      S
0005       0005     04      S
           0006     02      F
0002       00<esc>
>
```

Utility Commands Involving a System

These commands are only applicable when MICE is connected to a host system. Program and data can be downloaded from a file on a host computer into the memory on the target system. Conversely, the information can also be uploaded from the target back onto the host.

Two types of loading formats, Intel and Tektronix, are recognized by MICE. Each format is described in detail in the following pages.

If the transfer rate between the host computer and MICE is greater than 1200 baud, the RS-232C handshake signals must be monitored to avoid overrun errors when downloading into the target system.

Download Command (Intel Format)—:

: load-record

: is the command keyword.

load-record is an Intel load record containing up to 32 bytes of program information.

Each record transferred contains the record type, length, memory load address, and checksum in addition to the data. Because of the size limitations in the MICE transfer buffers, each transfer is limited to 32 bytes of program data. The general format of a record, shown with spaces separating each field, is:

: ## aaaa tt dd...dd cc

where:

: is the keyword used to signal the start of record.

is a two ASCII hexadecimal value indicating the record length, the number of data bytes in the record. A record length of zero indicates the end-of-file.

aaaa is a four ASCII hexadecimal value indicating the program memory load address, the address at which the first byte is to be loaded. The successive data bytes are stored in successive memory locations.

tt is a two ASCII hexadecimal value representing the record type: 0—data record, 1—end record.

dd...dd is a two ASCII hexadecimal value per byte representation of the program.

cc is a two ASCII hexadecimal value representing the negative sum of the record. Beginning with the record length, "##", and ending with the checksum, "cc", the hexadecimal sum, taken two at a time, modulo 256 should be zero.

When MICE receives a ":", it reads the record length, "##", and determines the length of the entire record to be read. The rest of the record is read and verified. If the incoming checksum agrees with the computer checksum, MICE stores the data into the program memory of the target system and reprompts after the following acknowledgement is sent:

ACK LF CR NUL NUL NUL NUL NUL NUL

On the other hand, if the checksum does not agree, MICE also reprompts but the alternative negative acknowledgement response is sent:

NAK LF CR NUL NUL NUL NUL NUL NUL

Example: Dowload our 8048 program with an end-record into our target system using the Intel format.

```
>:060000002300A8A917046B<ack-sequence>
>:00000001FF<ack-sequence>
>
```

In the above example, the first load record

```
##      = 06H
aaaa    = 0000H
tt      = 00H
dd...dd = 23H, 00H, A8H, A9H, 17H, 04H
cc      = 6BH
```

with 06H+00H+00H+00H+23H+00H+A8H+A9H+17H+04H+6BH = 00H

For the end record,

```
##      = 00H
aaaa    = 0000H
tt      = 01H
cc      = FFH
```

with 00H+00H+00H+01H+FFH = 00H

Download Command (Tektronix Format)—/

/ load-record

/ is the command keyword.

load-record is a Tektronix load record containing up to 32 bytes of program information.

Each record transferred contains the record type, length, memory load address, and checksum in addition to the data. Because of the size limitations in the MICE transfer buffers, each transfer is limited to 32 bytes of program data. The general format of a record, shown with spaces separating each field, is:

/ aaaa ## ss dd...dd cc <cr>

where:

/ is the keyword used to signal the start of record.

aaaa is a four ASCII hexadecimal value indicating the program memory load address, the address at which the first byte is to be loaded. The successive data bytes are stored in successive memory locations.

is a two ASCII hexadecimal value indicating the record length, the number of data bytes in the record. A record length of zero indicates the end-of-file.

ss is a two ASCII hexadecimal value representing the sum of the preceding six digits, load address and record length.

dd...dd is a two ASCII hexadecimal value per byte representation of the program.

cc is a two ASCII hexadecimal value representing the sum of the digits comprising the data, modulo 256.

When MICE receives a "/", the input is read until a carriage-return terminates the line. The checksums are then computed and compared. If the incoming checksum agrees with the computed checksum, MICE stores the data into the program memory of the target system and reprompts after following acknowledgement response is sent:

ACK LF CR NUL NUL NUL NUL NUL NUL

On the other hand, if the checksum does not agree, MICE also reprompts but the alternate negative acknowledgement response is sent:

NAK LF CR NUL NUL NUL NUL NUL NUL

Example: Download our 8048 program with an end-record into our target system using the Tektronix format.

```
>/000006062300A8A9170436<cr>
<ack-sequence>
>/00000000<cr>
<ack-sequence>
>
```

In the above example, the first load record

```
aaaa    = 0000H
##      = 06H
ss      = 06H
dd...dd = 23H, 00H, A8H, A9H, 17H, 04H
cc      = 36H
```

where ss = 0H+0H+0H+0H+0H+0H+0H+6H = 06H, and
cc = 2H+3H+0H+0H+AH+8H+AH+9H+1H+7H+0H+4H = 36H

For the end record,

```
aaaa    = 0000H
##      = 00H
ss      = 00H
```

where ss = 0H+0H+0H+0H+0H+0H = 00H

Upload Command—U

U start-address, end-address [, format]

U	is the command keyword (Upload).
start-address	is the hexadecimal address of the target processor's program memory from which to begin the transfer of data.
end-address	is the hexadecimal address for the last address of the range in the target processor's program memory whose contents is to be transferred.
format	is a single alphabetic character {I|T} indicating the load record format to be used in the transfer I—Intel record format T—Tektronix record format

The program memory contents defined by the memory range start-address through end-address are transferred to the host system using either the Intel or Tektronix format. If neither format type is specified, the Intel format is used. Finally, the end-address must be greater-than or equal-to the start-address or an error message is sent and the command ended.

MICE first reads the data from the target system's memory and sends a maximum of 32 bytes to the host system using one of the two formats described in the previous pages. MICE then waits for an "ACK" from the host system before sending the next block. If an "ACK" is not received, the same block is retransmitted. If after five retries the transmission is still unsuccessful, the command is aborted, and an error message is sent to the host system before MICE prompts for the next command. The command can also be aborted by the host system when MICE receives the escape character.

Example: Upload our 8048 program using first the Intel then the Tektronix formats.

```
>U 0 6 I<cr>
:060000002300A8A917046B <ack>
:00000001FF <ack>
>U, 0, 6, T<cr>
/000006062300A8A9170436<cr>
<nak>
/000006062300A8A9170436<cr>
<ack>
/00000000
<ack>
>
```

In the above example, the host system did not acknowledge the first transmission when the Tektronix format was used. Hence, the first line is retransmitted until the "ACK" is received.

APPENDIX A
SUMMARY OF MICE COMMANDS

MICE Utility Commands

? —Help Command
?

! —Attention Command
!

Memory and Port Contents Commands

M—Memory Display/Change Command
M [type,] start-address [, end-address [, end address [, value]]

T—Memory Test/Transfer Command
T [type,] start-address, end-address, {SIMIdest-address}

A—Assemble into Memory Command
A start-address

Z—Disassemble Memory Command
Z start-address [, end-address]

I—Port Input Command
I port [, count [, duration]]

O—Port Output Command
O port, value-1 [, value-2 [, value-3 [, value-4]]]

Registers and Control Signals Commands

R—Register Display/Change Command
R [bank,] [register]

J—Program Counter Change Command
J address

D—Disable Control Signal Command
D {AIBIHIIINIT}

E—Enable Control Signal Command
E {AIBIHIIINIT}

X—Reset Target Processor Command
X [address]

Emulation and Trace Control Commands

G — Begin Emulation Command
G [address]

H — Halt Emulation Command
H [address [, count]]

C — Cycle Step Command
C

S — Instruction Step Command
S [count]

F — Forward Trace Command
F trigger-address [, start-address [, end-address [, {A|C|F|I|I|O|R|S|W}]...]]

B — Backward Trace Command
B trigger-address [, start-address [, end-address [, {A|C|F|I|I|O|R|S|W}]...]]

L — List Trace Command
L [first-cycle [, last-cycle]]

Utility Commands Involving a System

: — Download Command
: load-record

/ — Download Command
* / load-record

U — Upload Command
U start-address, end-address [, {I|T}]

APPENDIX B
HEXADECIMAL-DECIMAL CONVERSION

To find the decimal equivalent of a hexadecimal number, locate the hexadecimal number in the correct position and note the decimal equivalent. Add the decimal numbers.

To find the hexadecimal equivalent of a decimal number, locate the next lower decimal number in the table and note the hexadecimal number and its position. Subtract the decimal number from the table starting number. Find the difference in the table. Continue this process until there is no difference.

BYTE				BYTE			
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0	0	0
1	4096	1	256	1	16	1	1
2	8192	2	512	2	32	2	2
3	12288	3	768	3	48	3	3
4	16384	4	1024	4	64	4	4
5	20480	5	1280	5	80	5	5
6	24576	6	1536	6	96	6	6
7	28672	7	1792	7	112	7	7
8	32768	8	2048	8	128	8	8
9	36864	9	2304	9	144	9	9
A	40960	A	2560	A	160	A	10
B	45056	B	2816	B	176	B	11
C	49152	C	3072	C	192	C	12
D	53248	D	3328	D	208	D	13
E	57344	E	3584	E	224	E	14
F	61440	F	3840	F	240	F	15

APPENDIX C
ASCII CODES

Table C-1. ASCII Codes List

Hexadecimal	Decimal	Character	Hexadecimal	Decimal	Character
00	0	NUL	20	32	SP
01	1	SOH	21	33	!
02	2	STX	22	34	"
03	3	ETX	23	35	#
04	4	EOT	24	36	$
05	5	ENQ	25	37	%
06	6	ACK	26	38	&
07	7	BEL	27	39	'
08	8	BS	28	40	(
09	9	HT	29	41	)
0A	10	LF	2A	42	*
0B	11	VT	2B	43	+
0C	12	FF	2C	44	,
0D	13	CR	2D	45	—
0E	14	SO	2E	46	.
0F	15	SI	2F	47	/
10	16	DLE	30	48	0
11	17	DC1	31	49	1
12	18	DC2	32	50	2
13	19	DC3	33	51	3
14	20	DC4	34	52	4
15	21	NAK	35	53	5
16	22	SYN	36	54	6
17	23	ETB	37	55	7
18	24	CAN	38	56	8
19	25	EM	39	57	9
1A	26	SUB	3A	58	:
1B	27	ESC	3B	59	;
1C	28	FS	3C	60	<
1D	29	GS	3D	61	=
1E	30	RS	3E	62	>
1F	31	US	3F	63	?

Table C-1. ASCII Codes List

Hexadecimal	Decimal	Character	Hexadecimal	Decimal	Character
40	64	@	60	96	`
41	65	A	61	97	a
42	66	B	62	98	b
43	67	C	63	99	c
44	68	D	64	100	d
45	69	E	65	101	e
46	70	F	66	102	f
47	71	G	67	103	g
48	72	H	68	104	h
49	73	I	69	105	i
4A	74	J	6A	106	j
4B	75	K	6B	107	k
4C	76	L	6C	108	l
4D	77	M	6D	109	m
4E	78	N	6E	110	n
4F	79	O	6F	111	o
50	80	P	70	112	p
51	81	Q	71	113	q
52	82	R	72	114	r
53	83	S	73	115	s
54	84	T	74	116	t
55	85	U	75	117	u
56	86	V	76	118	v
57	87	W	77	119	w
58	88	X	78	120	x
59	89	Y	79	121	y
5A	90	Z	7A	122	z
5B	91	[	7B	123	{
5C	92	\	7C	124	|
5D	93	]	7D	125	}
5E	94	^	7E	126	~
5F	95	—	7F	127	DEL

Table C-2. ASCII Code Definitions

Abbreviation	Control	Meaning	Hexadecimal
NUL		NULL Character	00
SOH	A	Start of Heading	01
STX	B	Start of Text	02
ETX	C	End of Text	03
EOT	D	End of Transmission	04
ENQ	E	Enquiry	05
ACK	F	Acknowledge	06
BEL	G	Bell	07
BS	H	Backspace	08
HT	I	Horizontal Tabulation	09
LF	J	Line Feed	0A
VT	K	Vertical Tabulation	0B
FF	L	Form Feed	0C
CR	M	Carriage Return	0D
SO	N	Shift Out	0E
SI	O	Shift In	0F
DLE	P	Data Link Escape	10
DC1	Q	Device Control 1	11
DC2	R	Device Control 2	12
DC3	S	Device Control 3	13
DC4	T	Device Control 4	14
NAK	U	Negative Acknowledge	15
SYN	V	Synchronous Idle	16
ETB	W	End of Transmission Block	17
CAN	X	Cancel	18
EM	Y	End of Medium	19
SUB	Z	Substitute	1A
ESC		Escape	1B
FS		File Separator	1C
GS		Group Separator	1D
RS		Record Separator	1E
US		Unit Separator	1F
SP		Space	20
DEL		Delete	7F

APPENDIX D
WRITING A DRIVER PROGRAM

Driver programs for various computer systems have already been written. However, the driver programs available may not run or be suitable for your particular host computer.

When you have decided to write a driver program, the most important consideration is the availability of an RS-232C port. Without an RS-232C port, a driver program has no means to communicate with the MICE module. The programming language used to implement the driver program is not important as long as access to the console, the MICE communication port, and the file system are available.

Eight modules are necessary to establish a communications link between the console and the MICE:

InitMICE —initialize baud rate and parity of port to MICE
TestMICE —test if MICE sent a character
ReadMICE —read a character from MICE
WriteMICE—send a character to MICE
InitCons —initialize baud rate and parity of console
TestCons —test if console has a character
ReadCons —read a character from console
WriteCons —send a character to console

With these eight modules, we have the necessary ingredients to create a program replacing the simple terminal.

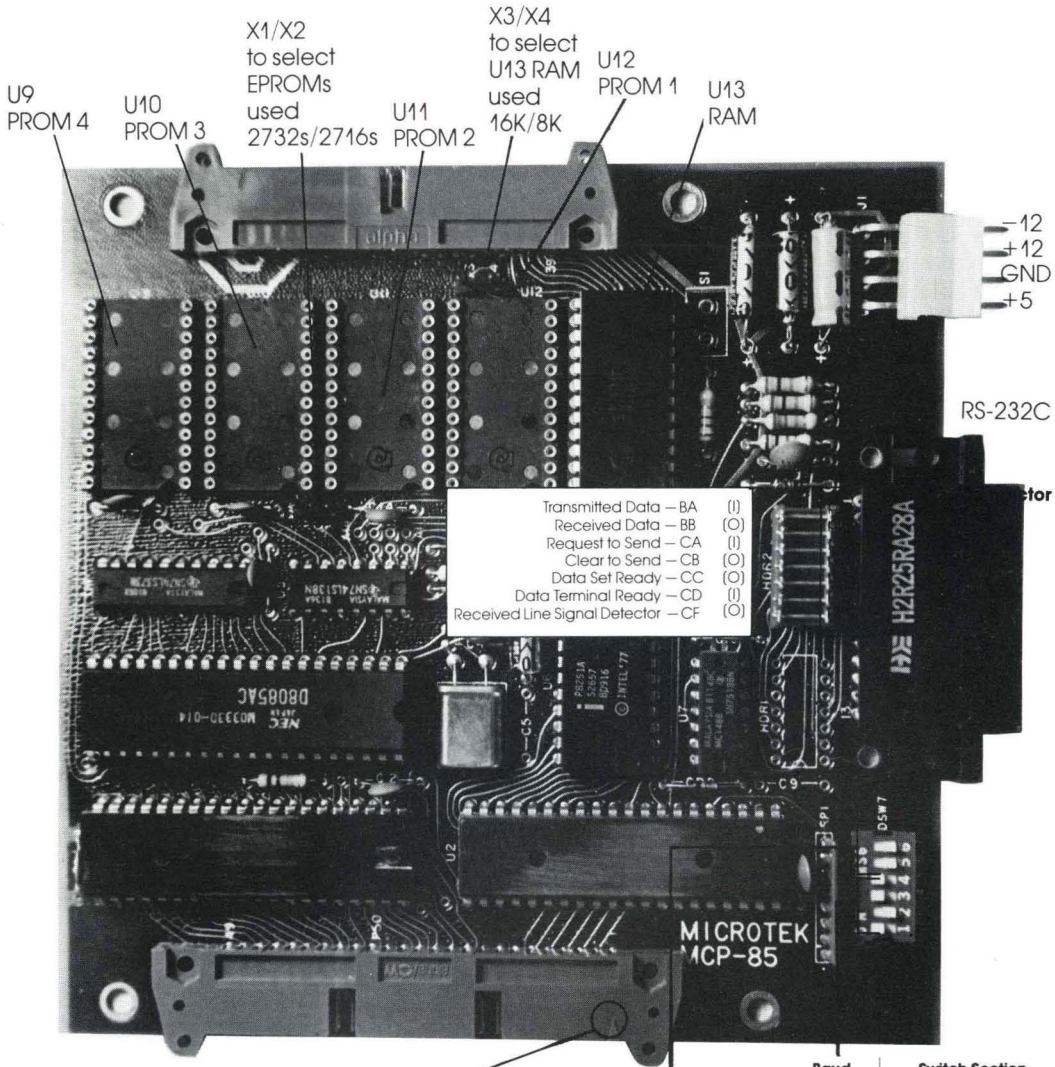
```
call InitCons
call InitMICE

do forever
   if TestCons then do
      Char = ReadCons
      call WriteMICE(Char)
   end

   if TestMICE then do
      Char = ReadMICE
      call WriteCons(Char)
   end

end
```

If the RS-232C handshake signals are not monitored, the baud rate selection for MICE must be less-than or equal-to the console baud rate to avoid overrun errors, where MICE is sending too fast for the display console. Refer to chapter 2 for configuring MICE to the appropriate baud rate, parity, and interface (DTE or DCE).

The interface to the file system requires similar but more complicated modules, which vary for different computer systems. The general concept is to have our program loop intercept the download and upload commands—not retransmitting them to MICE. The program then queries for the file to be transmitted or received. The appropriate command must then be created and transmitted to the MICE. It is important to remember that the maximum number of data bytes downloaded or uploaded at one time is limited to 32 bytes; the program may be required to reformat the data to one of the acceptable formats required by MICE.

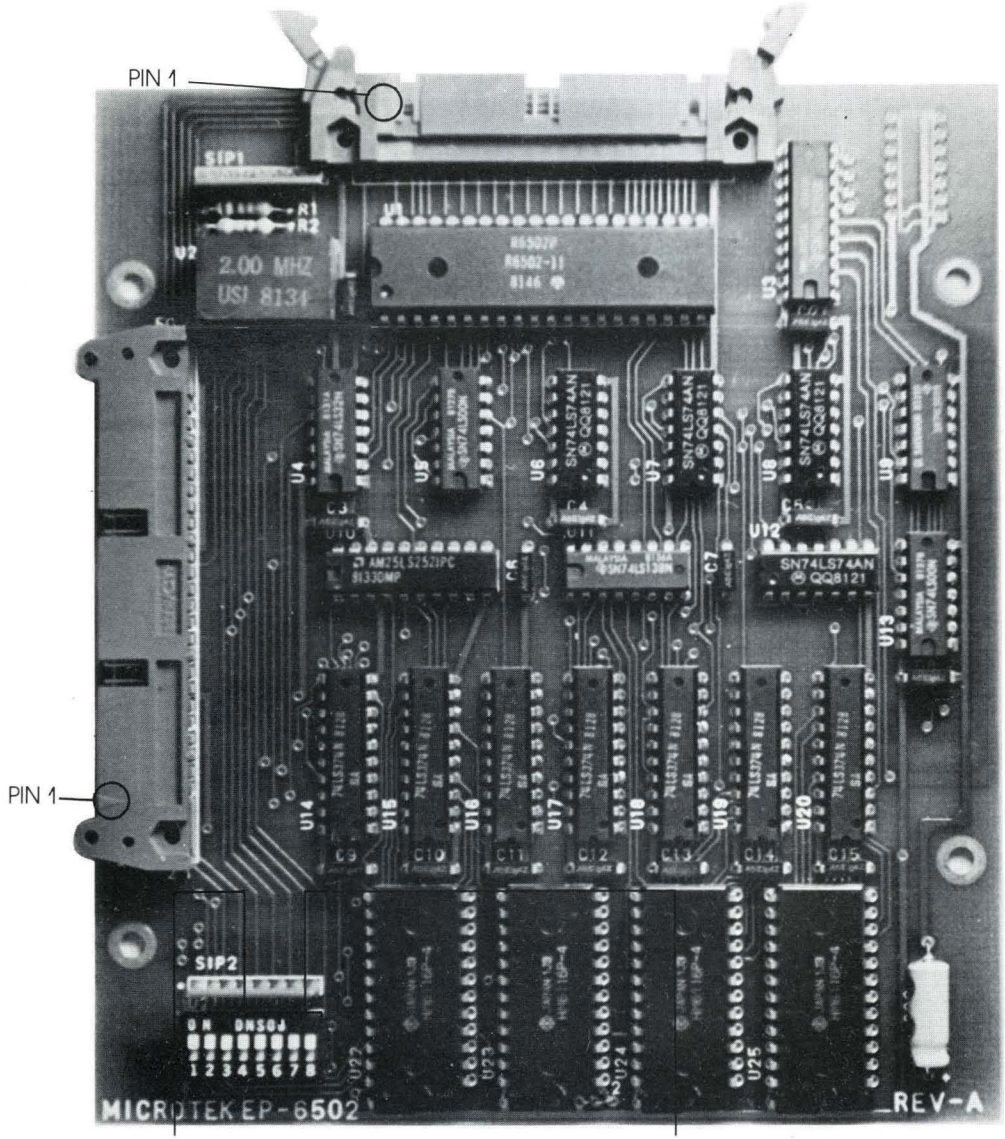
Finally, to add hardcopy for CRT consoles, an additional "write" to a remote printer can be added by calling the printer output module each time a "write" to MICE or a "write" to the console is performed.

APPENDIX E
CONTROLLER CARD



Mode	Switch Section 4	5	6
8 Data Bits	OFF		
7 Data Bits	ON		
Parity Enable		OFF	
Parity Disable		ON	
Even Parity			OFF
Odd Parity			ON

Baud Rate	Switch Section 1	2	3
110	OFF	OFF	OFF
150	ON	OFF	OFF
300	OFF	ON	OFF
600	ON	ON	OFF
1200	OFF	OFF	ON
2400	ON	OFF	ON
4800	OFF	ON	ON
9600	ON	ON	ON



Mode	Switch Section 2	3	4	5	6
0000H - 1FFFH	ON	ON	ON		
2000H - 3FFFH	ON	ON	OFF		
4000H - 5FFFH	ON	OFF	ON		
6000H - 7FFFH	ON	OFF	OFF		
8000H - 9FFFH	OFF	ON	ON		
A000H - BFFFH	OFF	ON	OFF		
C000H - DFFFH	OFF	OFF	ON		
E000H - FFFFH	OFF	OFF	OFF		
Memory Enable				ON	
Memory Disable				OFF	
Write Enable					ON
Write Protect					OFF

Clock Source	Section 7	8
From Target System	ON	OFF
MICE 2 MHz Crystal	OFF	ON
MICE Clock also Feeds Target	ON	ON

APPENDIX F
6502

To install a 6502 personality card in place of the current card, the cover to the MICE module must first be removed. With the power off, carefully remove the top, personality card, disconnecting the card from the ribbon cable that connects it to the bottom, control card. Next, remove the associated EPROMs from the socket locations U9, U10, U11, and U12 on the control card; being careful because the pins are easily bent.

Insert the EPROMs for the 6502 into their respective sockets as labeled on each EPROM. Currently, three 4K EPROMs (2732s) are used; the jumper selection X1/X2 must be set at X1.

Next, double check the jumper selection X3/X4. If an 8K RAM (4118) is used, the jumper must be set at X4; a 16K RAM (6116) requires the jumper to be at X3 instead.

Install the new personality card into the module, connecting the ribbon cable between the two boards; the arrows indicate pin 1 and must be aligned.

We are now ready to configure the switch settings on the personality card. The first section of the eight section switch is not used. The next five sections are used to configure the 8K of mappable memory. The five sections affect the memory as shown below.

	Switch Section				
Mode	**2**	**3**	**4**	**5**	**6**
0000H-1FFFH	ON	ON	ON		
2000H-3FFFH	ON	ON	OFF		
4000H-5FFFH	ON	OFF	ON		
6000H-7FFFH	ON	OFF	OFF		
8000H-9FFFH	OFF	ON	ON		
A000H-BFFFH	OFF	ON	OFF		
C000H-DFFFH	OFF	OFF	ON		
E000H-FFFFH	OFF	OFF	OFF		
Memory Enable				ON	
Memory Disable				OFF	
Write Enable					ON
Write Protect					OFF

The last two sections select the clock source going into pin 37 of the target processor as shown below.

	Section	
Clock Source	**7**	**8**
From Target System	ON	OFF
MICE 2 MHz Crystal	OFF	ON
MICE Clock also Feeds Target	ON	ON

Finally, connect the 40-pin ribbon cable if MICE is to replace the processor in the target system; again, the arrows must be aligned when attaching to the personality card, and the pins inserted as marked. When power is applied and the start-up message is not properly displayed, refer to chapter 2 for the items to be rechecked.

6502 Hardware

The 6502 has only one type of memory which serves as program memory, data memory, and I/O ports. The addressable memory is:

Memory: 0000H – FFFFH

To control the emulation of the processor, MICE shares the use of the RDY (pin 2) input—pulling the signal low to suspend emulation. Other control signals on the pins of the processor are affected by the disable and enable commands. The list of control characters for the commands are listed below with the controlled signal names and pin numbers.

I —interrupt request

IRQ/ -pin 4

N—non-maskable interrupt request

NMI/ -pin 6

Next, the list of possible qualifiers for trace activity to be recorded using the backward and forward tracing commands are:

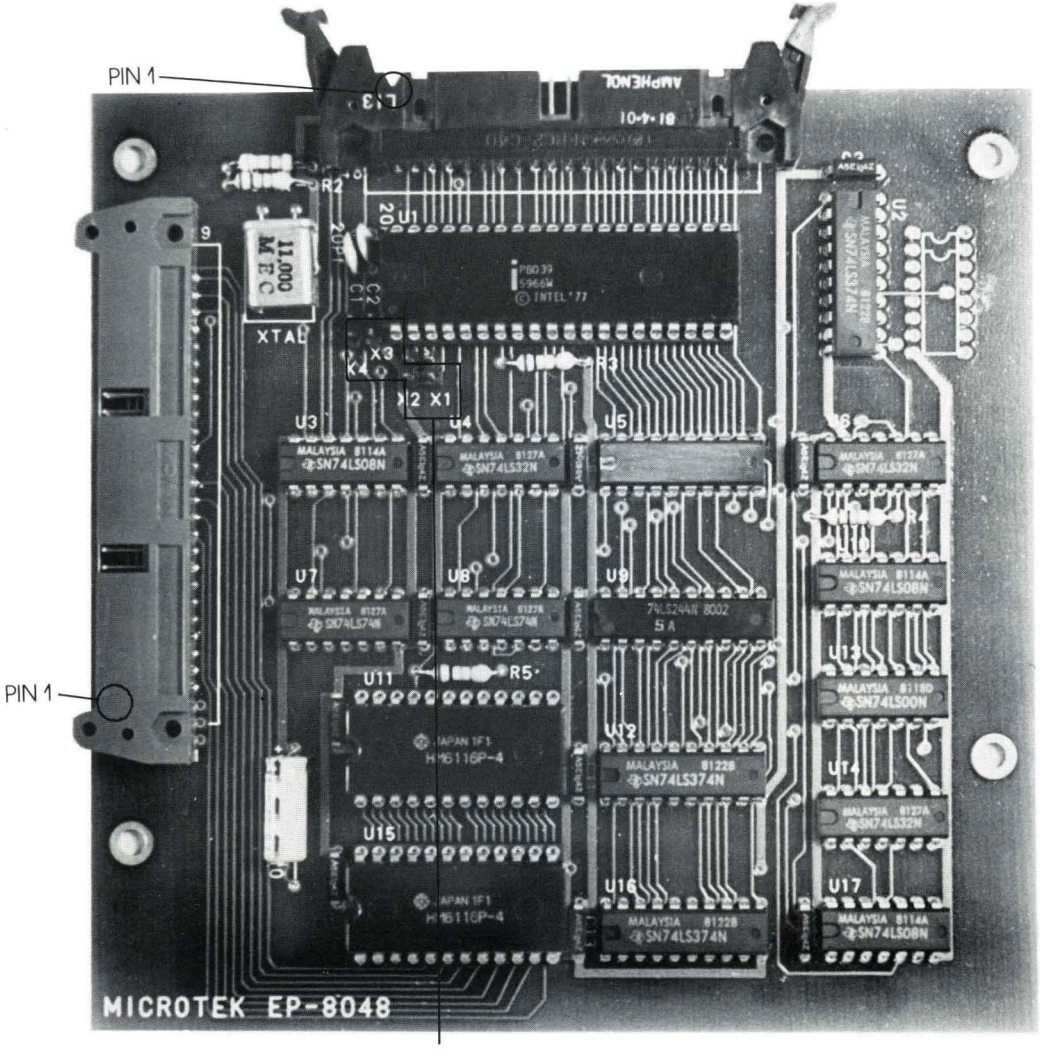
R —memory read
S —instruction fetch
W—memory write

The register set that can be displayed and changed by the "R" command is as follows:

Mnemonic	Register	Contents
A	Accumulator	8 bits
X	Index X	8 bits
Y	Index Y	8 bits
P	Program Status	8 bits (7 flags)
S	Stack Pointer	9-bit address

Bit 4 of the program status register is always displayed as a one by MICE. It can not be used to determine whether a break or an interrupt is in process.

Finally, single cycle operation is not supported by the 6502. The process can not be stopped on a "write" cycle during execution. If a "write" cycle is encountered, the processor is stopped on the next "read" cycle. Also, dummy cycles are recorded and displayed by the halt, cycle step, single step, forward trace, and backward trace commands.



Clock Source	Jumper X1	X2	X3	X4
From Target System	ON	ON	OFF	OFF
MICE 11 MHz Crystal	OFF	OFF	ON	ON

APPENDIX G
8048

The 8048 personality card is used to emulate the entire 8048 family (8048/8049/8050) of processors. To install an 8048 personality card in place of the current card, the cover to the MICE module must first be removed. With the power off, carefully remove the top, personality card, disconnecting the card from the ribbon cable that connects it to the bottom, control card. Next, remove the associated EPROMs from the socket locations, U9, U10, U11, and U12 on the control card; being careful because the pins are easily bent.

Insert the EPROMs for the 8048 into their respective sockets as labeled on each EPROM. Currently, three 4K EPROMs (2732s) are used; the jumper selection X1/X2 must be set at X1.

Next, double-check the jumper selection X3/X4. If an 8K RAM (4118) is used, the jumper must be set at X4; a 16K RAM (6116) requires the jumper to be at X3 instead.

Install the new personality card into the module, connecting the ribbon cable between the two boards; the arrows indicate pin 1 and must be aligned.

There are four jumpers designated as X1 through X4. The jumpers select the clock source going into pins 2 and 3 of the target processor. If the target system provides the clock oscillator, only X1 and X2 have jumpers. If there is no target system, the 11 MHz clock crystal on the personality card can be used by inserting the jumpers at X3 and X4 instead.

Finally, connect the 40-pin ribbon cable if MICE is to replace the processor in the target system; again, the arrows must be aligned when attaching to the personality card, and the pins inserted as marked. When power is applied and the start-up message is not properly displayed, refer to chapter 2 for the items to be rechecked.

8048 Hardware

The 8048 family of processors has three types of memory—program memory, "P"; internal data memory, "I"; and external data memory, "X". The difference between the members in the family is the amount of internal data memory and internal program memory available. The addressable memory and ports are:

	8048	8049	8050
Memory			
Program:	000H-FFFH	000H-FFFH	000H-FFFH
Internal Data:	00H -3FH	00H -7FH	00H -FFH
External Data:	00H -FFH	00H -FFH	00H -FFH
Ports			
	00H -02H	00H -02H	00H -02H
	04H -07H	04H -07H	04H -07H

Although the amount of program memory differs between family members, the addressable memory is the same.

To emulate the 8048 family, compatible processors without internal program memory are used. The number of processors the MICE module can emulate in the 8048 family depends on the processor used in the personality card.

Processor Used	Processor Emulated 8048	8049	8050
8035	YES	—	—
8039	YES	YES	—
8040	YES	YES	YES

However, the use of either the 8035 or the 8039 causes an error message during power-up or reset because 256 bytes of internal data memory is assumed and checked.

When MICE is powered-up or reset, the default processor type emulated is the 8048. To emulate a different family member, enter the processor type (8048/8049/8050) as a command in response to the prompt character ">".

Example: Display and change the processor type being emulated.

```
>!<cr>
EP8048
>8049<cr>
>!<cr>
EP8049
>
```

Although MICE allows any member of the 8048 family to be specified in the command, the results are not predictable if the proper processor is not used for the emulation.

To control the emulation of the processor, MICE shares the use of the SS/ (pin 5) input—pulling the signal low to suspend emulation. Other control signals on the pins of the processor are affected by the disable and enable commands. The list of control characters for the commands are listed below with the controlled signal names and pin numbers.

A—external memory access
    EA -pin 7
I—interrupt request
    INT/ -pin 6
T—T1 input
    T1 -pin 39

When the MICE module is powered-up or reset, the external program memory access control signal is disabled; the internal ROM of the 8048 family is emulated by the emulation memory on the personality card. If the control signal is enabled, the emulation memory only emulates the internal program memory; external program memory can be executed but can not be accessed by the memory display or change commands. Hence, information listed by the trace, halt, cycle step, and single step commands which represents the first cycle of any instruction from external program memory are not valid.

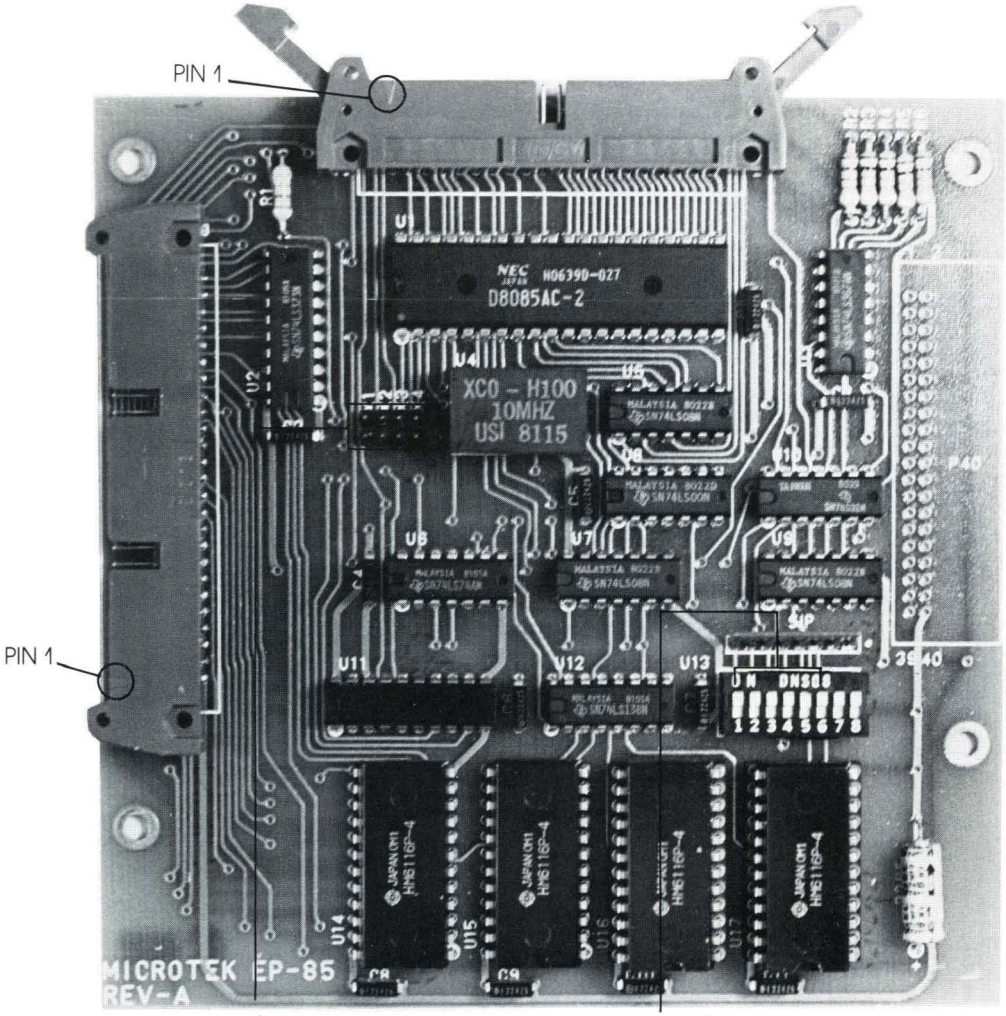
Next, the list of possible qualifiers for trace activity to be recorded using the backward and forward tracing commands are:

F—program memory fetch
R—memory read
S—instruction fetch
W—memory write

The 8048 family of processors has two sets of registers. The current bank is displayed and changed using the "R" command. To select bank 0, the characters "B0" must follow the "R" in the command specification; and "B1" if bank 1 is to be selected. The registers are listed in the following table; where alternates are available, it is listed under the mnemonic heading with an additional single quote.

Mnemonic	Register	Contents
A	Accumulator	8 bits
0 0'	General Purpose	8 bits
1 1'	General Purpose	8 bits
2 2'	General Purpose	8 bits
3 3'	General Purpose	8 bits
4 4'	General Purpose	8 bits
5 5'	General Purpose	8 bits
6 6'	General Purpose	8 bits
7 7'	General Purpose	8 bits
T	Time Counter	8 bits
P	Flags	8 bits (7 flags)

Finally, the 8048 family of processors do not support single cycle operation.



Clock Source	Jumper X1	X2	X3	X4
From Target System	ON	ON	OFF	OFF
MICE 10 MHz Crystal	OFF	OFF	ON	ON

Mode	Switch Section 1	2	3	4	5	6
Map as Memory	ON					
Map as Ports	OFF					
0000H - 1FFFH		ON	ON	ON		
2000H - 3FFFH		ON	ON	OFF		
4000H - 5FFFH		ON	OFF	ON		
6000H - 7FFFH		ON	OFF	OFF		
8000H - 9FFFH		OFF	ON	ON		
A000H - BFFFH		OFF	ON	OFF		
C000H - DFFFH		OFF	OFF	ON		
E000H - FFFFH		OFF	OFF	OFF		
Memory Enable					ON	
Memory Disable					OFF	
Write Enable						ON
Write Protect						OFF

APPENDIX H
8085

To install an 8085 personality card in place of the current card, the cover to the MICE module must first be removed. With the power off, carefully remove the top, personality card, disconnecting the card from the ribbon cable that connects it to the bottom, control card. Next, remove the associated EPROMs from the socket locations U9, U10, U11, and U12 on the control card; being careful because the pins are easily bent.

Insert the EPROMs for the 8085 into their respective sockets as labeled on each EPROM. Currently, four 2K EPROMs (2716s) are used; the jumper selection X1/X2 must be set at X2. If 4K EPROMs (2732s) are used, the jumper must be set at X1 instead.

Next, double check the jumper selection X3/X4. If an 8K RAM (4118) is used, the jumper must be set at X4; a 16K RAM (6116) requires the jumper to be at X3 instead.

Install the new personality card into the module, connecting the ribbon cable between the two boards; the arrows indicate pin 1 and must be aligned.

We are now ready to configure the jumpers and switch settings on the personality card. There are four jumpers designated as X1 through X4. The jumpers select the clock source going into pins 1 and 2 of the target processor. If the target system provides the clock oscillator, only X1 and X2 have jumpers. If there is no target system, the 10 MHz clock crystal on the personality card can be used by inserting the jumpers at X3 and X4 instead.

Six sections of the eight section switch are used to configure the 8K of mappable memory; the last two sections being unused. The six sections affect the memory as shown below.

	Switch Section					
Mode	**1**	**2**	**3**	**4**	**5**	**6**
Map as Memory	ON					
Map as Ports	OFF					
0000H - 1FFFH		ON	ON	ON		
2000H - 3FFFH		ON	ON	OFF		
4000H - 5FFFH		ON	OFF	ON		
6000H - 7FFFH		ON	OFF	OFF		
8000H - 9FFFH		OFF	ON	ON		
A000H- BFFFH		OFF	ON	OFF		
C000H - DFFFH		OFF	OFF	ON		
E000H - FFFFH		OFF	OFF	OFF		
Memory Enable					ON	
Memory Disable					OFF	
Write Enable						ON
Write Protect						OFF

Finally, connect the 40-pin ribbon cable if MICE is to replace the processor in the target system; again, the arrows must be aligned when attaching to the personality card, and the pins inserted as marked. When power is applied and the start-up message is not properly displayed, refer to chapter 2 for the items to be rechecked.

8085 Hardware

The 8085 has only one type of memory which serves both as program and data memory. The addressable memory and ports are:

Memory: 0000H - FFFFH
Ports: 00H - FFH

However, during the I/O write and read cycles, the I/O port address is duplicated in the upper byte of the address. If port addresses are to be used as halt or trace addresses, the upper byte must also be specified in the command.

To control the emulation of the processor, MICE shares the use of the READY (pin 35) input—pulling the signal low to suspend emulation. Other control signals on the pins of the processor are affected by the disable and enable commands. The list of control characters for the commands are listed below with the controlled signal names and pin numbers.

H—bus hold request
  HLDA - pin 38
  HOLD - pin 39
I—interrupt request
  RST 7.5 - pin 7
  RST 6.5 - pin 8
  RST 5.5 - pin 9
  INTR - pin 10
T—trap request
  TRAP - pin 6

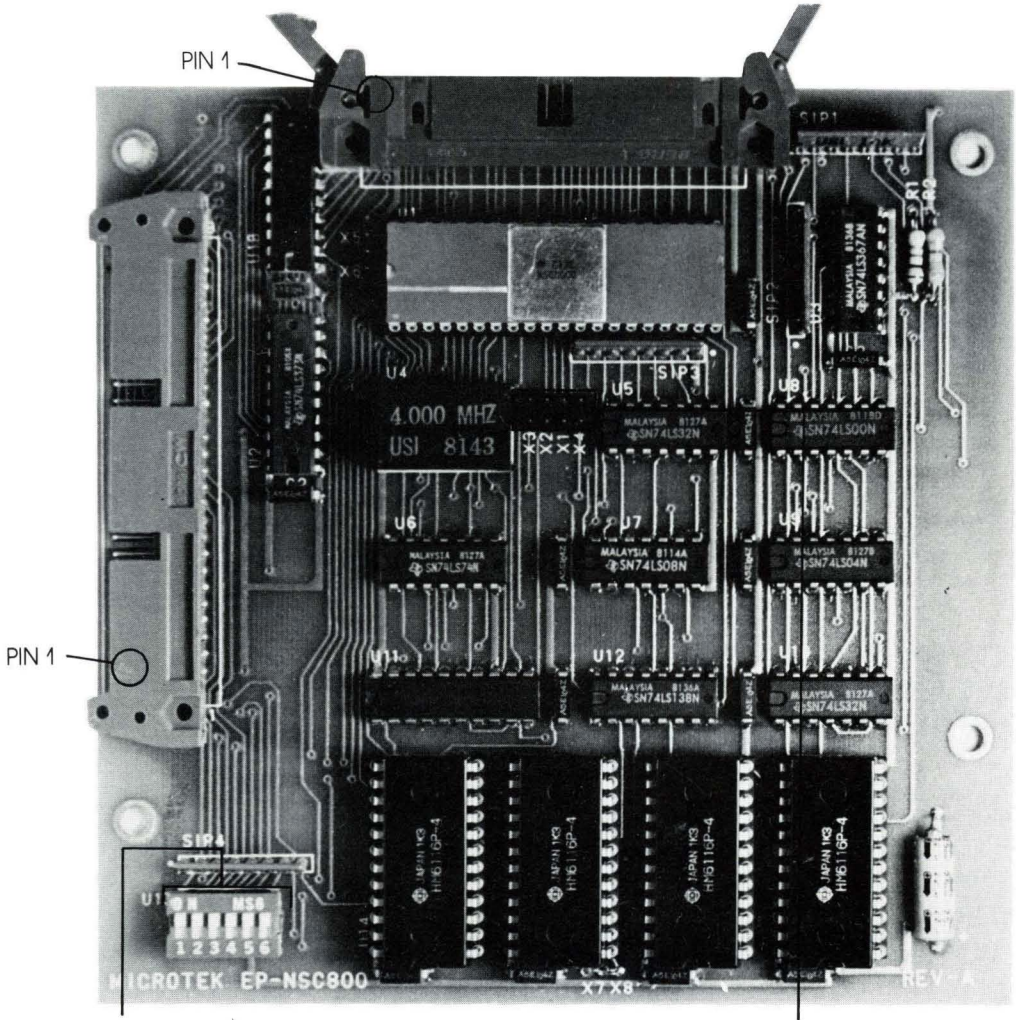
Next, the list of possible qualifiers for trace activity to be recorded using the backward and forward tracing commands are:

A—interrupt acknowledge
I—port input
O—port output
R—memory read
S—instruction fetch
W—memory write

The register set that can be displayed and changed by the "R" command is as follows:

Mnemonic	Register	Contents
A	Accumulator	8 bits
B	General Purpose	8 bits
C	General Purpose	8 bits
D	General Purpose	8 bits
E	General Purpose	8 bits
H	General Purpose	8 bits
L	General Purpose	8 bits
M	Interrupt Mask	8 bits
P	Program Status	8 bits (5 flags)
S	Stack Pointer	16-bit address

Finally, single cycle operation is supported by the 8085.



Mode	Switch Section 1	2	3	4	5	6
Map as Memory	ON					
Map as Ports	OFF					
0000H - 1FFFH		ON	ON	ON		
2000H - 3FFFH		ON	ON	OFF		
4000H - 5FFFH		ON	OFF	ON		
6000H - 7FFFH		ON	OFF	OFF		
8000H - 9FFFH		OFF	ON	ON		
A000H - BFFFH		OFF	ON	OFF		
C000H - DFFFH		OFF	OFF	ON		
E000H - FFFFH		OFF	OFF	OFF		
Memory Enable					ON	
Memory Disable					OFF	
Write Enable						ON
Write Protect						OFF

Clock Source	Jumper X1	X2	X3	X4
From Target System	ON	ON	OFF	OFF
MICE 4 MHz Crystal	OFF	OFF	ON	ON

APPENDIX I
NSC800

To install an NSC800 personality card in place of the current card, the cover to the MICE module must first be removed. With the power off, carefully remove the top, personality card, disconnecting the card from the ribbon cable that connects it to the bottom, control card. Next, remove the associated EPROMs from the socket locations U9, U10, U11, and U12 on the control card; being careful because the pins are easily bent.

Insert the EPROMs for the NSC800 into their respective sockets as labeled on each EPROM. Currently, three 4K EPROMs (2732s) are used; the jumper selection X1/X2 must be set at X1.

Next, double check the jumper selection X3/X4. If an 8K RAM (4118) is used, the jumper must be set at X4; a 16K RAM (6116) requires the jumper to be at X3 instead.

Install the new personality card into the module, connecting the ribbon cable between the two boards; the arrows indicate pin 1 and must be aligned.

We are now ready to configure the jumpers and switch settings on the personality card. There are six jumpers designated as X1 through X6. The jumpers X1 through X4 select the clock source going into pins 10 and 11 of the target processor. If the target system provides the clock oscillator, only X1 and X2 have jumpers. If there is no target system, the 4 MHz clock crystal on the personality card can be used by inserting the jumpers at X3 and X4 instead.

The jumpers X5 and X6 select the power supply voltage source going into pin 40 of the target processor. If the target system is to provide the power, only X5 has the jumper. For MICE to function properly, acceptable voltage levels from an external source are +4 VDC to +7 VDC. If the processor is to use the +5 VDC on the personality card, the jumper is on X6 instead.

A six section switch is used to configure the 8K of mappable memory.

	Switch Section					
Mode	**1**	**2**	**3**	**4**	**5**	**6**
Map as Memory	ON					
Map as Ports	OFF					
0000H-1FFFH		ON	ON	ON		
2000H-3FFFH		ON	ON	OFF		
4000H-5FFFH		ON	OFF	ON		
6000H-7FFFH		ON	OFF	OFF		
8000H-9FFFH		OFF	ON	ON		
A000H-BFFFH		OFF	ON	OFF		
C000H-DFFFH		OFF	OFF	ON		
E000H-FFFFH		OFF	OFF	OFF		
Memory Enable					ON	
Memory Disable					OFF	
Write Enable						ON
Write Protect						OFF

Finally, connect the 40-pin ribbon cable if MICE is to replace the processor in the target system; again, the arrows must be aligned when attaching to the personality card, and the pins inserted as marked. When power is applied and the start-up message is not properly displayed, refer to chapter 2 for the items to be rechecked.

NSC800 Hardware

The NSC800 has only one type of memory which serves both as program and data memory. The addressable memory and ports are:

Memory: 0000H - FFFFH
Ports: 00H - FFH

To control the emulation of the processor, MICE shares the use of the WAIT/(pin 38) input—pulling the signal low to suspend emulation. Other control signals on the pins of the processor are affected by the disable and enable commands. The list of control characters for the commands are listed below with the controlled signal names and pin numbers.

B or H—bus hold request
BACK/ - pin 35
BREQ/ - pin 36
I —interrupt request
RSTA/ - pin 22
RSTB/ - pin 23
RSTC/ - pin 24
INTR/ - pin 25
N or T—non-maskable interrupt request
NMI/ - pin 21

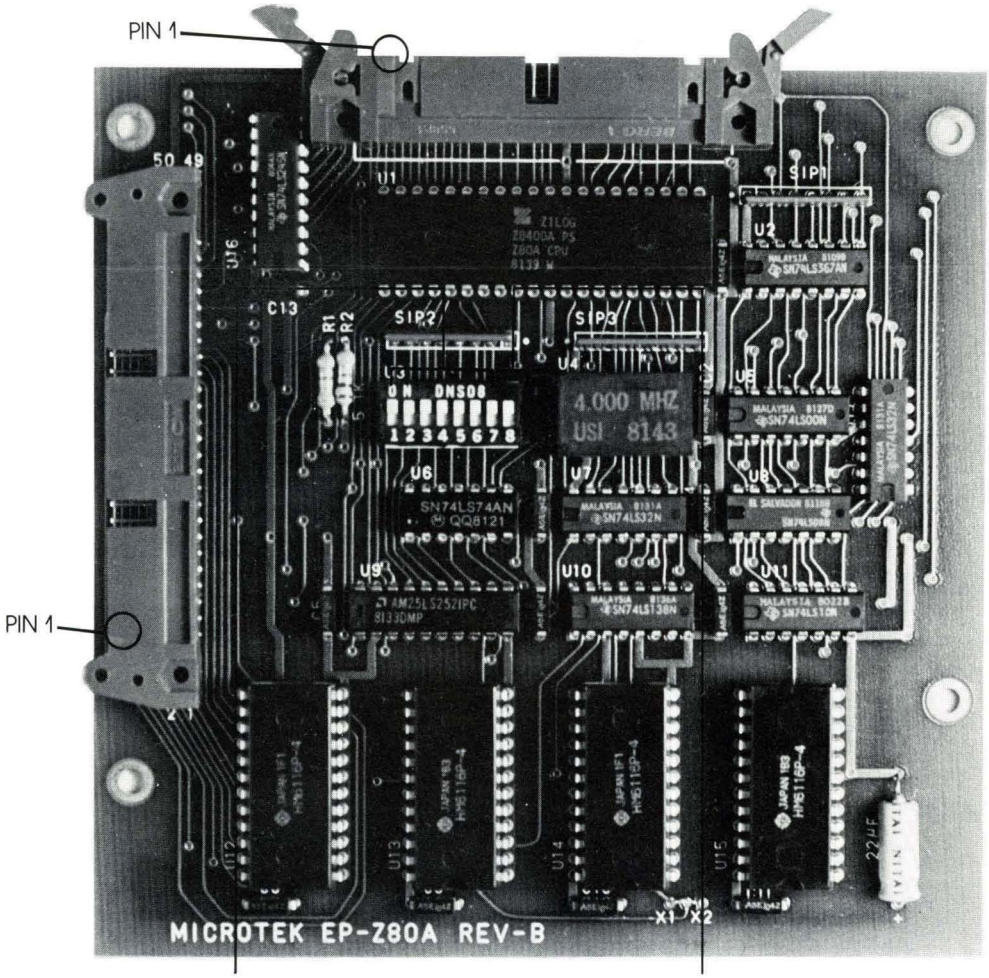
Next, the list of possible qualifiers for trace activity to be recorded using the backward and forward tracing commands are:

A —interrupt acknowledge
I —port input
O —port output
R —memory read
S —instruction fetch
W—memory write

The NSC800 has two sets of registers. The current set is displayed and changed using the "R" command; to select the alternate set, the character "P" must follow the "R" in the command specification. The registers are listed in the following table; where alternates are available, it is listed under the mnemonic heading with an additional single quote.

Mnemonic	Register	Contents
A A'	Accumulator	8 bits
B B'	General Purpose	8 bits
C C'	General Purpose	8 bits
D D'	General Purpose	8 bits
E E'	General Purpose	8 bits
H H'	General Purpose	8 bits
L L'	General Purpose	8 bits
I	Interrupt Mask	8 bits
F F'	Flags	8 bits (6 flags)
X	Index IX	16-bit address
Y	Index IY	16-bit address
S	Stack Pointer	16-bit address

Finally, single cycle operation is supported by the NSC800; but the dynamic memory refresh from the NSC800 is not supported by MICE.



Mode	Switch Section 1	2	3	4	5	6
Map as Memory	ON					
Map as Ports	OFF					
0000H - 1FFFH		ON	ON	ON		
2000H - 3FFFH		ON	ON	OFF		
4000H - 5FFFH		ON	OFF	ON		
6000H - 7FFFH		ON	OFF	OFF		
8000H - 9FFFH		OFF	ON	ON		
A000H - BFFFH		OFF	ON	OFF		
C000H- DFFFH		OFF	OFF	ON		
E000H - FFFFH		OFF	OFF	OFF		
Memory Enable					ON	
Memory Disable					OFF	
Write Enable						ON
Write Protect						OFF

Clock Source	Section 7	8
From Target System	ON	OFF
MICE 4 MHz Crystal	OFF	ON
MICE Clock also Feeds Target	ON	ON

APPENDIX J
Z80/Z80R

The difference between MICE-Z80 and MICE-Z80R is the support of dynamic memory refresh from the Z80. MICE-Z80 does not support the dynamic memory refresh, whereas MICE-Z80R does.

To install a Z80/Z80R personality card in place of the current card, the cover to the MICE module must first be removed. With the power off, carefully remove the top, personality card, disconnecting the card from the ribbon cable that connects it to the bottom, control card. Next, remove the associated EPROMs from the socket locations U9, U10, U11, and U12 on the control card; being careful because the pins are easily bent.

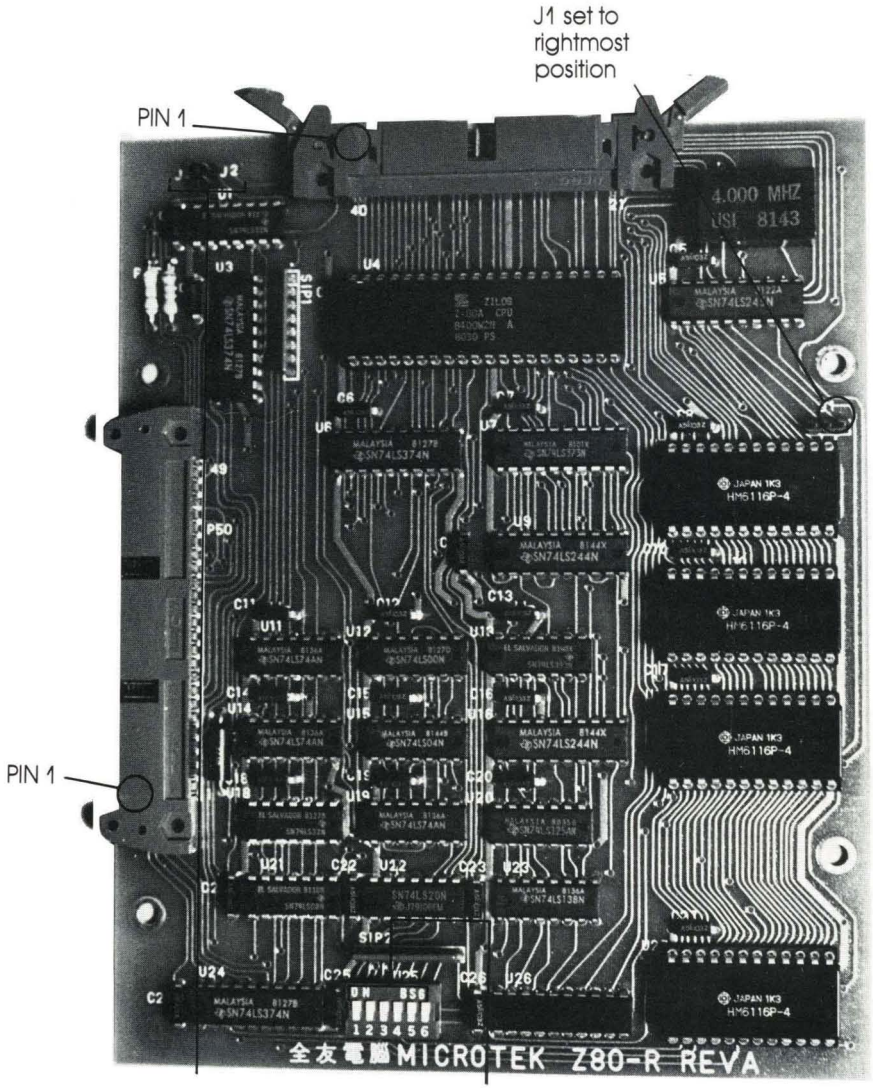
Insert the EPROMs for the Z80/Z80R into their respective sockets as labeled on each EPROM. Currently, three 4K EPROMs (2732s) are used by MICE-Z80, and four 4K EPROMs by MICE-Z80R; the jumper selection X1/X2 must be set at X1.

Next, double check the jumper selection X3/X4. If an 8K RAM (4118) is used, the jumper must be set at X4; a 16K RAM (6116) requires the jumper to be at X3 instead.

Install the new personality card into the module, connecting the ribbon cable between the two boards; the arrows indicate pin 1 and must be aligned.

We are now ready to configure the switch settings on the personality card. The first six sections of the eight section switch for the Z80 are used to configure the 8K of mappable memory; the Z80R only uses a six section switch. The six sections affect the memory as shown below.

	Switch Section					
Mode	**1**	**2**	**3**	**4**	**5**	**6**
Map as Memory	ON					
Map as Ports	OFF					
0000H - 1FFFH		ON	ON	ON		
2000H - 3FFFH		ON	ON	OFF		
4000H - 5FFFH		ON	OFF	ON		
6000H - 7FFFH		ON	OFF	OFF		
8000H - 9FFFH		OFF	ON	ON		
A000H- BFFFH		OFF	ON	OFF		
C000H - DFFFH		OFF	OFF	ON		
E000H - FFFFH		OFF	OFF	OFF		
Memory Enable					ON	
Memory Disable					OFF	
Write Enable						ON
Write Protect						OFF



Clock Source	Jumper J2	Jumper J3
From Target System	OFF	ON
MICE 4 MHz Crystal	ON	OFF
MICE Clock also Feeds Target	ON	ON

Mode	Switch Section 1	2	3	4	5	6
Map as Memory	ON					
Map as Ports	OFF					
0000H - 1FFFH		ON	ON	ON		
2000H - 3FFFH		ON	ON	OFF		
4000H - 5FFFH		ON	OFF	ON		
6000H - 7FFFH		ON	OFF	OFF		
8000H - 9FFFH		OFF	ON	ON		
A000H - BFFFH		OFF	ON	OFF		
C000H- DFFFH		OFF	OFF	ON		
E000H - FFFFH		OFF	OFF	OFF		
Memory Enable					ON	
Memory Disable					OFF	
Write Enable						ON
Write Protect						OFF

The last two sections for the Z80 selects the clock source going into pin 6 of the target processor as shown below.

Clock Source	Section 7	8
From Target System	ON	OFF
MICE 4 MHz Crystal	QFF	ON
MICE Clock also Feeds Target	ON	ON

In the Z80R, there are two jumpers designated as J2 and J3. The jumpers select the clock source as shown below.

Clock	Jumper J2	J3
From Target System	OFF	ON
MICE 4 MHz Crystal	ON	OFF
MICE Clock also Feeds Target	ON	ON

Both J2 and J3 are pre-inserted when shipped. If this is not the desired configuration, the appropriate adjustment must be made before power is applied. Although Z80/Z80R provides a 4 MHz clock, it can not directly drive PIOs and SIOs; the target system clock must be used instead.

The other jumper, designated as J1, should be set to the rightmost position.

Finally, connect the 40-pin ribbon cable if MICE is to replace the processor in the target system; again, the arrows must be aligned when attaching to the personality card, and the pins inserted as marked. When power is applied and the start-up message is not properly displayed, refer to chapter 2 for the items to be rechecked.

Z80/Z80R Hardware

The Z80/Z80R has only one type of memory which serves both as program and data memory. The addressable memory and ports are:

Memory: 0000H - FFFFH
Ports: 00H - FFH

To control the emulation of the processor, MICE-Z80 shares the use of the WAIT/ (pin 24) input; and MICE-Z80R shares BUSRQ/ (pin 25) input—pulling the signal low to suspend emulation.

MICE-Z80 is in a wait state when emulation is suspended. MICE-Z80R is in a hold state when emulation is suspended. Once in the hold state, the dynamic memory refresh is simulated and is generated once every four clock cycles.

Because MICE-Z80R is in a hold (bus acknowledge) state when emulation is suspended, the address, data, and some control signals on the bus are no longer valid. Similarly, the program counter values displayed by the halt, cycle step, single step, forward trace, and backward trace commands are the machine cycles just been executed by the processor. On the other hand, the value displayed by the register command is the machine cycle yet to be executed.

Other control signals on the pins of the processor are affected by the disable and enable commands. The list of control characters for the commands are listed below with the controlled signal names and pin numbers.

B or H—bus hold request
BUSAK/ - pin 23
BUSRQ/ - pin 25
I—interrupt request
INT/ - pin 16
N or T—non-maskable interrupt request
NMI/ - pin 17

Next, the list of possible qualifiers for trace activity to be recorded using the backward and forward tracing commands are:

A—interrupt acknowledge
I—port input
O—port output
R—memory read
S—instruction fetch
W—memory write

The Z80 has two sets of registers. The current set is displayed and changed using the "R" command; to select the alternate set, the character "P" must follow the "R" in the command specification. The registers are listed in the following table; where alternates are available, it is listed under the mnemonic heading with an additional single quote.

Mnemonic	Register	Contents
A A'	Accumulator	8 bits
B B'	General Purpose	8 bits
C C'	General Purpose	8 bits
D D'	General Purpose	8 bits
E E'	General Purpose	8 bits
H H'	General Purpose	8 bits
L L'	General Purpose	8 bits
I	Interrupt Mask	8 bits
F F'	Flags	8 bits (6 flags)
X	Index IX	16-bit address
Y	Index IY	16-bit address
S	Stack Pointer	16-bit address

For certain Z80 peripheral chips which recognize the "RETI" instruction to reset their interrupt requests or internal flags, the instruction must reside in the target system memory—not in the emulation memory.

Finally, single cycle operation is supported by the Z80/Z80R.