NEW MICE-II USER'S MANUAL

FOR 8-BIT INTEL

SERIES MICROPROCESSORS

-8085, 8048, 8031/8344, 8052/80C152, 80515/80535-

USER'S MANUAL FOR NEW MICE-II

8085
8048
8031/8344
8052/80C152
80515/80535

Doc No. I49-000114
Third Edition
August 1989

DISCLAIMER

This manual is subject to change without notice. Nothing herein shall be construed as a recommendation to use any product in violation of existing patents or other rights of third parties.

NEW MICE-II AND THIS MANUAL ARE COVERED BY THE LIMITED WARRANTY SUPPLIED WITH NEW MICE-II AND REPRODUCED IN APPENDIX O OF THIS MANUAL.

Printed in Taiwan, ROC
July 1989

Microtek International, Inc.
6 Industry Road East 3
Science-Based Industrial Park
Hsinchu, Taiwan, 30077, R.O.C.
TEL: (035)772155
TLX: 32169 MICROTEK
FAX: (035)772598

PREFACE

Microtek International Inc. proudly presents the award winning Micro-In-Circuit-Emulator, NEW MICE-II. You have made an excellent purchase and will soon benefit from the many advance features that Microtek designs into all of its products.

With the detailed instructions provided in this manual, you can rapidly set up and operate your new NEW MICE-II. The information provided in this user's manual is believed accurate and complete, but no responsibility is assumed for any error or omission.

Microtek is dedicated in providing leadership in the microcomputer-based industry. We continue to design new emulators with enhanced features, low cost and always geared toward the microprocessor designer's need. Thank you for choosing Microtek; and may we suggest that you also investigate the many other fine products that Microtek has to offer.

The sale of any Microtek product is subject to all Microtek Terms and Conditions of Sale and Sales Policies, copies of which are available upon request.

Microtek supplies a variety of hi-tech equipment to the microprocessor-based industry. For further information on other Microtek products and/or any other questions regarding this manual, please contact Microtek headquarters.

TABLE OF CONTENTS

APPENDICES

CHAPTER 1

GENERAL INTRODUCTION

1.1 Introduction To MICE

Microtek's versatile Micro-In-Circuit-Emulator (MICE) is a low-cost development tool that emulates most industry-standard microprocessors. It has set new standards for universal, high-performance emulation, at an exceptionally low cost-per-function.

Traditionally, microprocessor development was done on dedicated systems. The computer system had its complement of peripheral devices and sufficient mass storage to accommodate user programs, data, etc. A dedicated emulator was attached to the system to assist the designer in ensuring that both hardware and software were functioning properly.

Recently, a variety of general purpose computers have been designed for using in microprocessor development. Through the use of cross assemblers, code can be generated for the target system. However, these emulators are still dedicated to the particular computer.

With MICE, a third and more practical approach to microprocessor development is now available; one that uses fewer resources, performs most of the same functions, yet costs only a fraction compared with its predecessors. The MICE module is controlled via an RS-232C compatible interface. All software required to operate the MICE module are contained in EPROMs within the module. MICE can be operated by using a display

terminal only or in conjunction with a computer system. Different processors can be emulated by merely changing the personality card and associated EPROMs.

Some 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 space.

* Enabling and disabling of hardware control signals to the processor with console commands.

* Resident assembler and two-pass disassembler which assigns labels to subroutine and branch addresses.

* Built-in memory diagnostics and block memory transfer for target processor memory.

* Downloading and uploading of target program between MICE and host computer system.

* Help command that lists all commands along with the proper syntax.

1.2 Introduction to NEW MICE-II

1.2.1 Enhanced Features

NEW MICE-II is an enhanced member of the MICE family. Aside from retaining the original MICE features, NEW MICE-II has substantially been upgraded to include the following new features:

* High Performance Universal Emulation Memory (HUEM) with memory size selectable at 64K or 128K bytes (2 independent 64K settings) of static RAM for 8-bit NEW MICE-II models (user must specify if required for microcontroller versions).

* Super Universal Emulation Memory (SUEM) with memory size 256K bytes of static RAM for 8 bit NEW MICE-II models.

* 8/256 Kbytes block enable/disable capability for HUEM/SUEM.

* User-qualified trigger to specify start or end of tracing for source code and machine statuses. Up to 2048 cycles and 40 channels of signals may be recorded for program debug.

* Supports up to 8 hardware trace points for program debug.

* Real-time forward and backward trace for up to 2048 machine cycles.

* Two real-time breakpoints.

* Supports disassembly for both Instruction Step and List Trace commands.

* With built-in power supply and cooling fan.

1.2.2 New Breakpoint Processor (Optional)

NEW MICE-II may also be equipped with the new Breakpoint Processor (BPP) board at user's option. The BPP board will allow the NEW MICE-II to provide more advanced features as summarized below:

* Sophisticated breakpoint logic, with up to 120 new breakpoint constructs.

* Flexible trigger constructs that can define single events, multiple activities or external hardware signals.

* Two data breakpoints and two external hardware breakpoints.

* Breakpoint interval timer that displays the interval between initial trigger and emulation stops.

1.2.3 Major Hardware Overview

NEW MICE-II is a low-cost powerful emulation instrument basically equipped with the following boards:

Control Emulation Processor (CEP)
- also called personality board

Real-time Trace (RTT)

High Performance Universal Emulation Memory (HUEM) or Super Universal Emulation Memory (SUEM)
- also called memory board

BreakPoint Processor (BPP)

Remember that the CEP board can be interchanged to form any other NEW MICE-II. If more emulation memory is needed, additional HUEM boards may be added. And where more powerful emulation breakpoint or triggering capability is required, the BPP board is available.

1.2.4 Conclusion

At half the cost of competitive units, NEW MICE-II provides features not available with other emulators. Some of the advantages of NEW MICE-II over other emulators are:

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

versatility Emulation of a different processor only requires changing the personality board. Multiple design projects can be carried out concurrently, and the CEP board can be reused for different projects.

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

1.3 NEW MICE-II Operating Configurations

NEW MICE-II can be used in several configurations. The simplest configuration requires only an RS-232C compatible terminal as the display console and command entry device. A computer system can also be configured as the controlling device.

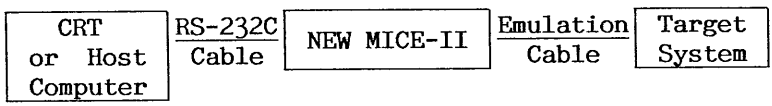


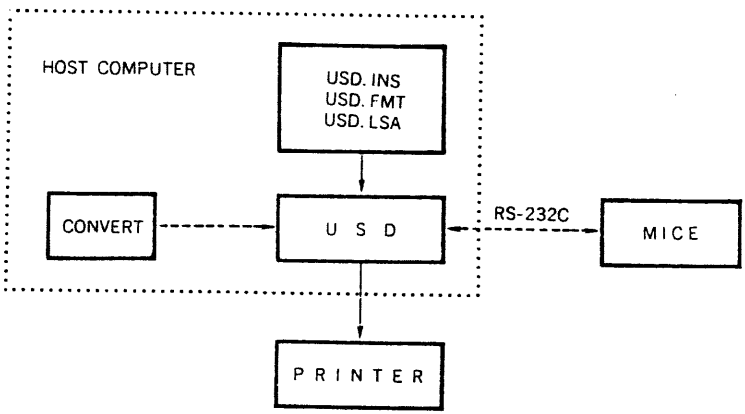
Figure 1-1
Configuring NEW MICE-II With an RS-232C
Compatible Terminal or Host Computer

When a computer system is interfaced with NEW MICE-II, a driver program must be resident in the system. Driver programs and symbolic translator drivers are available for various systems. (See Appendix E for a detailed listing.)

Microtek has access to various sources of driver programs and other software tools. For updated information contact your local Microtek representative.

1.3.1 Universal Symbolic Debugger

The USD (Universal Symbolic Debugger), is a sophisticated software package designed specifically to enhance the debugging and other capabilities of Microtek's MICE series of emulators. USD acts an interface between the MICE user, MICE, and the host computer system, to make the optimum use of the resources available on each. It enhances the range of development, emulation, analysis and debugging features of MICE to a new dimension and providing MICE users with a high quality and broad range of support to suit their various engineering applications.



* Universal
* Enhanced symbol processing capability
* File management
* Command file
* Logic state analysis
* Software performance analysis
* Sophisticated utility commands
* Fully supports object files, and MICE's

* Universal

Almost all popular cross-assembler/cross-compiler formats, such as 2500AD, IAR, intel OMF51/85/86, and MRI format symbol tables are accepted. MICE command for all type MICE are fully transparent.

* Enhanced symbolic processing capability

Symbol table file can be: loaded/listed/saved; symbols can also be created/deleted. Any address can be qualified to check for symbol equivalence.

* File management

Hex file can be downloaded to emulation/user memory, and data in memory can be uploaded to host computer. All MICE and USD commands can be logged as a macro command file; all debug commands as well as the command output can be logged as an output file. The logged output file is especially useful in detailed checking of sophisticated debug session.

* Command file

Unlimited command lines can be edited in a command file. 10 replaceable parameters are permitted; conditional sub-commands and loop block capability are provided.

* Logic state analysis

Trace buffer data, such as address bus, data bus, control, and spare signals can all be displayed in waveform.

* Software performance analysis

Module entry histogram:

Display the number of calls for specified program addresses, along with a bar graph showing the percentage of processor activity for each address.

Execution interval histogram:

Display the number of times a program routine executes within different timing frames, along with a bar graph showing the percentage of processor activity for each range.

*** Sophisticated utility commands**

Help for USD commands
Window facility is provided: foreground and background windows
Up to 16 command lines can be retained
DOS commands access

*** Fully supports object files, and MICE's**

Firmware version supported (for the firmware and higher version).

For NEW MICE-II

8085	V3.1 & later
8048	V3.2 & later
8031/8344	V3.1 & later
8052/80C152	V3.0 & later
80515/80535	V3.0 & later

* Host computers supported

IBM PC/XT/AT (MS-DOS 2.0)
NEC 9801 (MS-DOS 3.0)
SUN MICRO WORKSTATION (UNIX 2.0)
VAX/VMS 4.1
uVAX/ULTRIX

1.4 NEW MICE-II Applications

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

1. Inexpensive evaluation of new microprocessors without purchasing a special evaluation board or expensive development system.

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

3. NEW MICE-II's compact size, light weight, and rugged construction makes it an ideal field service instrument. Easily transported and set up in remote locations, NEW MICE-II can reduce downtime, provide on the spot diagnosis and resolution of field problems, avoiding customer inconvenience and expensive service delays. With its RS-232C interface, NEW MICE-II can be quickly interfaced with any compatible display terminal. Diagnostic programs can be generated using the resident assembler.

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

1.5 NEW MICE-II Definitions

MICE is a patented trade name for Microtek emulators and stands for Micro-In-Circuit-Emulator.

ICE cable is an In-Circuit-Emulator cable that joins NEW MICE-II to the target.

NEW MICE-IIH is a conventional MICE emulator module consisting of three interconnected printed circuit boards. These boards are:

1. Control Emulation Processor (CEP) board.
2. Real-time Trace (RTT) board.
3. High Performance Emulation Memory (HUEM) board.

NEW MICE-IIS is a conventional MICE emulator module consisting of four interconnected printed circuit boards. These boards are:

1. Control Emulation Processor (CEP) board.
2. Real-time Trace (RTT) board.
3. Super Universal Emulation Memory (SUEM) board.
4. Breakpoint Processor (BPP) Board.

CHAPTER 2

NEW MICE-II INSTALLATION PROCEDURES

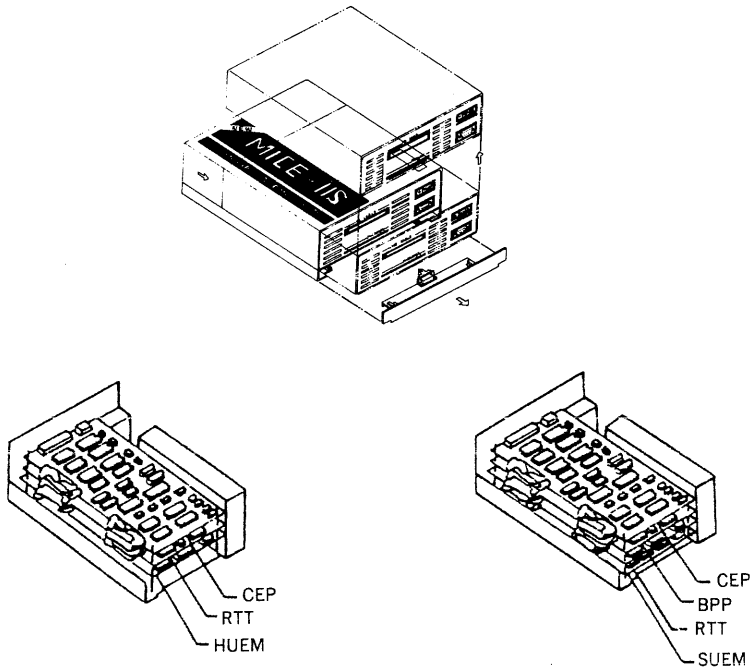
NEW MICE-II comes from the factory preset and completely assembled. ICE cables are also included which must be connected between NEW MICE-II and the target system.

2.1 Setting Up NEW MICE-II

Detailed instructions for setting up the NEW MICE-II module are as follows:

2.1.1 Opening the NEW MICE-II Case

NEW MICE-II is contained in a two piece high impact metal case. To open the case, hold the lower portion of the vertical end (rear) firmly with both hands and with your thumbs slide the top half forward about one half inch. This will separate the top from the bottom half.



New MICE-IIH

New MICE-IIS

Figure 2-1
Opening the NEW MICE-II Case

2.1.2 Changing the Personality Board (CEP) or Emulation/Verification CPU

1. To install a personality board in place of the current card, the cover of the NEW MICE-II case must first be removed. With power off, carefully remove the top personality card (CEP) from the NEW MICE-II module by disconnecting the two ribbon cable connectors which join the CEP board to the RTT board below, and then removing the six locking screws and six brass spacers.

   Install the new personality board into the module by refastening the brass spacers and locking screws to hold the replacement card permanently in position. Then reconnect the two ribbon cable connectors from the RTT board below to the CEP board; the arrows indicating pin 1 must be aligned.

2. Only one emulation CPU is supplied with MICE per the customer's specification. To emulate the other processor, replace the CPU indicated below:

CPU	Location	NEW MICE-II
8085	U29 on CEP	8085
8040	U33 "	8048
80C31F/others	U51 "	8031/8344
80515-E-A	U45 "	80515/80535
85C154VS/others	U14 on EPOD-8052	8052/80C152

3. NEW MICE-II 8031/8344 Emulation and Verification CPU Installation

   Depending on the target system, it may be necessary to concurrently use two CPU's on the CEP board. The

emulation CPU, which is always required regardless of target configuration, is located at U51; and the verification CPU, which is only required when performing internal program verification for the targets listed below, is located at U69.

a) The CEP-8031 uses an 80C31F (at U51) as the primary emulation CPU to emulate the 8031.

b) When emulating the 8032, use an 8032, 8052 or 8752 as the emulation CPU; and set $\overline{EA}$=0.

c) When emulating the 8344, use an 8344, 8044 or 8744 as the emulation CPU; and set $\overline{EA}$=0.

d) When performing internal program verification for the 8051/8751/8052/ 8752/8044/8744, put the target CPU on the CEP board at U69 to serve as a verification CPU. Remember that the verification CPU's "security bit" may be programmed to prohibit access to internal program memory.

4. NEW MICE-II 80515/83535 Emulation and Verification CPU Installation

Depending on the target system, it may be necessary to concurrently use two CPU's on the CEP board. The emulation CPU, which is always required regardless of the target configuration, is located at U45; and the verification CPU, which is only required when performing internal program verification, is located at U62.

a) The CEP-80515 uses an 80515-E-A (located at U45) as the emulation CPU to emulate both the 80515 and 80535.

b) When performing internal program verification for the 80515, put the target CPU on the CEP board at U62 to serve as a verification CPU.

5. NEW MICE-II 8052 Emulation and Verification CPU Installation

a) The NEW MICE-II 8052 (with EPOD-8052) uses an 85C154VS* as the primary emulation CPU to emulate the 8031, 8051, 8751, 8032, 8052, 8752, 80C154 and 83C154. An 8044/8344/8744 may be used as the primary emulation CPU for 8344. An 80C51FA/83C51FA/87C51FA may be used as the primary emulation CPU for 80C51FA.

* This device replaces standard built-in ROM of the 8052 with external EPROM, which is attached in a piggyback manner. In addition to the standard 256 bytes of on-chip RAM, the external EPROM is provided. (a maximum of 4K bytes is available when emulates the 8051/8751; 8K bytes for 8052/8752; 16K bytes for 83C154). Also note that timing and DC characteristics (including EPROM power) are identical with the 80C51.

b) When emulating the 8344, use an 8344, 8044 or 8744 as the emulation CPU; and set EA=0.

c) When emulating the 80C51FA, use an 80C51FA/83C51FA as the emulation CPU; and set EA=0.

d) When emulating the 80C152JA/83C152JA/80C152JB, replace EPOD with EPOD-80C152 and F/W V3.2 or later.

2.1.3 Internal Adjustments

Access to all adjustment switches is possible with the NEW MICE-II cover removed. The DIP switch on the personality board is readily accessed from the top; and the three DIP switches for memory board selection are located in the opening at the end of the NEW MICE-II case, which results when the cover is removed. Preset instructions are explained in the following sections of this chapter.

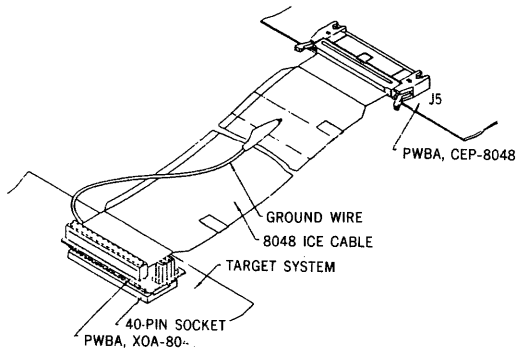
2.1.4 Connecting the ICE Cable

1. ICE Cable Installation for NEW MICE-II 8085

   The ICE (In-Circuit-Emulator) cable assembly consists of one flat-wire cable with a 40-pin connector at one end and an IC header at the other end. To install the ICE cable after setup has been completed, remove the cover to NEW MICE-II, and attach the 40-pin connector at position J5 on the CEP board. Then thread the IC header end through the rectangular opening in the NEW MICE-II case and slide the cover shut (section 2.1.7). Note that the ICE cable cannot be interchanged with other NEW MICE-II models.

2. ICE Cable Installation for NEW MICE-II 8048

   The 8048 ICE (In-Circuit-Emulator) cable assembly consists of one flat-wire cable with a 40-pin connector at one end and an IC header at the other end. To install the ICE cable after setup has been completed, remove the cover to MICE-II, and attach the 40-pin connector at position J5 on the CEP board. Then thread the IC header end through the rectangular opening in the MICE-II case and slide the cover shut (see section 2.1.7). Note that the ICE cable cannot be interchanged with other MICE-II models.



3. ICE Cable Installation for NEW MICE-II 8031/8344

The NEW MICE-II 8031/8344 ICE cable assembly consists of two flat-wire cables with dual 40-pin connectors at one end (CH) and a single 40-pin connector at the other end (TH8031). To install the ICE cable after setup has been completed, remove the NEW MICE-II cover and attach the end with dual 40-pin connectors at position J5 on the CEP board. Thread the IC header end through the rectangular opening in the NEW MICE-II case and slide the cover shut (section 2.1.7). Then attach the end with a single 40-pin connector to the 40-pin target processor socket on the target board. (Note that the ICE cable cannot be interchanged with other NEW MICE-II models.)

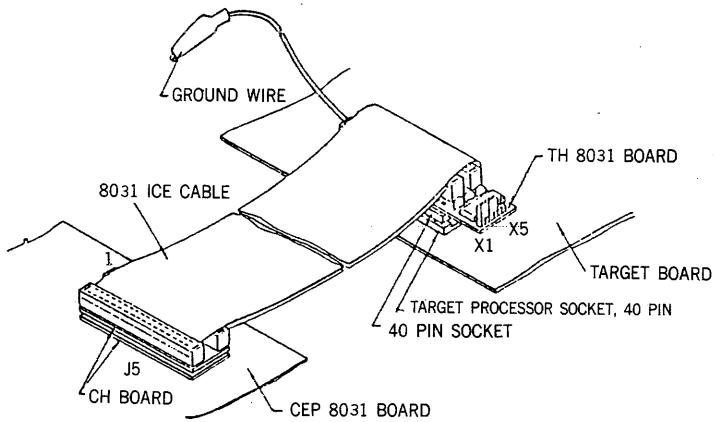


Figure 2-2
ICE Cable Installation for NEW MICE-II 8031/8344

4. ICE/Piggyback/EPOD Cable Installation for NEW MICE-II 8052/80C152

The installations for 8052/80C152 ICE/Piggyback/EPOD cables are illustrated as follows (Figure 2-3 through 2-8).

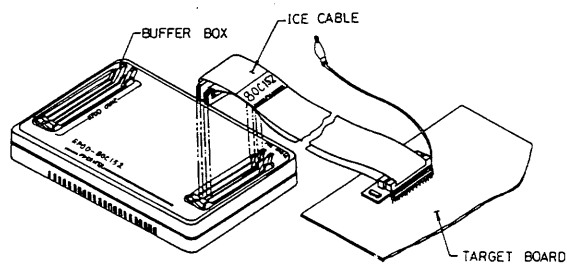


Figure 2-3
ICE Cable Installation for NEW MICE-II 80C152 DIP type

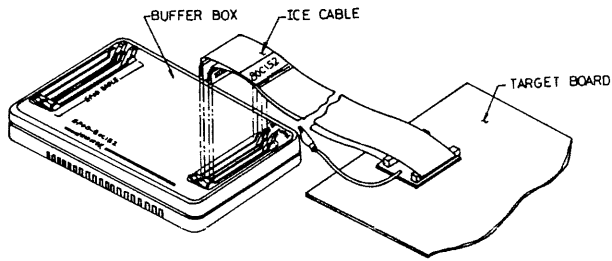


Figure 2-4
ICE Cable Installation for NEW MICE-II 80C152
PLCC type

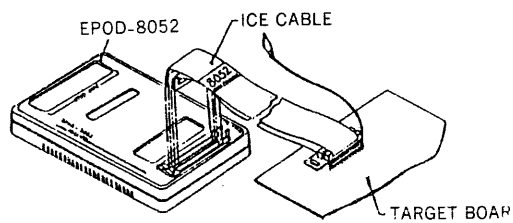


Figure 2-5
ICE Cable Installation for NEW MICE-II 8052
DIP type

NEW MICE-II 8052 PLCC type ICE cable assembly consists of two flat-wire connectors at one end and an IC header at the other end.

To install the ICE cable, first check the pin assignment orientation on the PLCC socket and plug the PLCC IC header (on the TCB-8052CP board) at PLCC socket, then attach the two connectors at positions mark "ICE CABLE" on the EPOD 8052(refer to Figure 2-4).

Note: If the orientation of TCB-8052CP is incorrectly connected on the PLCC socket, NEW MICE-II or target may result permanent damage when power-up.

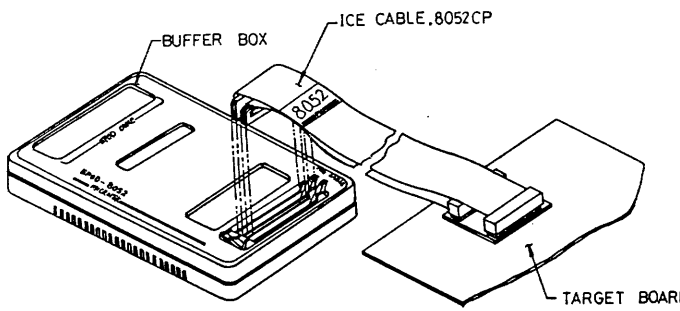


Figure 2-6
ICE Cable Installation for NEW MICE-II 8052 PLCC Type

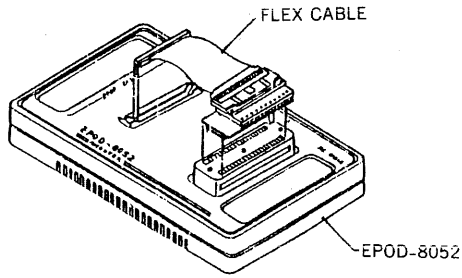


Figure 2-7
Piggyback Cable Installation for NEW MICE-II 8052

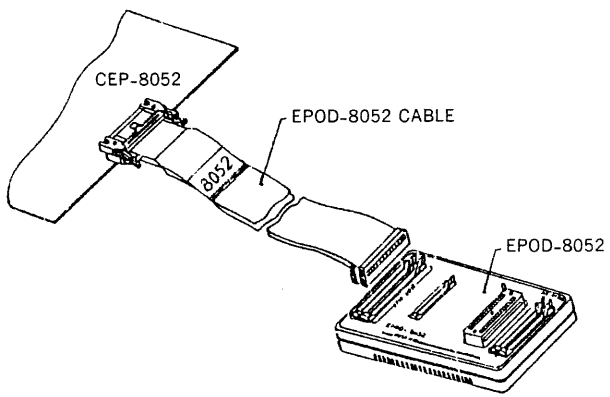


Figure 2-8
EPOD Cable Installation for NEW MICE-II 8052 (with EPOD 8052)

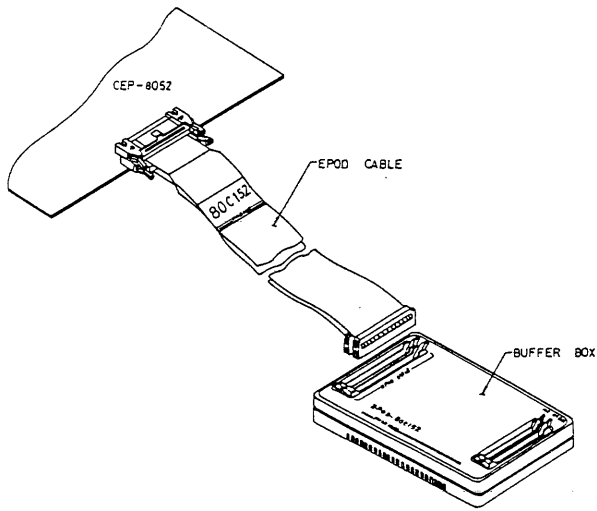


Figure 2-9
EPOD Cable Installation for NEW MICE-II
80C152 (with EPOD 80C152)

5. ICE Cable Installation for NEW MICE-II 80515C/535C*

The 80515C/535C ICE cable applies to CEP 80515, Rev: B and later. The ICE (In-Circuit-Emulator) cable assembly consists of two flat-wire cable with a 40-pin connector at one end and a PLCC (Plastic Leaded Chip Carrier) IC header at the other end. A Target Connect Board Header (TCBH-515) and a 68-pin PLCC socket are included with the package.

On MICE-II Side

Remove the cover of MICE-II, thread the connector end

---

* "C" indicates PLCC.

through the rectangular opening in the MICE-II case, attach the 40-pin connector at position J5 and J6 on the CEP board and slide the cover shut.

On Target Board Side

To install the ICE cable, first check the pin assignment orientation on the PLCC socket (pin-1 pointing to MICE-II side). Plug the PLCC IC header (on the TCBH-515 board) at PLCC socket. As for jumper wire settings (X1-3), refer to page 2-21.

Note that if the socket on the target is designed as PGA type, plug the PLCC socket on it prior to connecting the PLCC IC header.

Warning: If the orientation of TCBH-515 is incorrectly connected on the PLCC socket, MICE-II or target may result permanent damage when power-up

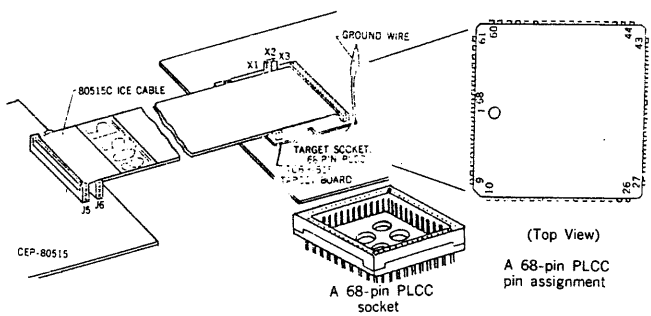


Figure 2-10
ICE Cable Installation for NEW MICE-II 80515C/535C

Note that the ICE cable cannot be interchanged with other NEW MICE-II models.

2.1.5 Connecting the External Trace Cable

The external trace cable shown below is used for recording external signals during trace operations. To monitor any external activity not automatically recorded by trace commands, the input cable must be connected from the trace point connector on the CEP board to the target. This connector consists of a 9-post stick header* on the CEP board designated X0-X7 from right to left, plus XG for ground. (* The CEP-8048 only provides a 5-post stick header designated X0-X3, plus XG.)

The monitored signals are called spare bits and support concurrent trace for the address, data and status bus. Any of these bits can be monitored by trace commands to provide hardware status for display (in the SPARE column) for List, Cycle Step and Instruction Step commands; where "1" represents a high (or floating) signal and "0" indicates a low signal.

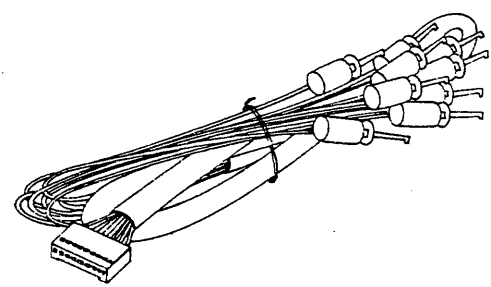


Figure 2-10
Signal Monitor Cable

2.1.6 Closing the NEW MICE-II Case

To close the cover, set it over the base leaving a 1/2 inch gap. Holding the base firmly in the front, use your thumbs to push the cover the remaining distance. This will force the locking tabs into position and firmly attach the top to the base.

2.2 NEW MICE-II Specifications

Mechanical specifications for NEW MICE-II are:

Width	26.0 cm	(10.24 in.)
Height	12.7 cm	(5.00 in.)
Length	33.0 cm	(13.00 in.)
Weight	5.0 kg	(11.00 lb.)

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

Operating temperature: 0°~50°C(32°~122°F)
Storage Temperature: -10°~65°C(14°~149°F)
Relative Humidity: 20 ~80%

NEW MICE-II equips one built-in power supply.

Power Consumption: AC100-120V~: 1.3A MAX
AC200-240V~: 0.65A MAX
47-63HZ

Voltage requirements: AC100-120V~ or
AC200-240V~
Factory-set

2.3 Communicating with NEW MICE-II

Whether NEW MICE-II is connected to a terminal or to a computer system, the connection between the controlling device and NEW MICE-II is across a programmable RS-232C compatible interface.

2.4 Control Emulation Processor Board (CEP) Setup

Other NEW MICE-II emulators can be formed simply by replacing the personality (CEP) board. If a change of personality boards is required, or modification to the factory preset conditions is necessary, the adjustment options are described as follows:

2.4.1 Interface Parameter Selection

On the CEP board locate the 6 position DIP switch (8085 - U34, 8048 - U17, 8031 - U5, 8052 - U5, 80515- U3). This switch sets the following communication modes which are only examined by the control processor during power-up and reset.

Switch Selection	Description
S1-S2-S3	Baud Rate
S4	7/8 Data Bits
S5	Disable/Enable Parity
S6	Odd/Even Parity

The number of stop bits is permanently set at two; and communication is full duplex. Each data frame consists of 1 start bit, 2 stop bits, 7/8 data bits and 1 parity bit if parity is enabled.

The transmission rate can be specified from 150-19200 baud by setting S1-S2-S3 of the DIP switch as follows:

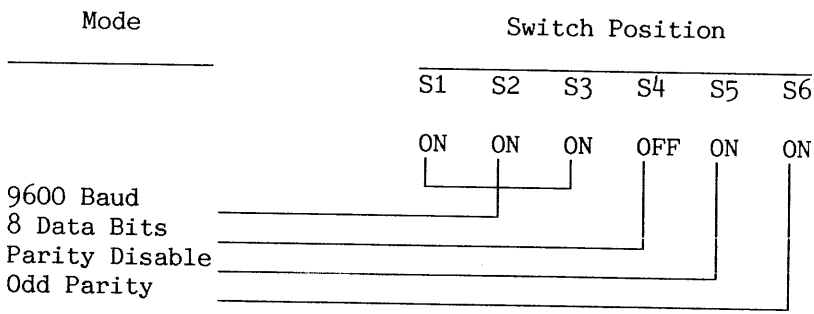
Baud Rate	Switch Section S1	S2	S3
150	ON	OFF	OFF
300	OFF	ON	OFF
600	ON	ON	OFF
1200	OFF	OFF	ON
2400	ON	OFF	ON
4800	OFF	ON	ON
9600	ON	ON	ON
19200	OFF	OFF	OFF *

* This setting is only effective for the following (or later) firmware versions: V3.0 - 8031/8344, 8052 and 80515 /80535, V3.1 - 8085, and V3.2 - 8048. For all previous versions, this switch setting indicates 110 baud.

The number of data bits and parity can be specified by setting S4, S5 and S6 of the DIP switch as follows:

Mode	Switch Section S4	S5	S6
8 Data Bits	OFF		
7 Data Bits	ON		
Parity Enable		OFF	
Parity Disable		ON	
Even Parity			OFF
Odd Parity			ON

New MICE-II Personality boards are shipped from the factory preset to 9600 baud, 8 data bits, and none parity. Switch position for this default selection is as follows:



Default setting

When parity is disabled, the odd/even parity switch (S6) section can be set to either position since it is ignored.

2.4.2 Clock Selection

1. Clock Selection for NEW MICE-II 8085

   The 10 MHz internal clock, external clock, or external crystal can be selected by placing jumpers on the personality board and on the X0A8085 adapter of the ICE cable as follows.

   The X0A8085 adapter is designed for an external clock or crystal. If the target uses an RC or RL circuit then a crystal of the corresponding value should be used to assure that the correct frequency is generated.

XOA8085
Adapter

	X8	X10	X1	X2
INT	O	C	O	O
EXT CLOCK	C	O	O	C
EXT CRYSTAL	C	O	C	O

C: Close
O: Open

Internal/External Clock Selection

2. Clock Selection for NEW MICE-II 8048

The target's clock or the on-board 11 MHz clock can be selected by placing jumpers on the personality board and XOA8048 adapter as follows:

XOA8048
Adapter

	X4	X5	X6	X7	X8	X9	X10	X1	X2
INT (CRYSTAL)	C	C	O	O	O	O	O	O	O
EXT (LC OSC.)	O	O	O	O	O	C	C	O	O
EXT CRYSTAL	O	O	C	C	C	O	O	C	C
EXT CLOCK	O	O	C	C	C	O	O	O	O

Internal/External Clock Selection For XOA REV. A, B

O: Open
C: Close

							XOA8048 Adapter				
	X4	X5	X6	X7	X8	X9	X10	X1	X2	X3	X4	X5
INT (CRYSTAL)	C	C	O	O	O	O	O	O	O	O	O	O
EXT (LC OSC.)	O	O	O	O	O	C	C	C	O	O	C	O
EXT CRYSTAL	O	O	C	C	C	O	O	O	C	C	O	C
EXT CLOCK	O	O	C	C	C	O	O	O	O	O	C	O

Internal/External Clock Selection For XOA REV. C

O: Open
C: Close

Note that the ICE Cable for XOA Rev: C has major changes than Rev: B.

3. Clock Selection for NEW MICE-II 8031/8344

The target's clock or the on-board 12MHz clock can be selected by placing jumpers on the personality board as follows:

					TH8031* Adapter			
		X8	X9	X10	X1	X2	X3	X5
INT		O	O	C	X	X	X	X
EXT CLOCK	XTAL1	O	C	O	O	O	O	C
	XTAL2	C	O	O	O	C	O	O
EXT CRYSTAL		C	O	O	C	O	C	O

Internal/External Clock Selection C: Close
O: Open
X: Don't Care

*The TH8031 Adapter is located on the target side of the ICE cable.

4. Clock Selection for NEW MICE-II 80515/80535

The target's clock or the on-board 12MHz clock can be selected by placing jumpers on the personality board as follows:

			*OAB-515 Adapter or TCBH-515		
	X8	X9	X1	X2	X3
Internal Clock	C	O	X	X	X
External Clock	O	C	O	C	O
External XTAL	O	C	C	O	C

Internal/External Clock Selection C: Close
O: Open
X: Don't Care

2.4.3 Ready Signal Selection for NEW MICE-II 8085

The target ready signal can be selected by removing the jumper at X12; or if NEW MICE-II emulation memory is selected, the on-board ready signal can be used by connecting a jumper at X12.

---

* The OAB-515 adapter (for 80515G/535G ICE Cable) or TCBH-515 (for 80515C/535C ICE Cable) is located on the target side of the ICE cable.

2.4.4 Port Setting for Microcontroller/Microcomputer Selection

1. Microcontroller/Microcomputer Selection for NEW MICE-II 8031/8344

   Operation is supported for standard microcontroller I/O functions or for microcomputer Read/Write functions to external data memory according to the setting of Port 3 bits 6 & 7. Selection can be made to use P3.6 for I/O or WR and P3.7 for I/O or RD by setting jumpers on the personality board as follows:

	X11	X12	X13	X14
I/O	C	O	C	O
$\overline{RD}/\overline{WR}$	O	C	O	C

C: Close
O: Open

Microcontroller/Microcomputer Selection

2. Microcontroller/Microcomputer Selection for NEW MICE-II 80515/80535

   Operation is supported for standard microcontroller I/O functions or for microcomputer Read/Write functions to external data memory according to the setting of Port 3 bits 6 and 7. Selection can be made to use P3.6 for I/O or Write and P3.7 for I/O or Read by setting jumpers on the personality board as follows:

	X10	X11	X12	X13
I/O	C	O	C	O
R/W	O	C	O	C

C: Close
O: Open

Microcontroller/Microcomputer Selection

2.5 High Performance Universal Emulation Memory Board (HUEM) Setup

There are four DIP switches located at the front of the HUEM board (bottom board in the NEW MICE-II module); U14/U28 (enabled for both the standard 64K byte version and the optional 128K byte version) and U36/U43 (enabled only for the optional 128K byte version). The individual keys on the switches are designated S1 through S8 or S10. (Refer to the placement chart on page F-2 for the board location of these switches.)

There are two separate 64K memory banks in the HUEM. U14 and U28 set bank 1; U36 and U43 set bank 2. (Bank 2 is only populated for the optional 128K byte version.)

The DIP switches are set up per the following instructions:

2.5.1 U14 and U36: Emulation Memory Enable/Disable

The user can select any memory block of 8K bytes to disable or enable depending on the amount of emulation memory needed. Set individual keys ON for enable and OFF for disable.

S1	S2	S3	S4	S5	S6	S7	S8
Lowest 8K	.....						**Highest 8K**

Position S9 and S10 are used as follows:

Mode	S9	S10
Write Enable	ON	
Write Protect	OFF	
Memory Enable		OFF
Memory Disable		ON

2.5.2 U28 and U43: Memory Segment Select

Memory segments may be selected within 16M bytes by setting U28 and U43 as follows:

Segment	Start Address	S1	S2	S3	S4	S5	S6	S7	S8
0K	0H	ON	ON	ON	ON	ON	ON	ON	ON
64K	10000H	OFF	ON	ON	ON	ON	ON	ON	ON
128K	20000H	ON	OFF	ON	ON	ON	ON	ON	ON
"	"				"				
"	"				"				
16320K	FF0000H	All OFF							

Note: If bank 1 and bank 2 have the same start address (i.e. U28 and U43 have the same key setting), HUEM will select bank 1 only and disregard the U43 key setting for bank 2. For 64K HUEM, only bank 1 memory is populated.

2.5.3 Factory Preset

HUEM is preset at the factory with the following switch positions (0-128K bytes):

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
U14,U36	ON	ON	ON	ON	ON	ON	ON	ON	ON	OFF

In this configuration all blocks (64K byte x 2) of emulation memory are enabled and write enabled.

U28	S1	S2	S3	S4	S5	S6	S7	S8
	ON	ON	ON	ON	ON	ON	ON	ON

All U28 keys are in the ON position. In this configuration, thebank 1 memory segment start address is 0000H.

U43	S1	S2	S3	S4	S5	S6	S7	S8
	OFF	ON	ON	ON	ON	ON	ON	ON

Only S1 is OFF. In this configuration, the bank 2 memory segment start address is 10000H.

Setting U14, U28, U36 and U43 completes setup of the HUEM board.

2.6 Memory Selection for Microcontrollers

2.6.1 Memory Selection for NEW MICE-II 8048

1. The 8048 family of processors has 3 types of memory: program memory (P), internal data memory (I) and external data memory (X). The difference between members of this family is the amount of internal data memory and internal program memory available as indicated in the charts below. Although the amount of internal program memory differs between family members, overall addressable memory is the same, except for the 8035/8039/8040 which have no internal program memory.

   All commands that address memory (i.e. Download and A/M/T/U/Z) default to program memory. To indicate a specific memory type, input P, I or X after the command keyword (e.g. >MX 12).

   The following charts indicate addressable memory and I/O for Static Port Mode and Non-Static Port Mode (section 2.7.1). Remember that the 8035, 8039 and 8040 cannot be emulated in Static Port Mode because they have no internal program memory.

Non-Static Port Mode				
CPU	Addressable Memory			Addressable I/O
	Program	Internal Data	External Data	
8035	000H-FFFH	00H-3FH	00H-FFH	00H-02H,04H-07H
8039	000H-FFFH	00H-7FH	00H-FFH	00H-02H,04H-07H
8040	000H-FFFH	00H-FFH	00H-FFH	00H-02H,04H-07H
8048	000H-FFFH	00H-3FH	00H-FFH	00H-02H,04H-07H
8049	000H-FFFH	00H-7FH	00H-FFH	00H-02H,04H-07H
8050	000H-FFFH	00H-FFH	00H-FFH	00H-02H,04H-07H
8748	000H-FFFH	00H-3FH	00H-FFH	00H-02H,04H-07H
8749	000H-FFFH	00H-7FH	00H-FFH	00H-02H,04H-07H

Static Port Mode				
CPU	Addressable Memory			Addressable I/O
	Program	Internal Data	External Data	
8048	000H-3FFH	00H-3FH	00H-FFH	00H-02H,04H-07H
8049	000H-7FFH	00H-7FH	00H-FFH	00H-02H,04H-07H
8050	000H-FFFH	00H-FFH	00H-FFH	00H-02H,04H-07H
8748	000H-3FFH	00H-3FH	00H-FFH	00H-02H,04H-07H
8749	000H-7FFH	00H-7FH	00H-FFH	00H-02H,04H-07H

2. When the MICE module is powered-up or reset, the external program memory access control signal is disabled (DA). If the control signal is enabled, emulation memory only emulates internal program memory.

3. MICE uses the internal 4K byte emulation memory. Emulation memory 0H-7FFH is located at position U55; and 800H-FFFH is at position U56. Internal memory selection is configured by the EA control, target EA signal and CPU type (see Target Select command, section 2.7.2) as follows:

Command	H/W EA Signal	8048	8049	8050
DA	don't care	0-4K	0-4K	0-4K
EA	logic "1"	none	none	none
EA	logic "0"	0-1K	0-2K	0-4K

4. The 8048 also provides a memory transfer command (TM) to move the Target Program to MICE Program Memory at the same location (section 5.2).

2.6.2 Memory Selection for NEW MICE-II 8031/8344

Syntax for the memory access commands is indicated below:

```
Program & Internal/External Data Memory - M[P|X|I][ a1[ a2[ d1[..d8][S]]]]
Program & Internal/External Data Memory - T[P|X|I] a1 a2 {S|M|a3[V]}
On-Chip Internal Program Memory         - MV[ a1[ a2[ d1[..d8] S]]]
On-Chip Internal Program Memory         - TV a1 a2 {S|a3[V]}
```

1. The 8051 and 8344 processor families have four types of memory as listed below:

CPU	Addressable Memory				Addressable I/O
	Internal Program	Internal Data	External Program	External Data	
8031	none	0H-7FH	0000H-FFFFH	0000H-FFFFH	Ports P1&P3
8032	none	0H-FFH	0000H-FFFFH	0000H-FFFFH	Ports P1&P3
8344	none	0H-BFH	0000H-FFFFH	0000H-FFFFH	Ports P1&P3

a) "On-chip" internal program memory (V) which is located in the verification CPU at U69. Note that on-chip internal program memory can only be read, **it cannot be executed.**

b) Program memory (P) which consists of external emulation program memory located on HUEM/SUEM or on the target.

c) "On-chip" internal data memory (I) which is located in the emulation CPU at U51.

d) External data memory (X) which is located on HUEM/SUEM or on the target.

V, P and X serve as optional qualifiers for M/T commands, with program memory (P) as the default value. Note that if qualifiers V or I are used to address memory outside the range of 0~1FFFH or 0~FFH respectively, an error message will be displayed:

"LOCATION ERROR!".

2. If external program memory or external data memory is used, then an HUEM or SUEM board must be used for emulation memory.

3. The emulation memory range required for external program memory and external data memory can be completely covered with HUEM and SUEM (max 128K bytes).

Emulation Memory Selection

Target CPU	Emulation CPU	Emulation Memory	Memory Structure			
			Segregated*		Combined	
			SUEM**	HUEM**	SUEM**	HUEM**
8031	8031/8051/8751 or 80C31F	SUEM or HUEM	128KB	128KB	64KB	64KB
8032	8032/8052/8752	SUEM or HUEM	128KB	128KB	64KB	64KB
8344	8344/8044/8744	SUEM or HUEM	128KB	128KB	64KB	64KB

* Use for external program memory, data memory, or both.

** Maximum values are indicated.

4. However, if HUEM or SUEM is selected and memory space is segregated, input the EX command (see Chapter 6). Then locate program memory in one memory bank (0000H-FFFFH) and data memory in the other bank (10000H-1FFFFH). (HUEM has two separate 64K memory banks.) In this configuration, only the MOVX instruction can be used to execute external data memory.

   Example: If the program memory range is 0_3FFFH and data memory range is 1000H_1FFFH, then set emulation memory as indicated below for HUEM or SUEM.

HUEM Enable all emulation memory by setting -
* U14 and U36 S1-S9 ON and S10 OFF
Emulate program memory in bank 1 by setting -
* memory segment to 0H (U28 S1-8 all ON)
Emulate data memory in bank 2 by setting -
* memory segment to 10000H (U43 S1 OFF, all others ON)

SUEM Mapping the SUEM memory as follows:

* RANGE 0
* MAP 0 3FFF I
* MAP 4000 10FFF E
* MAP 11000 11FFF I
* MAP 12000 1FFFF E

5. If HUEM/SUEM is selected and memory is combined, input the DX command. Then locate program and data memory in the same memory bank with the memory segment set to 0H (HUEM - U28/U43 S1-8 all ON or SUEM- the memory ranged from 0 to 0FFFFH should be mapped to "I"), and **do not** use the MX or TX commands.

6. If an 8-bit address is being used (MOVX @Ri), the contents of the Port2 SFR remain at Port2 pins throughout target memory; but throughout HUEM/SUEM A8~A15 may be set to reflect the contents of the Port2 SFR or may be set to all 0 by placing jumpers on the CEP board as indicated below.

A8~A15	X15	X16
P2	O	C
O	C	O

C: Close
O: Open

Status of HUEM/SUEM A8~A15 during MOVX @Ri

Note: A8~A15 all equal 0 for CEP-8031, Revision B, and earlier.

2.6.3. Memory Selection for NEW MICE-II 8052/80C152

Memory access commands:

```
Program & Internal/External Data Memory - M [P|X|I][ a1[ a2[ d1[..d8][S]]]]
Program & Internal/External Data Memory - T [P|X|I] a1 a2 {S|M|a3[ V]}
On-Chip Internal Program Memory         - M V[ a1[ a2[ d1[..d8] S]]]
On-Chip Internal Program Memory         - T V a1 a2 {S|a3[V]}
```

1. The 8051 and 8044 processor families have four types of memory:

   a) "On-chip" internal program memory (V) which is located in the Programmer and Verification Box. Note that on-chip internal program memory can only be read, unable to execute.

b) Program memory (P) which consists of external emulation program memory located on HUEM/SUEM or on the target, where the "on-board" internal emulation program memory is located in RAM at U73 on CEP board. Note that the RAM at U73 emulates internal CPU program memory.
c) "On-chip" internal data memory (I) is located in the emulation CPU at U14 of EPOD-8052 (or EPOD-80C152).
d) External data memory (X) is located on HUEM/SUEM or on the target.
e) If EPOD-8052 is malfunction after power-on, MICE will display an error message instead of sign-on message:

   "EPOD-8052 FAILURE!"

f) To select V type memory, PVPOD-8052 is required. Without connecting to PVPOD-8052, MICE will display: "NO PVPOD-8052!"

V, P and X serve as optional qualifiers for M/T commands, program memory (P) is the default. Note that if qualifier V or I is used to address memory outside the range of 0~1FFFH or 0~FFH respectively, an error message will display:

"LOCATION ERROR!".

CPU	Addressable Memory				Addressable I/O
	Internal Program	Internal Data	External Program	External Data	
8051/8751	0H-FFFH	0H-7FH	1000H-FFFFH	0000H-FFFFH	Ports P0-P3
8031	none	0H-7FH	0000H-FFFFH	0000H-FFFFH	Ports P1&P3
8052/8752	0H-1FFFH	0H-FFH	2000H-FFFFH	0000H-FFFFH	Ports P0-P3
8032	none	0H-FFH	0000H-FFFFH	0000H-FFFFH	Ports P1&P3
83C154	0H-3FFFH	0H-FFH	4000H-FFFFH	0000H-FFFFH	Ports P0-P3
80C154	none	0H-FFH	0000H-FFFFH	0000H-FFFFH	Ports P1&P3
8344	none	0H-BFH	0000H-FFFFH	0000H-FFFFH	Ports P1&P3
8044/8744	0H-FFFH	0H-BFH	1000H-FFFFH	0000H-FFFFH	Ports P0-P3
83C152JA	0H-1FFFH	0H-FFH	2000H-FFFFH	0000H-FFFFH	Ports P0-P4
80C152JA	none	0H-FFH	0000H-FFFFH	0000H-FFFFH	Ports P1&P3, P4
80C152JB	none	0H-FFH	0000H-FFFFH	0000H-FFFFH	Ports P0-P6

2. The 8051/8751, 8052/8752 or 83C154 internal "emulation" program memory (memory type P) must be located in emulation memory on the CEP board (62256 RAM at U73) to permit execution.

   a) If internal program memory is located in (EP)ROM for an 80C51VS/85C154VS target system, either read this data to the host computer and download it to MICE emulation memory, or place the target (EP)ROM (if a 2732/2764/27128) on the CEP board at U73. (2732 pin-1 must be mated with socket pin-3.)*

3. If external program memory or external data memory is used, an HUEM or SUEM board must be used for emulation memory.

4. The emulation memory range required for external program memory and external data memory can be completely covered with HUEM/SUEM (128K bytes maximum).

Target CPU	Emulation CPU	Emulation Memory	Memory Structure			
			Segregated*		Combined	
8051/8751	85C154VS or 80C51VS	<4K bytes (CEP U73)	SUEM**	HUEM**	SUEM**	HUEM**
		>4k bytes (HUEM/SUEM)	128KB	128KB	64KB	64KB
8031	8031/8051/8751/ 8032/8052/8752 80C51VS/85C154VS/ 83C154/80C154	HUEM or SUEM	128KB	128KB	64KB	64KB
8052/8752	85C154VS	<8K bytes (CEP U73)	SUEM**	HUEM**	SUEM**	HUEM**
		>8K bytes (HUEM/SUEM)	128KB	128KB	64KB	64KB
8032	8032/8052/8752 85C154VS/80C154/ 83C154	HUEM or SUEM	128KB	128KB	64KB	64KB
83C154	85C154VS	<16K bytes (CEP U73)	SUEM**	HUEM**	SUEM**	HUEM**
		>16K bytes(HUEM/SUEM)	128KB	128KB	64KB	64KB
80C154	85C154VS/83C154 or 80C154	HUEM or SUEM	128KB	128KB	64KB	64KB
8344	8344/8044/8744	HUEM or SUEM	128KB	128KB	64KB	64KB
80C51FA	83C51FA/80C51FA	HUEM or SUEM	128KB	128KB	64KB	64KB
83C152JA	80C152JB	<8K (CEP U73)	128KB	128KB	64KB	64KB
		>8K bytes (HUEM/SUEM)	128KB	128KB	64KB	64KB
80C152JA	83C152JA/80C152JA/ 80C152JB	HUEM or SUEM	128KB	128KB	64KB	64KB
80C152JB	80C152JB	HUEM or SUEM	128KB	128KB	64KB	64KB

Emulation Memory Selection

5. Since HUEM has two separate 64K memory banks, if HUEM is selected and memory space is segregated, input the EX command (see Section M.3). Then locate program memory in one memory bank (0000H-FFFFH) and data memory in the other bank (10000H-1FFFFH). In this configuration, only the MOVX instruction can be used to access external data memory.

---

* Used for external program memory, data memory, or both.

** Maximum values indicated.

6. If HUEM is selected and memory is combined, input the DX command. Then locate program and data memory in the same memory bank with the memory segment set to OH (HUEM - U28/U43 S1-8 all ON), and do not use the MX or TX commands.

7. If the CEP emulation RAM at U73 is being executed ($\overline{EA}$=1 and the PC<4KB for 8051, PC<8KB for 8052, PC<16K for 83C154, PC<8K for 83C152JA, PC<8K for 80C152JB*), at the same range program memory located in HUEM/SUEM or on the target will be ignored (except when binary download or high speed download incorporated with SUEM is used).

8. If the on-board CEP internal emulation program memory (CEP-8052 U73) is enabled and the binary download** or high speed download*** incorporated with SUEM is used and the SUEM's memory in the range of internal program memory is mapped to be enabled, the program in the range of internal program memory will not be downloaded to the on-board CEP internal emulation program memory. They will be downloaded to the SUEM instead.

---

* For 80C152JB, the input status are $\overline{EA}$ = 1, EBEN = 1.

** Refer to Section 5.1, SUEM SECTION of this manual.

*** Refer to Section 4.3, SUEM SECTION of this manual.

2.6.4 Memory Selection for NEW MICE-II 80515/80535

Syntax for the memory access commands is indicated below:

```
Program & Internal/External Data Memory - M[P|X|I][ a1[ a2[ d1[..d8][S]]]]
Program & Internal/External Data Memory - T[P|X|I] a1 a2 {S|M|a3[V]}
On-Chip Internal Program Memory         - MV[ a1[ a2[ d1[..d8] S]]]
On-Chip Internal Program Memory         - TV a1 a2 {S|a3[V]}
```

1. The 80515/80535 have the following memory types:

   a) The 80515 has "on-chip" internal program ROM (V) which is located in the verification CPU at U62. Remember that this memory type can only be displayed, i.e. it cannot be modified or executed. (Note that the 80535 does not provide this memory type.)

   b) Program memory (P) which consists of external program memory located on HUEM/SUEM or on the target, and CEP "on-board" internal emulation program memory located in RAM at U58/59. Note that the RAM at U58/59 emulates internal CPU program memory.

   c) "On-chip" internal data memory (I) which is located in the emulation CPU at U45.

   d) External data memory (X) which is located on HUEM/SUEM or on the target.

V, P, I and X serve as optional qualifiers for M/T commands as indicated above, with program memory (P) as the default selection. Note that if qualifiers V or I are used to address memory outside the range of 0~1FFFH or 0~FFH respectively, an error message will be displayed:

"LOCATION ERROR!".

CPU	Addressable Memory				Addressable I/O
	Internal Program	Internal Data	External Program	External Data	
80515	0H-XH*	0H-FFH	0000H-FFFFH	0000H-FFFFH	Ports P0-P5
80535	none	0H-FFH	0000H-FFFFH	0000H-FFFFH	Ports P1,P3-P5

*The standard size of internal program memory for the 80515 is 8K bytes, but the 80515-E-A emulation processor provides an optional 2, 4, 8 or 16K bytes which may be specified using the "EM #" command (section 2.6).

2. 80515 on-chip internal program memory (0-8K bytes, memory type V) must be transferred (using the TV command) to the CEP on-board internal emulation program memory (memory type P) to permit modification or execution.

3. HUEM or SUEM may be used to emulate external memory if it is not available in the target system.

4. HUEM or SUEM can support the entire emulation memory range required for external memory (max 128K bytes).

Emulation Memory Selection

Target CPU	Emulation Memory	Memory Structure			
		Segregated*		Combined	
80515	Internal Program Memory (CEP U58/59)				
	External Memory	HUEM**	SUEM**	HUEM**	SUEM**
	(HUEM/SUEM)	128KB	128KB	64KB	64KB
80535	HUEM or SUEM	128KB	128KB	64KB	64KB

5. When using HUEM, set up emulation memory as follows:

   a) If memory space is segregated, input the EX command. Then locate external program memory in memory bank 1 (0H-FFFFH) and external data memory in bank 2 (10000H-1FFFFH). (HUEM has two separate 64K memory banks.) In this configuration, only the MOVX instruction can be used to access external data memory.

      Example: If the external program memory range is 0~FFFFH and external data memory range is also 0~FFFFH, then set emulation memory as indicated below for HUEM.

      Enable all emulation memory by setting -
      * U14 and U36 S1-S9 ON and S10 OFF
      Emulate external program memory in bank 1 by setting -
      * memory segment to 0H (U28S1-8 all ON)
      Emulate external data memory in bank 2 by setting -
      * memory segment to 10000H (U43 S1 OFF, all others ON)

b) If memory space is combined, input the DX command. Then locate external program/data memory in the same memory bank (0-FFFFH) with the memory segment set to 0H (U28/U43 S1-8 all ON), and **do not** use the MX or TX commands.

6. When using SUEM, set up emulation memory as follows:

   a) If the EX command is input to specify segregated memory space, then mapping the SUEM memory as follows:

      * RANGE 0
      * MAP 0 1FFFF I

      Note: If the CEP "on-board" internal emulation program memory (CEP-80515 U58/59) is to be accessed, and SUEM is used, then do not use binary download or high speed download.

   b) If the DX command is used to specify combined memory space, then mapping the SUEM memory ranged from 0 to 0FFFFH should be mapped to "I".

7. If CEP on-board internal emulation program memory is enabled ($\overline{EA}$=1 and the PC is within the internal program memory range) then external program memory of the same address range in HUEM/SUEM or target memory will be ignored (except when binary download or high speed download incorporated with SUEM is used).

2.7 RS-232C Cable Connection

After selecting the proper data rate and transmission characteristics, next determine whether the controlling device has a data terminal equipment (DTE) interface or a data communication equipment (DCE) interface, with or without handshaking. Display terminals are usually equipped with DTE interface; computer systems usually have both.

There are several methods for determining the type of interface on the controlling device. The first method is by simple trial and error. If this fails, procedures two and three listed below can be used to determine the interface type. (It must be known beforehand whether or not handshaking is selected as a software option for the controlling device, as it cannot be easily detected by examining hardware signals.) Subsequent instructions detail how to accomplish interface header rewiring if required.

1. Try connecting the two devices together. If the response is correct, the interfaces match and no further adjustment is required. If the interconnect does not work, a mismatch exists and indicates that a connection change is necessary.

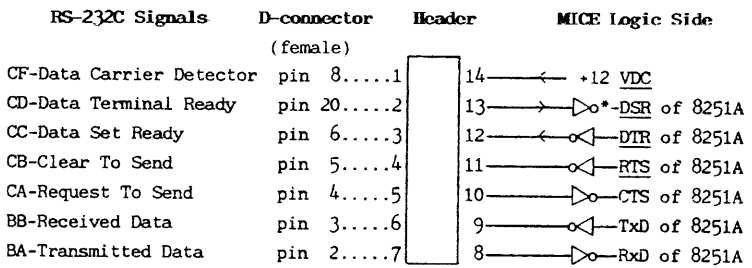
2. Data are transmitted on pin 2 and received on pin 3 of the D-connector for DTE devices; the reverse is true for DCE devices.

3. When not transmitting, the voltage (with respect to pin 7 [ground] of the D-connector) is -12 VDC on pin 2 for DTE devices and -12 VDC on pin 3 for DCE devices.

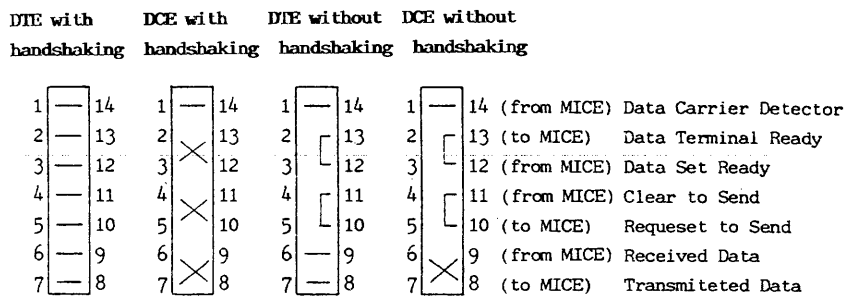
The header designated as U13 on the CEP board is used to configure the interface between NEW MICE-II and the controlling device. NEW MICE-II are shipped from the factory with a straight-wired header for DTE with handshaking.

Pins 1-7 on the interface header are connected to the female D-connector (J2) while pins 8-14 are connected to the NEW MICE-II, RS-232C interface logic.

Pin definitions on the header are shown below:



* RS-232C interface gate for 1488 or 1489 depending on signal Input or Output. The wiring configuration for all three header types is shown below. Note that if the controlling device has a DCE interface, incoming signals are now outgoing, and vice versa, on the same pins. By removing the cover to the header, the interface type used by MICE can be determined according the following wiring configuration.



When wired for DTE, connect NEW MICE-II using a straight-wire cable with male D-connectors; for other header types, the cable must be rewired accordingly. If a display terminal is connected after making the proper connections, NEW MICE-II should respond when power is applied. (A display terminal should always be used to check new NEW MICE-II units to ensure that they are properly functioning.)

Finally, NEW MICE-II requires that both Request to Send and Data Terminal Ready inputs, pins 10 and 13 respectively on the logic side of the header, be at +12 VDC before it transmits any data. If the controlling device does not supply the necessary voltage, it can be obtained by reconnecting pins 10 and 13 directly to pin 14 (Received Line Signal Detector) of header U13 which is always at +12 VDC.

To accommodate computer systems which are sending commands and data too fast for NEW MICE-II, the Data Set Ready output, normally at +12 VDC, is pulled low by NEW MICE-II to -12 VDC. The signal is restored to +12 VDC when NEW MICE-II is again ready. Also, to accommodate display terminals with slow carriage returns or linefeeds, six null characters are always transmitted after a carriage-return is issued.

2.8 Applying Power to NEW MICE-II

Connect the RS-232C cable from the controlling device to the female D-connector at the rear of NEW MICE-II. (The recommended power-on sequence is to first apply power to the target and then to NEW MICE-II; and to use the opposite sequence when powering off.) With the controlling device ready, connect the power cable with the locking tab pointing up, and apply power. Within a few seconds, the following start-up message should display. (Also refer to section 2.8.1 for a detailed description of power-on messages for the 8048.)

```
***MICE-II-type V#.#***
>
```

---

type identifies target processor being emulated by the personality card.

#.# indicates version number of controlling program on personality card.

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

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 NEW MICE-II command summary is listed; otherwise, the error message **"WHAT?"** is printed. Reset the switch section where necessary and remember to wait a few seconds before power is turned on again.

Notes 1. If target VCC is not provided when NEW MICE-II is powered up, the following message will display:

```
8085          - NO TARGET VCC; TRAP, INTR, HOLD DISABLED!
8048          - NO TARGET VCC; INT, T1 DISABLED!
8031/8344     - NO TARGET VCC; EA ENABLED,RESET DISABLED,
                MEMORY IS SEGREGATED!
8052/80C152   - NO TARGET VCC; EA ENABLED,RESET DISABLED,
                MEMORY IS SEGREGATED!
80515/80535   - NO TARGET VCC; EA ENABLED,RESET DISABLED,
                MEMORY IS SEGREGATED!
                Internal Program Memory = 8K Bytes!
```

2. The CPU is automatically reset after power-up, the software reset command "r" (section 2.9) or MICE Reset command X (section 5.7). If the target system has any peripheral devices or slave processors which must be synchronized with a CPU reset, then the reset must be performed on the target side. A target reset signal will also reset the emulation CPU. However, the reset control signal for MICE must be enabled (section 6.2) for the 8031/8344 and 80515/80535 otherwise the target reset signal cannot reset the emulation CPU. Note that a synchronized reset may be performed by either MICE or the target system for the 8085, since this processors support a RESET output signal when the reset line for peripheral devices or slave processors on the target side is connected to reset pin-3.

2.8.1 Applying Power for NEW MICE-II 8048

1. After power-on, NEW MICE-II 8048 displays the following start-up message:

```
***MICE-II 8048 V#.#***
IS DATA BUS USED IN STATIC PORT MODE ? (Y/N)_
```

   Key in"Y" if data bus isusedas Input/Output only, or "N" if data bus is also used for External Addresses.

2. Then another message is displayed:

```
IS PORT 2 USED IN STATIC PORT MODE ? (Y/N)_
```

   Key in "Y" if PORT 2 is used as Input/Output only, or "N" if PORT 2 is used as 8243 External port or External address bus.

3. To terminate parameter display and select the default setting to use data bus and port 2 as Input/Output only, key in <ESC>, BACKSPACE or RUBOUT.

4. If anything other than Y, N, <ESC>, BACKSPACE or RUBOUT is keyed in, the query will display again.

5. When MICE is powered-up or reset, the default processor type emulated is 8048. To emulate a different family member, use the Target Select command (section 2.8.2).

Example: Display and change the processor type currently being emulated.

```
>!
CEP48
>8049
>!
CEP49
>
```

6. To check which type DATA BUS and PORT 2 are used, enter the !M command.

Example: Display the type used when I/O only is selected.

```
>!M
DATA BUS IS USED AS STATIC PORT MODE
PORT 2 IS USED AS STATIC PORT MODE
>
```

2.8.2 Microprocessor Selection for NEW MICE-II 8048

The CEP-8048 can emulate the following microcontrollers:

8035	8048	8748
8039	8049	8749
8040	8050	

NEW MICE-II uses an 8040 as the emulation processor. (Note that the 8040 has no internal program memory.) The target processor currently under emulation is selected by specifying the appropriate Target Select command as indicated in the following chart.

Target Select Command	Processor Emulated							
	8035	8039	8040	8048	8049	8050	8748	8749
8048	YES	--	--	YES	--	--	YES	--
8049	YES	YES	--	YES	YES	--	YES	YES
8050	YES	YES	YES	YES	YES	YES	YES	YES

2.8.3 No Response

If there is no response from NEW MICE-II, check the following items:

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 on 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 NEW MICE-II are the same, resetting if necessary. If a message does appear but is garbled, any combination of the 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 program is running and the RS-232C cable is connected to the correct port and that the port is working.

Note: When changing the communication configuration switch, note that it is only read during a power-up or reset. After turning the power off, wait a few seconds before the power is turned on again to allow the capacitors to fully discharge.

If the problem still cannot be found, contact your local Microtek representative for further assistance.

2.8.4 Failure Device

During the delay prior to the start-up message the RAMs and EPROMs on the CEP board are tested. If any component failures exist, they are listed as detected in the following format: "U## - FAILURE", where ## is the component number on the personality board.

2.9 Control Processor Software Reset Command

The character "r" is the command to reset NEW MICE-II. Remember that except for this command all alphabetic characters must be entered in upper-case; no other lower-case characters are recognized.

CHAPTER 3

NEW MICE-II COMMAND LANGUAGE

All NEW MICE-II products have a common set of commands identified by a single or double character regardless of the processor being emulated. No additional time or effort is required to learn a new command language when the target processor changes.

These commands are described in the following chapters along with a variety of available options, though not all options are applicable for the different types of processors. For the specific processor being emulated, consult the Help "?" command. The commands described in the following chapters are grouped as follows:

NEW MICE-II Utility Commands
- ? Help Command
- ! Attention Command

Memory, Port and Register Commands
- M Memory Display/Examine/Modify/Fill/Search Command
- T Memory Checksum/Test/Transfer/Compare Command
- A Line Assembly Command
- Z Disassembly Command
- I Port Input Command
- O Port Output Command
- R Register Display/Modify Command
- J Jump/Branch Command
- X Reset/Initialization Command

Control Signal Commands

D Disable/Display Control Signal Command
E Enable/Display Control Signal Command

Emulation and Trace Control Commands

G Go/Execution Command
H Halt/Breakpoint Set Command
F Forward Trace Command
B Backward Trace Command
L List Trace Buffer Command

Stepped Emulation Commands

C Single Cycle, Step Command
S Instruction Step Command

Utility Commands Involving a System

: Download Command (Intel Format)
/ Download Command (Tektronix Format)
U Upload Command

3.1 Command Syntax

NEW MICE-II 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 and must be terminated by a carriage-return <CR>. The general syntax of NEW MICE-II commands is:

command [parameters] <CR>

where: **command** is the command representation.

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

Where a space is shown in the syntax, either a space or comma can be used. A <CR> must be used to terminate a command input line. In most cases line feed <LF> or <CR> have the same effect, except where otherwise noted. Note that brackets [ ] and braces { } are used for describing command syntax only, and are not used in command input.

3.2 Notations and Conventions

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

1. An upper-case entry must be input.

2. A lower-case entry in the description of a command is the class-name for a parameter. A particular value for this class must be entered. A class-name never appears in an actual operable command. For example, the lower-case entry - "**start-address**" means that NEW MICE-II will only accept a hexadecimal value as an address in the target processor's memory space.

3. A required entry is shown without any enclosures; whereas an optional entry is denoted by enclosing it in brackets. For example, in the command description - "**G[address]**", the command "G" is required, and the brackets around the entry "address" means that it 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 are first entered. For example, in the command description - "**I port[count[time]]**", the command "I" is required; and the brackets mean that the selection of "count" and "time" is optional in the command. However, a "count" must first be entered if a "time" value is to be specified.

4. Where an entry must be selected from a choice of two or more, the choices for the required entry are enclosed in braces and separated by vertical bars. For example - "**{S|M|a3[V]}**" indicates that either "S", "M" or "a3" must be entered.

5. Where a choice exists for an optional entry, the choices are enclosed in brackets and separated by vertical bars. For example, "**[I|H|T]**" indicates that either "I", "H" or "T" may be entered.

6. "Addx" is a hex address with a wildcard byte pair of "XX" or "XXXX", or a wildcard nibble of "X" [only for the third digit (e.g. X23) for 8048]; where X indicates that the digits are "don't care". For example - "**12XX**" means all addresses between 1200H and 12FFH.

7. Commands A/C/S/I/R all use carriage-return <CR> or line-feed <LF> to display the next line.

8. In the Memory Modify (M) command, use a <CR> to advance to the next location and a <LF> to return to the previous location.

9. Entries underlined in command examples indicate user input.

3.3 Editing Characters

Each character entered on the keyboard is stored in a line editing buffer until <CR> is entered. If more than 80 characters are entered without inputting a <CR>, 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 <CTRL> while the character is typed. Control character input is expressed with the control command enclosed in < > brackets.

BACKSPACE deletes the preceding character from the line buffer and from the display. Repeated usage is allowed. <CTRL-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 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. ESCAPE is also expressed as <ESC>. On some terminals, this key may be labeled ESC. <CTRL-Y> performs the same function.

<CTRL-R> causes a <CR> or <LF>, followed by a redisplaying of the current undeleted contents in the line buffer. This is useful to see a clean copy of the command line after RUBOUT has been used.

<CTRL-X> ignores the current contents of the line buffer and shifts the cursor to the first position of the next line, awaiting input of new data.

3.4 Control Characters and Delimiters

The following control characters have special meaning for all NEW MICE-II firmware:

<CTRL-J>: is the same as <LF>.
<CTRL-M>: is the same as <CR>.
<CTRL-S>: stops data transmission from NEW MICE-II.
<CTRL-Q>: continues data transmission from NEW MICE-II.
<CTRL-Y>: is the same as <ESC>.

3.5 Reference Program for Examples

The following sample program for an 8085 target is used as the basis for most of the examples listed in this user's guide. It includes commonly used instructions for performing various bus cycles.

```
LOC   OBJ      LINE   LABEL    SOURCE CODE
0000  31F07F   0001            LXI SP,7FF0
0003  213412   0002   B0003    LXI H,1234
0006  E5       0003            PUSH H
0007  212143   0004            LXI H,4321
000A  0E06     0005            MVI C,06
000C  71       0006   B000C    MOV M,C
000D  23       0007            INX H
000E  E3       0008            XTHL
000F  0D       0009            DCR C
0010  C20C00   0010            JNZ 000C
0013  E1       0011            POP H
0014  C30300   0012            JMP 0003
```

CHAPTER 4

NEW MICE-II UTILITY COMMANDS

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

Notes 1. If any code other than FFH*1 is placed in firmware at the location indicated in table 4-1, it will cause the handshaking code at this location to be sent to the host computer (or terminal) when MICE is waiting for input. This extra code can be used to improve interface efficiency with the host.

2. The handshaking code 03H*2 is sent to the host computer when MICE is waiting for input. Any code other than 03H may be placed in firmware (table 4-1) to suit the user's specific requirements.

NEW MICE-II	CEP Location	Address	Handshaking Code	
			FFH*1	03H*2 (ETX)
8085	U4	1FFDH	V1.0-V3.0	V3.1 & later
8048	U1	3FFDH	V1.0-V3.1	V3.2 & later
8031/8344	U1	7FFDH	V3.0	V3.1 & later
80515/80535	U1	7FFDH	---	V3.0 & later
8052/80C152	U1	AFFDH	---	V3.0 & later

Table 4-1 Handshaking Codes

4.1 Help Command - ?

?

? is the command for Help.

The command summary or the target processor currently being emulated is displayed on the terminal. All commands are common regardless of target processor type, but command parameters may differ for the different processors under emulation.

Example: Display the command summary for NEW MICE-II 8085.

```
>?
ASSEMBLY            A [loc]
BACKWARD TRACE      B [R]addx[ c[ q]]
CYCLE STEP          C [c]
DISABLE             D [I|H|T]
ENABLE              E [I|H|T]
FORWARD TRACE       F [R]addx[ c[ q]]
EXECUTION           G [address]
BREAKPOINT          H [0[1|2]|1 [addx[ c[ q]]]|2 [addr]]
INPUT               I port[ c[ time]]
JUMP                J address
LIST TRACE          L [step[ a1[ a2[ q..]]]|S[step]|Z[step]|N]
MEMORY              M [a1[ a2[ d1[ d2[ ..d8]][ S]]]]
OUTPUT              O port d1[ d2[ ..d8]]
REGISTER            R [A|B|C|D|E|H|L|M|P|S]
INSTRUCTION STEP    S [S|R][c]|[Z]
TRANSFER/TEST       T a1 a2 {S|M|a3[ V]}
UPLOAD              U a1 a2 [T|I]
RESET               X
DISASSEMBLY         Z [a1 [a2]]
DOWNLOAD            : (INT), / (TEK)
HELP                ? [B]
ATTENTION           !
>
```

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

NEW MICE-II 8085 is used as the basis for all examples in this manual. Command syntax varies for different NEW MICE-II emulators; for the proper syntax consult the Help (?) command.

4.2 Attention Command - !

!

! is the command for Attention.

The target processor type currently being emulated by the personality card in NEW MICE-II is displayed on the terminal. Six characters are used for identification.

Example: Display the processor type currently being emulated (8085).

```
>!
EP8085
>
```

The messages for 8-bit series processor types are:

NEW MICE-II		TARGET PROCESSOR TYPE
8085	(Intel Corporation)	- 8085
8048	(Intel Corporation)	- 8048/8049/8050 *
8031/8344	(Intel Corporation)	- 8031/8032/8344
8052/80C152	(Intel Corporation)	- 8031/8051/8032/ 8052/80154/83154/ 8344/80C51FA/ 80C152JA/83C152JA/ 80C152JB
80515/80535	(Siemens Corporation)	- 80515/80535

* Refer to section 2.8.2.

CHAPTER 5

MEMORY, PORT AND REGISTER COMMANDS

These commands give access to the contents or current values stored in designated memory locations, I/O ports and registers. The memory type, maximum amount of addressable memory, and total amount of addressable I/O are shown below. Note that targets with only one memory type have no distinction between program and data memory. For microcontroller versions with multiple memory types, refer to section 2.6.

CPU	Addressable Memory	Addressable I/O
8085	0000H-FFFFH	00H-FFH

NEW MICE-II always checks that memory type and memory/port addresses are valid before an operation is performed. References to an invalid address result in an error message (ERROR!) being printed and the command ended. NEW MICE-II will verify memory write operations. However, it will not verify memory read or I/O operations. If there is no memory when executing a memory read operation, then data read is random. And if there is nothing connected to the port when executing an I/O operation, then data written is lost and data read is random.

Notes 1. When performing READ/WRITE functions, if external circuitry does not respond with the following signal, "TARGET IS NOT READY!!" will appear on the console and the contents of the registers will be retained.

8085 8048	READY $\overline{SS}$

However, if the target does not respond when performing READ/WRITE functions for NEW MICE-II 8031/8344, 8052/80C152 or 80515/535, incorrect data will be displayed.

2. A blank space (ASCII 20H) along with a handshaking code (ASCII 03H) is sent at the end of each displayed line immediately before the cursor for the Assembly, Memory Modify and Register Modify commands. This code serves as an end of data message to the host computer; and is provided to help in implementing symbolic debuggers. (Refer to page 4-1.)

Memory Type Specification for NEW MICE-II 8031/8344

1. The following commands can be used to verify the on-chip internal program memory at U69 (for 8051/8751/8052/8752/8044/8744). (Refer to section 2.6 for a detailed description of memory types.)

   a) Use the following command to display or search on-chip program memory.

      >MV[a1[a2[d1[...d8] S]]]

   b) Use the following command option for on-chip program memory to perform checksum (S), transfer to program emulation memory (a3) at HUEM/SUEM, or block compare (V).

      TV a1 a2 {S|a3[V]}

c) Use the following command to disassemble on-chip memory.

ZV[a1[a2]]

Notes 1) a1 and a2 must be within 0~1FFFH, otherwise MICE will display an error message: "**LOCATION ERROR!**".

2) When performing MV/TV/ZV commands, **do not** insert or remove any device from the target verification CPU socket at U69.

3) a3 indicates address space in emulation program memory.

2. All memory manipulation commands are supported for on-chip data memory by using the M and T commands with the "I" memory qualifier (i.e. MI/TI).

3. The values of a1-a3 must be within 0~0FFH when accessing internal data memory (i.e. MI/TI), otherwise MICE will display "**LOCATION ERROR!**".

   The values of a1-a3 must be within 0~FFFFH when accessing external program memory or "on-board CEP internal emulation program memory located in RAM at U58 (i.e. M[P]/T[P]), or external data memory (i.e. MX/TX), otherwise MICE will display "**ERROR!**".

4. Do not use MX/TX commands when HUEM/SUEM is selected and emulation memory is combined.

Memory Type Specification for NEW MICE-II 8052/80C152

Refer to NEW MICE-II 8052 Appendix (For PVPOD 8052).

Memory Type Specification for NEW MICE-II 80515/80535

1. The following commands can be used to verify the 80515 on-chip internal program ROM in the verification CPU at U62.

   a) Use the following command to display or search 80515 on-chip internal program ROM.

      >MV[a1[a2[d1[..d8] S]]]

   b) Use the following command for 80515 on-chip internal program ROM to perform checksum (S), transfer to emulation program memory (a3) at the U58/59 RAM or HUEM/SUEM, or block compare (V).

      >TV a1 a2 {S|a3[V]}

   c) Use the following command to disassemble the 80515 on-chip internal program ROM.

      >ZV[a1[a2]]

   Notes: 1) a1 and a2 must be within 0~1FFFH, otherwise MICE will display an error message: "LOCATION ERROR!".
   2) When performing MV/TV/ZV commands, do not insert or remove any device at the verification CPU socket (U62).
   3) a3 indicates address space in emulation program memory.

2. All memory manipulation commands are supported for on-chip internal data memory by using the M/T commands with memory qualifier "I" (i.e. MI/TI).

3. The values of a1-a3 must be within 0~0FFH when accessing internal data memory (i.e. MI/TI), otherwise MICE will display "**LOCATION ERROR!**".

4. Do not use MX/TX commands when external program memory and external data memory are combined.

Referring Special Function Registers for NEW MICE-II 8031/8344

1. There are two different ways to refer special function registers in A/Z commands:

   a) Referring the register by mnemonic*.
   b) Referring the register by address*.

LOC	OBJ	LINE	LABEL	SOURCE CODE
0000	85E0D0	0001		MOV PSW,ACC
0002	85E0D0	0002		MOV D0,ACC

   * Refer to Memory Organization, section 6.1 for 8031 family or figure 18-6 for 8344 family, Intel Microcontroller Handbook, 1984.

2. There are four different ways to refer bit addresses for special function registers in the Assembly command.

   a) referring the register by mnemonic and specifying the bit number,
   b) referring the register by address and specifying

the bit number,

c) referring the bit address*,

d) referring the bit-addressable register by mnemonic*.

Example: Assemble a program instruction using all the methods listed above to refer a bit-addressable register.

```
>A
LOC     OBJ          LINE     LABEL    SOURCE CODE
0000    92D4         0001              MOV PSW.4,C
0002    92D4         0002              MOV D0.4,C
0004    92D4         0003              MOV D4,C
0006    92D4         0004              MOV .RS1,C
0008                                   <ESC>
>
```

* Refer to Special Function Register Bit Address, figure 7-4 for 8031 family or figure 18-7 for 8344 family, Intel Microcontroller Handbook, 1984.

3. Disassembly will always display bit-addressable registers by mnemonic (where a mnemonic has been defined by the manufacturer), regardless of the method used to specify them during assembly.

Example: Disassemble the program instructions from the preceding example.

```
>Z0 6
LOC     OBJ          LINE     LABEL    SOURCE CODE
0000    92D4         0001              MOV .RS1,C
0002    92D4         0002              MOV .RS1,C
0004    92D4         0003              MOV .RS1,C
0006    92D4         0004              MOV .RS1,C
     DISASSEMBLY COMPLETED
>
```

Referring Special Function Registers for NEW MICE-II 8052/80C152

1. There are two different ways to refer special function registers in A/Z commands:

   a) Referring the register by mnemonic*.
   b) Referring the register by address*.

LOC	OBJ	LINE	LABEL	SOURCE CODE
0000	85E0D0	0001		MOV PSW,ACC
0002	85E0D0	0002		MOV D0,ACC

   * Refer to Memory Organization, section 6.1 for 8051 family or figure 18-6 for 8344 family, Intel Microcontroller Handbook, 1984.

2. There are four different ways to refer bit addresses for special function registers in the Assembly command.

   a) referring the register by mnemonic and specifying the bit number,
   b) referring the register by address and specifying the bit number,
   c) referring the bit address*,
   d) referring the bit-addressable register by mnemonic*.

Example: Assemble a program instruction using all the methods listed above to refer a bit-addressable register.

```
>A
LOC     OBJ          LINE     LABEL    SOURCE CODE
0000    92D4         0001              MOV PSW.4,C
0002    92D4         0002              MOV D0.4,C
0004    92D4         0003              MOV D4,C
0006    92D4         0004              MOV .RS1,C
0008                                   <ESC>
>
```

* Refer to Special Function Register Bit Address, figure 7-4 for 8051 family or figure 18-7 for 8344 family, Intel Microcontroller Handbook, 1984.

3. Disassembly will always display bit-addressable registers by mnemonic (where a mnemonic has been defined by the manufacturer), regardless of the method used to specify them during assembly.

   Example: Disassemble the program instructions from the preceding example.

```
>Z0 6
LOC     OBJ          LINE     LABEL    SOURCE CODE
0000    92D4         0001              MOV .RS1,C
0002    92D4         0002              MOV .RS1,C
0004    92D4         0003              MOV .RS1,C
0006    92D4         0004              MOV .RS1,C
     DISASSEMBLY COMPLETED
>
```

Referring Special Function Registers for NEW MICE-II 80515/80535

1. There are two different ways to refer Special Function Registers (SFR) in A/Z commands:

   a) Referring the register by mnemonic.*
   b) Referring the register by address.

```
>A0
LOC      OBJ              LINE   LABEL  SOURCE CODE
0000     8556D0           0001           MOV PSW,56
0003     8556D0           0002           MOV D0,56
0006                      0003           <ESC>
>Z0 3
LOC      OBJ              LINE   LABEL  SOURCE CODE
0000     8556D0           0001           MOV    PSW,56
0003     8556D0           0002           MOV    PSW,56
     DISASSEMBLY COMPLETED
>
```

   *Refer to Table 1, Special Function Registers, Siemens Microcomputer Components, SAB 80515/80535 Single-Chip Microcontroller User's Manual, 7/85.

2. There are four different ways to refer direct-addressable bits for SFR's in the Assembly command:

   a) referring the register by mnemonic and specifying the bit number,
   b) referring the register by address and specifying the bit number,
   c) referring the bit address*,
   d) referring the bit by mnemonic*.

Example: Assemble a program instruction using all the method mentioned above to refer a direct-addressable bit for SFR's.

```
>A0
LOC     OBJ          LINE    LABEL    SOURCE CODE
0000    92D4         0001             MOV PSW.4,C
0002    92D4         0002             MOV D0.4,C
0004    92D4         0003             MOV D4,C
0006    92D4         0004             MOV .RS1,C
0008                 0005             <ESC>
>
```

*Refer to Figure 52, Special Function Register Bit-Addressable Locations, Siemens Microcomputer Components, SAB 80515/80535 Single-Chip Microcontroller User's Manual, 7/85.

3. Disassembly will always display direct-addressable bits for SFR's by mnemonic (where mnemonics have been defined by the manufacturer), regardless of the method used to specify them during assembly.

Example: Disassemble the program instructions from the preceding example.

```
>Z0 6
LOC     OBJ          LINE    LABEL    SOURCE CODE
0000    92D4         0001             MOV    .RS1,C
0002    92D4         0002             MOV    .RS1,C
0004    92D4         0003             MOV    .RS1,C
0006    92D4         0004             MOV    .RS1,C
     DISASSEMBLY COMPLETED
>
```

5.1 Memory Display/Examine/Modify/Fill/Search Command - M

```
8085  - M[start-address[ end-address[ data-1[ ...data-8][ S]]]]
8048  - M[[P|X|I][ a1[ a2[ d1[..d8]][ S]]]]
8031  - M[[P|X|I][ a1[ a2[ d1[..d8][ S]]]]|V[ a1[ a2[ d1[..d8] S]]]]
8052  - M[[P|X|I][ a1[ a2[ d1[..d8][ S]]]]|V[ a1[ a2[ d1[..d8] S]]]]
80515 - M[[P|X|I][ a1[ a2[ d1[..d8][ S]]]]|V[ a1[ a2[ d1[..d8] S]]]]
```

M	is the command for Memory Display/ Examine/Modify/Fill/Search. If no other parameters are specified, inputting "M<CR>" will display the next 256 bytes starting at the current PC.
P|X|I|V	are memory type designations for micro-controllers. (Refer to section 2.7 for a detailed description of memory types. For the 8031/8344, 8052/80C152 and 80515/ 80535, also refer to the application notes at the beginning of this chapter on memory type specification.)
start-address	is a hexadecimal address of the emulation CPU indicating a memory location where the operation is to begin.
end-address	is a hexadecimal address of the emulation CPU indicating the last memory location of the specified range.
data-1...8	are hexadecimal or ASCII data. Data in ASCII must be enclosed within two apostrophes (e.g. 'AR'). Where a "block fill" or "block search" operation is specified, the block may be up to 8

bytes in hex or 8 characters in ASCII. Combined use of hex and ASCII is permitted. If the address range (from start to end-address) is smaller than the block size (data1 - data8), then block-fill is still executed but excess trailing data is ignored; for block-search, however, **"MEMORY SEARCH FAILURE!"** will display. (Note that the apostrophe ['], lower case "r" and MICE editing/control characters are not allowed for ASCII input.)

S — defines a "search" option which looks for specified data.

Where the start and end-address are defined for a memory range in the emulation processor, the end-address must be greater than or equal to the start-address or an error message is printed and the command ends.

5.1.1 Memory Display

M[start-address[end-address]]

Input a start and end-address to display the content of a memory range. Memory contents are displayed in hexadecimal and ASCII. (Note that data without an ASCII equivalent is indicated by a period.) The end-address must be greater than or equal to the start-address or an error message is printed and the command ends. The display can be terminated by entering an <ESC>.

Example: Display program memory contents 0H to 16H.

```
>M0 16
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
0000  31 F0 7F 21 34 12 E5 21 21 43 0E 06 71 23 E3 0D  1..!4..!!C..q#..
0010  C2 0C 00 E1 C3 03 00                            .......
>
```

5.1.2 Memory Examine/Modify

M start-address

If only the start-address is specified, the content of that memory location is first displayed. NEW MICE-II then waits for input. To advance to the next memory address, a <CR> should be entered; the next memory location's content is then displayed, and NEW MICE-II again waits. To go back to the previous memory address a <LF> should be entered; the previous memory location's content is then displayed, and NEW MICE-II again waits. Entering a <CR> or <LF> repeats the entire sequence. If an <ESC> is entered, the command ends.

To change the memory content displayed, enter a new value and a <CR> or <LF>. (Note that the apostrophe ['], lower case "r" and MICE editing/control characters are not allowed for ASCII input.) After writing data to memory, verification of memory starts automatically. If any data does not match, the following message displays: "MEMORY VERIFICATION FAILURE!".

Example: Examine and modify the memory contents from address location 6H, then display the results.

```
>M6
 0006 F1 E5<CR>
 0007 FF 21<CR>
 0008 00 21<CR>
 0009 43 <LF>
 0008 21 <ESC>
>M0 16
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
0000  31 F0 7F 21 34 12 E5 21 21 43 0E 06 71 23 E3 0D  1..!4..!!C..q#..
0010  C2 0C 00 E1 C3 03 00                             .......
>
```

5.1.3 Memory Fill

M start-address end-address data-1[data-2[...data-8]]

Input a start and end-address for the memory range to be filled. A specified data value or data block may be written into the defined range.

Example: Write the ASCII value "NEW DATA" into the memory range 17H to 2FH and t h e display the memory contents for the range 0H to 2FH.

```
>M17 2F 'NEW DATA'
>M0 2F
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
0000  31 F0 7F 21 34 12 E5 21 21 43 0E 06 71 23 E3 0D  1..!4..!!C..q#..
0010  C2 0C 00 E1 C3 03 00 4E 45 57 20 44 41 54 41 4E  .......NEW DATAN
0020  45 47 20 44 41 54 41 4E 45 57 20 44 41 54 41 4E  EW DATANEW DATAN
>
```

5.1.4 Memory Search

M start-address end-address data-1[data-2[...data-8]] S

Input a start and end-address for the memory range to be searched. A specified data value or data block may be searched for within the defined range.

Example: Search for string "BA 10" in the range 0H to 1FH, which is not found; then search for string 'NEW' in the same address range.

```
>M0 1F
     00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
0000  31 F0 7F 21 34 12 E5 21 21 43 0E 06 71 23 E3 0D  1..!4..!!C..q#..
0010  C2 0C 00 E1 C3 03 00 4E 45 57 20 44 41 54 41 4E  .......NEW DATAN
>
>M0 1F BA 10 S
 MEMORY SEARCH FAILURE !
>M0 1F 'NEW' S
 MEMORY MATCH AT ADDRESS 0017
>
```

5.2 Memory Checksum/Test/Transfer/Compare Command - T

```
8085  - T start-address end-address {S|M|address-3[ V]}
8048  - T {M|[P|X|I] a1 a2 {S|M|a3[ V]}}
8031  - T {[P|X|I] a1 a2 {S|M|a3[ V]}|V a1 a2 {S|a3[ V]}}
8052  - T {[P|X|I] a1 a2 {S|M|a3[ V]}|V a1 a2 {S|a3[ V]}}
80515 - T {[P|X|I] a1 a2 {S|M|a3[ V]}|V a1 a2 {S|a3[ V]}}
```

T is the command for Memory Checksum/Test/Transfer/Compare.

M is a memory transfer command for NEW MICE-II 8048 which moves the Target Program to MICE Program Memory at the same location. Remember that after executing this command, the external program memory access control signal will be disabled (DA).

P|X|I|V are memory type designations for microcontrollers. (Refer to section 2.6 for a detailed description of memory types. For the 8031/8344, 8052/80C152 and 80515/80535, also refer to the application notes at the beginning of this chapter on memory type specification.)

start-address is a hexadecimal address of the emulation CPU indicating a memory location where the operation is to begin.

end-address is a hexadecimal address of the emulation CPU indicating the last memory location of the specified range.

S	displays the checksum of the contents in the specified range.
M	performs a memory test for the specified range.
address-3	is a hexadecimal address in the emulation CPU specifying either a destination address, where data is to be transferred, or the start of a second memory range used in "block compare" (section 5.2.4).
V	specifies "block compare".

Define the start and end-address for a memory range in the emulation CPU. The end-address must be greater than or equal to the start-address or an error message is printed and the command ends.

5.2.1 Memory Checksum

T start-address end-address S

An "S" following the range specification causes the checksum to be displayed. The checksum is calculated by taking the hexadecimal sum of the contents for the indicated range, with carry added back, modulo 256.

Example: First display the program memory contents for the range 5H to AH, and then calculate the checksum for the same range.

```
>M5 A
     00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
0000                 12 E5 21 21 43 0E                    ..!!C.
>T5 A S
THE CHECKSUM IS: 8B
>
```

5.2.2 Memory Test

T start-address end-address M

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 (XOR) and written into memory for the entire range. The data are verified and then their complements are written into the entire range and thoroughly checked again. If the comparison fails in either pass, the test is stopped at the failed address and that address is displayed. The original contents in memory are destroyed in this test.

Example: Test the memory range 19H through 23H, and then display the memory contents for the range 10H to 2FH.

```
>M10 2F
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
0010  C2 0C 00 E1 C3 03 00 4E 45 57 20 44 41 54 41 4E  .......NEW DATAN
0020  45 47 20 44 41 54 41 4E 45 57 20 44 41 54 41 4E  EW DATANEW DATAN
>T19 23 M
 RAM OK!
>M10 2F
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
0010  C2 0C 00 E1 C3 03 00 4E 45 E6 E5 E4 E3 E2 E1 E0  .......NE.......
0020  DF DE DD DC 41 54 41 4E 45 57 20 44 41 54 41 4E  ....ATANEW DATAN
>
```

Example: Test the memory range 3FF0H through 400FH, where memory is enabled only through 3FFFH.

```
>T3FF0 400F M
ADDRESS (4000) RAM ERROR!
>M3FF0 400F
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
3FF0  30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F  0123456789:;<=>?
4000  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
>
```

5.2.3 Memory Transfer

8048 - T M
all others - T start-address end-address address-3

An address-3 (dest-address) specification indicates that the memory content in the defined range is to be transferred into memory beginning at address-3. Data is transferred, one location at a time beginning at the start address of the destination range. If the memory ranges defined in the transfer command overlap, the transfer will begin at the end address of the destination range to prevent overwrite.

Example: First display the memory content for the range 0H to 2FH, then transfer the memory content in the range 5H through AH to memory beginning at 1AH. Finally, display the results of the transfer.

```
>M0 2F
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
0000  31 F0 7F 21 34 12 E5 21 21 43 0E 06 71 23 E3 0D  1..!4..!!C..q#..
0010  C2 0C 00 E1 C3 03 00 4E 45 57 20 44 41 54 41 4E  .......NEW DATAN
0020  45 47 20 44 41 54 41 4E 45 57 20 44 41 54 41 4E  EW DATANEW DATAN
>T5 A 1A
>M0 2F
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
0000  31 F0 7F 21 34 12 E5 21 21 43 0E 06 71 23 E3 0D  1..!4..!!C..q#..
0010  C2 0C 00 E1 C3 03 00 4E 45 57 12 E5 21 21 43 0E  .......NEW..!!C.
0020  45 47 20 44 41 54 41 4E 45 57 20 44 41 54 41 4E  EW DATANEW DATAN
>
```

Example: For NEW MICE-II 8048, move target program memory to internal (4K byte) emulation memory and then modify it. Remember that this operation transfers the whole program memory (4K bytes) at one time.

```
>TM  (Data read in)
>MO
P-0000 00 04<CR>
P-0001<ESC>
>
```

5.2.4 Memory Compare

T start-address end-address address-3 V

A "V" following the address-3 specification executes block compare for the indicated memory ranges. The block from start-address to end-address is compared with the block beginning at address-3. If comparison is successful, "**COMPARISON OK!**" will display; otherwise the failure addresses along with the incorrect data are displayed.

Example: First display the memory contents from 0H to 2FH, and compare the data block from 5H to AH with the address range beginning at 1AH.

```
>M0 2F
     00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
0000  31 F0 7F 21 34 12 E5 21 21 43 0E 06 71 23 E3 0D  1..!4..!!C..q#..
0010  C2 0C 00 E1 C3 03 00 4E 45 57 12 E5 21 21 43 0E  .......NEW..!!C.
0020  45 47 20 44 41 54 41 4E 45 57 20 44 41 54 41 4E  EW DATANEW DATAN
>T5 A 1A V
 COMPARISON OK!
>
```

Example: Compare the data block from 5H to AH with the address range beginning at 11H.

```
>T5 A 11 V
(0005)=12  ;(block 1 failure address) = data 1
(0011)=0C  ;(block 2 failure address) = data 2
>
```

5.3 Line Assembly Command - A

A[start-address]

A	is the command for Assembly. If no other parameters are specified, inputting "A<CR>" will execute assembly beginning at the current PC.
start-address	is the hexadecimal address in the emulation processor's program memory where MICE begins storing the entered assembly language program. Note that only program memory can be accessed (i.e. memory type "P") when using the Assembly command for the 8031/8344, 8052/80C152 or 80515/80535. (Refer to section 2.6 for a detailed description of memory types).

Rather than enter programs or program changes in machine code using the previously described memory (M) command, the NEW MICE-II resident assembler accepts and converts mnemonic input into machine code. The converted code is then stored in the emulation processor's program memory starting at the indicated start-address. The assembler does not recognize symbolic labels or constants other than hexadecimal values. The instruction mnemonics accepted are those adopted by the original manufacturer for the processor being emulated. In addition to these mnemonic codes (listed in the appendices), the following three instructions are also supported during assembly.

DB -	Define Byte	(1-6 bytes)
DW -	Define Word	(1 word)*
DS -	Define Storage	(0H-FFFFH)

* NEW MICE-II 8048 can only accept a hex value from 0H-FFFH for DW.

DB accepts both hex and ASCII codes, while DW and DS are restricted to hex data. (Note that the apostrophe ['], lower case "r" and MICE editing/control characters are not allowed for ASCII input.) If more than 6 bytes are keyed in for DB, excess data is truncated on the right and NEW MICE-II displays: "**WARNING: ONLY 6 BYTES ARE VALID!**". If more than 1 word (4 digits) are keyed in for DW or DS, the input data is ignored and "**ERROR CODE, TRY AGAIN!**" is displayed.

After inputting the assembly command, NEW MICE-II displays the following column headings:

```
>A0
LOC      OBJ     LINE      LABEL          SOURCE CODE
0000             0001                     *
```

where the decimal value of 0001 under the column "LINE" indicates the line number being entered and the "*" indicates the new cursor position. NEW MICE-II then waits for an assembly language line input from the user. The source code column consists of an opcode and operand, and must be terminated with either a <CR> or <LF>. NEW MICE-II assembles the line and stores the machine code beginning at the indicated start-address. The code for the program memory location to be stored is printed under the column "OBJ". The next line number is printed and NEW MICE-II again waits. If an <ESC> is entered instead, the command ends.

NEW MICE-II performs data verification after executing a memory WRITE. If any data does not match, the following message displays: "**MEMORY WRITE FAILURE!**". If an invalid program instruction is entered, NEW MICE-II displays: "**ERROR CODE, TRY AGAIN!**".

Example: Enter a simple program starting at location OH.

```
>A0
LOC     OBJ      LINE    LABEL    SOURCE CODE
0000    31F07F   0001             LXI SP,7FF0<CR>
0003    213412   0002             LXI H,1234<CR>
0006    E5       0003             PUSH H<CR>
0007    212143   0004             LXI H,4321<CR>
000A    0E06     0005             MVI C,06<CR>
000C    71       0006             MOV M,C<CR>
000D    23       0007             INX H<CR>
000E    E3       0008             XTHL<CR>
000F    0D       0009             DCR C<CR>
0010    C20C00   0010             JNZ 000C<CR>
0013    E1       0011             POP H<CR>
0014    C30300   0012             JMP 0003<CR>
0017             0013             <ESC>
>
```

Example: An error occurs when an incorrect mnemonic is entered. In this case, non-hexadecimal data cannot be used in the data field.

```
>A3F
LOC          OBJ        LINE    LABEL    SOURCE CODE
003F                    0001             LVI SP,7FF0<CR>
ERROR CODE, TRY AGAIN!
003F         31F07F     0001             LXI SP,7FF0<CR>
0040                    0002             <ESC>
>
```

When making changes in an existing program, it is advisable to check the program around the modified area using the Disassembly command (section 5.4) before and after the change. Rechecking the modified area is important to assure that new program changes do not affect the surrounding program.

5.4 Disassembly Command - Z

8085 & 8048 - **Z[start-address[end-address]]**
8031 & 8052 & 80515 - **Z[V][a1[a2]]**

Z — is the command for Disassembly. If no other parameters are specified, inputting "Z<CR>" will execute disassembly for the next 16 lines starting at the current PC.

V — is the 8031/8344, 8052/80C152 (with PVPOD-8052) and 80515/80535 memory type designation for on-chip internal program memory. The disassembly command can access either program memory (default is memory type "P") or on-chip internal program memory. Remember that the address specification for memory type V must be within 0H~1FFFH. (See section 2.6.2 for a detailed description of memory types. Also refer to the application notes at the beginning of this chapter on memory type specification.)

start-address — is a hexadecimal address in the emulation processor's program memory where display of disassembled memory content begins. Remember that either program memory or the 80515 on-chip internal program ROM (at U62) can be accessed when using the Disassembly command for the 80515/80535.

end-address is the hexadecimal address in the emulation processor's program memory indicating the last memory location of the range to be disassembled and displayed.

If only the start-address is specified, the content for that memory location is disassembled and displayed; more data are then read from subsequent locations to complete the instruction if necessary. 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 (**INPUT ADDRESS ERROR**) and the command ends. If an illegal machine code is encountered, disassembly terminates at the illegal code's address.

NEW MICE-II uses a two-pass disassembler with all branch and subroutine call addresses first identified and converted to labels; a total of 900 labels can be stored for disassembly. Depending on the range to be disassembled, there may be a pause before any data is displayed.

Example: Disassemble the previously entered program. Note that the byte following the end-address is also read in order to complete the instruction for this example. Labels include a prefix (B for branch or S for subroutine) and address.

```
>Z0 16
LOC     OBJ      LINE   LABEL    SOURCE CODE
0000    31F07F   0001            LXI SP,7FF0
0003    213412   0002    B0003   LXI H,1234
0006    E5       0003            PUSH H
0007    212143   0004            LXI H,4321
000A    0E06     0005            MVI C,06
000C    71       0006    B000C   MOV M,C
000D    23       0007            INX H
000E    E3       0008            XTHL
000F    0D       0009            DCR C
0010    C20C00   0010            JNZ 000C
0013    E1       0011            POP H
0014    C30300   0012            JMP 0003
     DISASSEMBLY COMPLETED
>
```

Example: If an illegal machine code is encountered, disassembly terminates at the illegal code's address.

```
>Z3F 5C
LOC     OBJ      LINE   LABEL    SOURCE CODE
003F    31F07F   0001            LXI SP,7FF0
0040    100E14   0002
        ERROR CODE
>
```

5.5 Port Input Command - I

I port[count[duration]]

I is the command for Port Input.

port is the hexadecimal address of the emulation processor's input port that is to be read and displayed. The following microcontrollers/ microcomputers are limited to ports indicated below.

NEW MICE-II	Accessible Port
8048	0,1,2,4,5,6,7
8031/8344	0,1,2,3
8052	0,1,2,3
80C152JA/83C152JA	0,1,2,3,4
80C152JB	0,1,2,3,4,5,6
80515/80535	0,1,2,3,4,5

count is a hexadecimal value from 00H to FFH specifying the number of times the input port is to be read, with 00H indicating 256 times.

duration is a hexadecimal value from 00H to FFH specifying the interval in milliseconds between each read, with 00H indicating 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 indicated port address.

NEW MICE-II first reads and displays the contents of the port specified and waits for input from the user. To advance to the next port, enter either a <CR> or <LF>; the next port's content is then read and displayed, and NEW MICE-II again waits. Entering a <CR> or <LF> repeats the entire sequence.

Example: Read and display the contents of ports 80, 81 and 82.

```
>I80
 80 DC<CR>
 81 87<LF>
 82 FF<ESC>
>
```

Specifying a count with or without a duration interval, NEW MICE-II first reads from the specified input port the number of times indicated and then displays the contents.

Duration defines the interval between each read in milliseconds, permitting compensation for data inputs which are time critical. If no interval is specified, then data is read at one millisecond intervals.

Example: Read and display the next 64 (40H) values for port D0H at 10 (0AH) millisecond intervals.

```
>ID0 40 A
PORT 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
D0   36 22 AE 9B 2B 18 C7 77 26 D6 86 35 E5 95 44 F4  6"..+..w&..5..D.
     A3 53 03 B2 62 12 C1 71 20 D0 80 2F DF 8E 3E EE  .S..b..q ../..>.
     9D BD FD AC 5C 0B BB 6B 1A CA 7A 29 D9 88 38 E8  .M..\..k..z)..8.
     97 47 F6 A6 56 05 B5 65 14 C4 73 23 D3 82 32 E0  .G..V..e..s#..2.
>
```

Note that the interval can be up to 256 milliseconds (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 displayed is approximately 65 seconds. Therefore, to end the command before display begins, enter an <ESC>.

5.6 Port Output Command - O

O port data-1[data-2[...data-8]]

O is the command for Port Output.

port is a hexadecimal address of the emulation CPU's output port. The following microcontrollers/microcomputers are limited to ports indicated below.

NEW MICE-II	Accessible Port
8048	0,1,2,4,5,6,7
8031/8344	0,1,2,3
8052	0,1,2,3
80C152JA/83C152JA	0,1,2,3,4
80C152JB	0,1,2,3,4,5,6
80515/80535	0,1,2,3,4,5

data-1...8 are hexadecimal values to be written into the specified output port of the emulation processor in sequential order.

If only one value is specified, it is written to the indicated output port. If more than one value is specified, each value is written to the indicated port at one millisecond intervals with the first value going out immediately.

Example: Write the value 48H to port 01H.

```
>O 1 48
>
```

Example: Write values 1, 2 and 3 to port 05H. Value 1 is written immediately followed by value 2 after one millisecond and value 3 after another millisecond.

```
>0 5 1 2 3
>
```

5.7 Reset/Initialization Command - X

X[address]

X is the command for Reset/Initialization.

address is a hexadecimal addresses in the emulation CPU's program memory where emulation is to begin. (NEW MICE-II 8048 and 8085 do not support an optional address.)

A signal pulse is generated on the appropriate pin of the emulation processor to reset it. The processor's program counter, internal registers and flags are all altered. A start address can be defined which sets the emulation processor's program counter to the specified program memory address. Only the emulation processor is reset; any logic or peripherals connected to the target input reset pin are not affected.

In addition, control and interrupt signals to the emulation processor that are selectively controllable by the Disable and Enable commands are activated if MICE is connected to a power-applied target system. However, if no target system is connected to MICE, those signals will be disabled.

Example: Reset the emulation processor, showing changes in the program counter (PC) and registers before and after.

```
>R
   A   B   C   D   E   H   L   M   P     S   PC
  00  00  00  00  00  00  00  07  00  AAAA 0010
>X
>R
   A   B   C   D   E   H   L   M   P     S   PC
  00  00  00  00  00  00  00  07  00  AAAA 0000
>
```

5.8 Register Display/Modify Command - R

5.8.1 Register Command for NEW MICE-II 8085 and 8048

R[register]

R is the command for Register Display/ Modification.

register are alphanumeric characters indicating the target register whose content is to be displayed or modified. Register mnemonics and sizes depend on the target being emulated. The register set that can be displayed and/or changed by the R command is listed in the following tables.

NEW MICE-II 8085 Register Set

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 bits

NEW MICE-II 8048 Register Set

The 8048 family of processors support two sets of registers. The current bank (set) is displayed and changed using the "R" command. Input "RBØ" to select bank Ø, or "RB1" to select bank 1. The registers are listed in the following table; where alternates are available, they are indicated under the mnemonic heading with an apostrophe.

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)
*PC	Program Counter	12 bits
*MB	CPU DBF Flag	1 bit
*CY	Carry Flag	1 bit
*AC	Auxiliary Carry	1 bit
*F0	Flag 0	1 bit
*BS	Register Bank Select	1 bit
*F1	Flag 1	1 bit
*SP	Stack Pointer	3 bits

* These registers cannot be changed by the R command.

5.8.1.1 Display Registers

R

All register contents are displayed if no specific register is defined. Note that the displayed value of the program counter (PC) always indicates the address for the current instruction's opcode.

Example: Display register contents.

```
>R
   A   B   C   D   E   H   L   M   P     S   PC
  00  00  00  00  00  00  00  07  00  AAAA 0000
>
```

5.8.1.2 Modify Registers

R register

If a register is specified, the content of that register is first displayed; NEW MICE-II then waits for input. To advance to the next register enter either a <CR> or <LF>; the next register's content is then displayed and NEW MICE-II again waits. Entering a <CR> or <LF> repeats the entire sequence unless the next register to display is "PC". The J command (section 5.9) must be used to change the program counter. If an <ESC> is entered, the command ends. To change the content of the displayed register, enter a new value in hex or ASCII (enclosed with two apostrophes, e.g. 'A') and <CR> or <LF>.

Example: Examine the specified register and change its content.

```
>RA
 A 00 02<CR>
 B 00 03<CR>
 C 00 <ESC>
>R
   A   B   C   D   E   H   L   M   P     S   PC
  02  03  00  00  00  00  00  07  00  AAAA 0000
>
```

5.8.2 Register Command for NEW MICE-II 8031/8344

R[S|register|BIT[.symbol| start-address[end-address[data-1[...data-8]]]]|INT]

The NEW MICE-II 8031/8344 supports display/modification of registers and direct-addressable bits, as well as display of interrupt enable and pending status:

R	is the Register command. If no other parameters are specified, contents of the ACC, B, DPTR, PSW, SP, current bank register and the PC are displayed. Note that the PSW displays both hex and binary values for the current bank registers.
S	displays the contents of all Special Function Registers, except for those listed above.
register	are alphabetic characters indicating Special Function Registers, or bank registers, whose content is to be displayed or modified. Register mnemonics and addresses are listed in section 2.7.2.1. (Note that all of the registers listed are 8-bit registers.)
BIT	indicates a specified operation for direct-addressable bits.

.symbol	is the mnemonic for a bit-addressable register location. Note that all mnemonics must be prefixed by a period ".". (Refer to Special Function Register Bit Address, figure 7-4 for the 8031 family or figure 18-7 for the 8344 family, Intel Microcontroller Handbook, 1984.)
start-address	is a hexadecimal pointer for a bit-addressable register in the emulation CPU indicating a memory location where the operation is to begin. The permissible address range is 0~FFH.
end-address	is a hexadecimal pointer for a bit-addressable register in the emulation CPU indicating the last memory location of the specified range. The permissible address range is 0~FFH.
data-1...8	are binary values of "0" or "1". Note that if the address range (from start to end-address) is smaller than the block size (data1 - data8), then block fill is still executed but excess trailing data is ignored.
INT	displays interrupt enable and pending status.

5.8.2.1 Available Registers

The following is a list of accessible registers for the 8031.

Mnemonic	Register	Address
ACC	Accumulator	0E0H
B	B Register	0F0H
PSW	Program Status	0D0H
SP	Stack Pointer	81H
DPH DPTR	Data Pointer High Byte	83H
DPL	Data Pointer Low Byte	82H
IP	Interrupt Priority Control	0B8H
IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
TCON	Timer/Counter Control	88H
TH0	Timer/Counter 0 (high byte)	8CH
TL0	Timer/Counter 0 (low byte)	8AH
TH1	Timer/Counter 1 (high byte)	8DH
TL1	Timer/Counter 1 (low byte)	8BH
SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

In addition to the above registers, the 8032 also includes the following registers.

Mnemonic	Register	Address
T2CON	Timer/Counter 2 Control	0C8H
TH2	Timer/Counter 2 (high byte)	0CDH
TL2	Timer/Counter 2 (low byte)	0CCH
CAP2H	Timer/Counter 2 Capture (high byte)	0CBH
CAP2L	Timer/Counter 2 Capture (low byte)	0CAH

The following are accessible registers for the 8344.

Mnemonic	Register	Address
B	B Register	F0H
ACC	Accumulator	E0H
FIFOH	FIFO	DFH
FIFOM	FIFO	DEH
FIFOL	FIFO	DDH
TBS	Transmit Buffer Start	DCH
TBL	Transmit Buffer Length	DBH
TCB	Transmit Control Bytes	DAH
SIUST	SIU State Counter	D9H
NSNR	Send Count Receive Count	D8H
PSW	Program Status Word	D0H
DMA	DMA Count	CFH
STAD	Station Address	CEH
RFL	Receive Field Length	CDH
RBS	Receive Buffer Start	CCH
RBL	Receive Buffer Length	CBH
RCB	Receive Control Byte	CAH
SMD	Serial Mode	C9H
STS	Status Register	C8H
IP	Interrupt Priority Control	B8H
IE	Interrupt Enable Control	A8H
TH1	Timer High 1	8DH
TH0	Timer High 0	8CH
TL1	Timer Low 1	8BH
TL0	Timer Low 0	8AH
TMOD	Timer Mode	89H
TCON	Timer Control	88H
DPH DPTR	Data Pointer High	83H
DPL	Data Pointer Low	82H
SP	Stack Pointer	81H

5.8.2.2 Display/Modify Registers

R[S|register]

Example: Display contents for ACC, B, PSW, SP, DPTR, current bank registers and the PC. **(The numbers in parentheses are register addresses.)**

```
>R
  ACC   B   PSW   SP DPH DPL R0 R1 R2 R3 R4 R5 R6 R7     PSW     PC
  (E0)(F0)(D0)(81)(83)(82)                              CAFBBO-P
   12   23   34   45   56   67 31 32 33 34 35 36 37 38 00110100 0234
>
```

Example: Display contents for the SFR registers not listed in the example above for the 8031.

```
>RS
   IP     IE    TMOD   TCON   TH0    TL0    TH1    TL1    SCON   SBUF   PCON
  (B8)   (A8)   (89)   (88)   (8C)   (8A)   (8D)   (8B)   (98)   (99)   (87)
   F1     62     33     C4     14     66     77     88     9B     FF     FF
>
```

Example: Display contents for the SFR registers not listed in the example above for the 8032.

```
>RS
   IP     IE    TMOD   TCON   TH0    TL0    TH1    TL1
  (B8)   (A8)   (89)   (88)   (8C)   (8A)   (8D)   (8B)
   D1     62     33     C4     B0     66     77     88
 T2CON  TH2    TL2    CAP2H  CAP2L  SCON   SBUF   PCON
  (C8)   (CD)   (CC)   (CB)   (CA)   (98)   (99)   (87)
   99     12     23     34     45     56     00     FF
>
```

Example: Display contents for the SFR registers not listed in the preceding examples for the 8344.

```
>RS
   IP   IE  TMOD  TCON TH0  TL0  TH1  TL1  FIFOH FIFOM FIFOL TBS
  (B8) (A8) (89)  (88) (8C) (8A) (8D) (8B) (DF)  (DE)  (DD)  (DC)
   E0   60   00    00   00   00   00   00   F9    99    93    1F
  TBL  TCB  SIUST NSN4 DMA  STAD RFL  RBS  RBL   RCB   SMD   STS
  (DB) (DA) (D9)  (D8) (CF) (CE) (CD) (CC) (CB)  (CA)  (C9)  (C8)
   12   37   01    94   14   63   24   87   12    77    24    68
>
```

Example: Modify the contents of the accumulator.

```
>RACC<CR>
 ACC  (E0) 12 5A<CR>
 B    (F0) 23<ESC>
>
```

5.8.2.3 Display/Modify Bit-Addressable Locations

```
RBIT[.symbol| start-address[ end-address[ data-1[ ...data-8]]]]
```

Example: Display all direct-addressable bits for the 8031. The dash symbol "-" is used where no bit-addressable register exists for the indicated address.

```
>RBIT
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00  0  0  1  0  0  0  1  0  1  0  0  0  1  0  1  0
10  1  0  1  0  1  0  1  0  0  0  1  0  0  0  1  0
20  1  0  0  0  1  0  1  0  1  0  0  0  1  0  1  0
30  0  0  1  0  1  0  1  0  1  0  1  0  0  0  0  0
40  1  0  1  0  0  0  1  0  0  0  0  0  1  0  1  0
50  1  0  0  0  1  0  1  0  1  0  1  0  0  0  1  0
60  1  0  1  0  0  0  1  0  0  0  0  0  1  0  0  0
70  0  0  1  0  0  0  1  0  0  0  0  0  1  0  1  0
80  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
90  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0
A0  1  1  1  1  1  1  1  1  0  0  0  0  0  -  -  0
B0  1  1  1  1  1  1  1  1  0  0  0  0  0  -  -  -
C0  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
D0  0  0  0  0  0  0  0  0  -  -  -  -  -  -  -  -
E0  0  0  0  0  0  0  0  0  -  -  -  -  -  -  -  -
F0  0  0  0  0  0  0  0  0  -  -  -  -  -  -  -  -
>
```

Example: Display all direct-addressable bits for the 8032.

```
>RBIT
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00  0  0  1  1  0  0  1  0  0  0  0  1  0  0  0  1
10  0  1  1  1  1  0  1  1  0  1  1  1  0  1  1  1
20  0  0  1  1  0  0  1  1  0  0  1  1  0  1  1  1
30  1  0  1  1  1  1  0  1  1  0  0  0  0  0  0  1
40  0  0  0  1  0  1  0  1  0  0  1  0  0  0  0  1
50  0  0  0  1  0  0  0  0  0  0  1  1  0  0  0  1
60  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0
70  0  1  0  1  0  0  1  1  0  1  1  1  0  1  1  0
80  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
90  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0
A0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  -  0
B0  1  1  1  1  1  1  1  1  0  0  0  0  0  0  -  -
C0  -  -  -  -  -  -  -  -  0  0  0  0  0  0  0  0
D0  0  0  0  0  0  0  0  0  -  -  -  -  -  -  -  -
E0  0  0  0  0  0  0  0  0  -  -  -  -  -  -  -  -
F0  0  0  0  0  0  0  0  0  -  -  -  -  -  -  -  -
>
```

Example: Display all direct-addressable bits for the 8344.

```
>RBIT
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00  1  0  1  1  0  0  1  1  0  0  0  1  0  1  1  1
10  1  0  0  0  1  0  0  1  1  0  0  1  1  0  1  1
20  1  0  0  0  0  1  1  1  1  1  1  0  1  0  0  0
30  0  0  1  1  1  0  1  0  1  0  1  0  1  0  1  1
40  1  1  0  0  1  0  0  0  1  1  0  1  1  0  0  1
50  1  0  0  0  0  1  1  0  0  0  1  0  1  0  1  0
60  1  1  1  1  0  0  0  1  0  1  0  1  1  0  0  0
70  0  0  0  0  1  1  1  0  0  0  0  0  1  0  1  0
80  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
90  1  1  1  1  1  1  1  1  -  -  -  -  -  -  -  -
A0  0  1  0  0  0  0  0  0  0  0  0  0  0  -  -  0
B0  1  1  1  1  1  1  1  1  0  0  0  0  0  -  -  -
C0  -  -  -  -  -  -  -  -  0  0  0  0  0  0  0  0
D0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
E0  0  0  0  0  0  0  0  0  -  -  -  -  -  -  -  -
F0  0  0  0  0  0  0  0  0  -  -  -  -  -  -  -  -
>
```

Example: Modify the bit register EA, displaying the affected area both before and after register modification.

```
>RBITA0 AF
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
A0  0  0  0  0  0  0  0  0  0  0  0  0  0  -  -  0
>RBIT.EA
 AF 0 1<CR>
 B0 0 <ESC>
>RBITA0 AF
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
A0  0  0  0  0  0  0  0  0  0  0  0  0  0  -  -  1
>
```

Example: Fill register bit range 56H-67H with binary values, displaying the affected area both before and after the fill operation.

```
>RBIT 56 67
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
50                     0  0  0  0  0  0  0  0  0  0
60 0  0  0  0  0  0  0  0
>RBIT 56 67 1 0 1 0 1 1 0 1
>RBIT 56 67
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
50                     1  0  1  0  1  1  0  1  1  0
60 1  0  1  1  0  1  1  0
>
```

5.8.2.4 Display Interrupt Enable and Pending Status

RINT

Example: Display interrupt enable and pending status for the 8031. (The interrupt enable/disable register bits are listed in the 1st row with corresponding pending status flags listed in the 2nd row.)

```
>RINT
  EA   ES   EX0   EX1   ET0   ET1
   0    0    0     0     0     0
  RI   TI   IE0   IE1   TF0   TF1
   0    0    0     0     0     0
>
```

Example: Display interrupt enable and pending status for the 8032.

```
>RINT
  EA   ES   EX0   EX1   ET0   ET1   ET2
   0    0    0     0     0     0     0
  RI   TI   IE0   IE1   TF0   TF1   TF2
   0    0    0     0     0     0     0
>
```

Example: Display interrupt enable and pending status for the 8344.

```
>RINT
  EA   ES   EX0   EX1   ET0   ET1
   0    0    0     0     0     0
       SI   IE0   IE1   TF0   TF1
        0    0     0     0     0
>
```

5.8.3 Register Command for NEW MICE-II 80515/80535

```
R[S|register|BIT[.symbol| start-address[ end-address[ data-1[ ...data-8]]]]|INT]
```

NEW MICE-II 80515/80535 supports display/modification of registers and direct-addressable bits, as well as display of interrupt enable and pending status:

R is the Register command. If no other parameters are specified, contents of the ACC, B, DPTR, PSW, SP, current bank registers and the PC are displayed. Note that the PSW displays both hex and binary values.

S displays the contents of all Special Function Registers, except for those listed above.

register are alphabetic characters indicating Special Function Registers, or bank registers, whose content is to be displayed or modified. Register mnemonics and addresses are listed in section 5.8.3.1. (Note that all the registers listed are 8-bit registers.)

BIT indicates a specified operation for direct-addressable bits.

.symbol	is the mnemonic for a bit-addressable register location. Note that all mnemonics must be prefixed by a period ".". Refer to Figure 5-2, Special Function Register Bit-Addressable Locations, Siemens Microcomputer Components, SAB 80515/80535 Single-Chip Microcontroller User's Manual, 7/85.)
start-address	is a hexadecimal pointer for a bit-addressable register in the emulation CPU indicating a memory location where the operation is to begin. The permissible address range is 0~FFH.
end-address	is a hexadecimal pointer for a bit-addressable register in the emulation CPU indicating the last memory location of the specified range. The permissible address range is 0~FFH.
data-1...8	are binary values of "0" or "1". Note that if the address range (from start to end address) is smaller than the block size (data1-data8), then block fill is still executed but excess trailing data is ignored.
INT	displays interrupt enable, pending status and interrupt(s) in progress (from level 0 to 3).

5.8.3.1 Available Registers

The following is a list of accessible registers for the 80515/80535.

Mnemonic	Register	Address
SP	Stack Pointer	81H
DPL	Data Pointer, Low Byte	82H
DPH	Data Pointer, High byte	83H
PCON	Power Control Register	87H
TCON	Timer Control Register	88H
TMOD	Timer Mode Register	89H
TL0	Timer 0, Low Byte	8AH
TL1	Timer 1, Low Byte	8BH
TH0	Timer 0, High Byte	8CH
TH1	Timer 1, High Byte	8DH
SCON	Serial Port Control Register	98H
SBUF	Serial Port Buffer Register	99H
IEN0	Interrupt Enable Register 0	0A8H
IP0	Interrupt Priority Register 0	0A9H
IEN1	Interrupt Enable Register 1	0B8H
IP1	Interrupt Priority Register 1	0B9H
IRCON	Interrupt Request Control Register	0C0H
CCEN	Compare/Capture Enable Register	0C1H
CCL1	Compare/Capture Register 1, Low Byte	0C2H
CCH1	Compare/Capture Register 1, High Byte	0C3H
CCL2	Compare/Capture Register 2, Low Byte	0C4H
CCH2	Compare/Capture Register 2, High Byte	0C5H
CCL3	Compare/Capture Register 3, Low Byte	0C6H
CCH3	Compare/Capture Register 3, High Byte	0C7H
T2CON	Timer 2 Control Register	0C8H
CRCL	Compare/Reload/Capture Register, Low Byte	0CAH
CRCH	Compare/Reload/Capture Register, High Byte	0CBH
TL2	Timer 2, Low Byte	0CCH
TH2	Timer 2, High Byte	0CDH

Mnemonic	Register	Address
PSW	Program Status Word Register	0D0H
ADCON	A/D Converter Control Register	0D8H
ADDAT	A/D Converter Data Register	0D9H
DAPR	D/A Converter Program Register	0DAH
ACC	Accumulator	0E0H
B	B Register	0F0H

5.8.3.2 Display/Modify Registers

R[S|register]

Example: Displays contents of ACC, B, PSW, SP, DPTR, current bank registers and the PC. (The number in parentheses are register addresses.)

```
>R
  ACC  B  PSW  SP DPH DPL  R0 R1 R2 R3 R4 R5 R6 R7    PSW     PC
  (E0)(F0)(D0)(81)(83)(82)                          CAFBBO-P
   00  00  00  07  00  00  00 C4 88 CC C8 E8 0A CA 00000000 0000
>
```

Example: Display contents for the SFR registers not listed in the preceding example.

```
>RS
  PCON  TCON  TMOD  TL0   TL1   TH0   TH1   SCON  SBUF
  (87)  (88)  (89)  (8A)  (8B)  (8C)  (8D)  (98)  (99)
   7F    00    00    00    00    00    00    00    00

  IEN0  IP0   IEN1  IP1   IRCON CCEN  CCL1  CCH1  CCL2  CCH2
  (A8)  (A9)  (B8)  (B9)  (C0)  (C1)  (C2)  (C3)  (C4)  (C5)
   00    80    00    C0    00    00    00    00    00    00

  CCL3  CCH3  T2CON CRCL  CRCH  TL2   TH2   ADCON ADDAT DAPR
  (C6)  (C7)  (C8)  (CA)  (CB)  (CC)  (CD)  (D8)  (D9)  (DA)
   00    00    00    00    00    00    00    00    00    00
>
```

Example: Modify the content of the Accumulator and display the result.

```
>RACC
ACC    (E0) 00 DE<CR>
B      (F0) 00 <ESC>
>R
  ACC  B  PSW  SP DPH DPL  R0 R1 R2 R3 R4 R5 R6 R7     PSW     PC
  (E0)(F0)(D0)(81)(83)(82)                          CAFBBO-P
   DE  00  00  07  00  00  00 C4 88 CC C8 E8 0A CA 00000000 0000
>
```

5.8.3.3 Display/Modify Direct-Addressable Bits

```
RBIT[.symbol|start-address[ end-address[ data-1[ ...data-8]]]]
```

Example: Display all direct-addressable bits for the 80515/80535.

```
>RBIT↵
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00  0  0  0  1  0  0  1  1  0  0  1  1  0  0  1  1
10  0  0  0  0  0  0  1  1  0  0  1  1  0  0  0  1
20  0  0  0  0  0  0  0  1  0  0  1  1  0  1  1  1
30  0  0  0  1  0  0  1  1  0  1  1  1  0  0  1  1
40  0  0  0  1  0  0  0  1  0  0  1  1  0  0  1  1
50  0  0  0  1  0  1  1  1  0  0  1  1  0  1  1  1
60  0  1  0  1  0  0  0  0  0  0  1  1  0  0  0  1
70  0  0  0  1  0  1  1  1  0  0  1  1  0  0  1  1
80  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0
90  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0
A0  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0
B0  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0
C0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
D0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
E0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1
F0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1
>
```

Example: Modify bit register EAL, displaying the affected area both before and after register modification.

```
>RBIT A0 AF
   0 1 2 3 4 5 6 7 8 9 A B C D E F
A0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>RBIT.EAL
 AF 0 1<CR>
 B0 0 <ESC>
>RBITA0 AF
   0 1 2 3 4 5 6 7 8 9 A B C D E F
A0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
>
```

Example: Fill the direct-addressable bit range 56H-67H with binary values, displaying the affected area before and after the fill operation.

```
>RBIT56 67
   0 1 2 3 4 5 6 7 8 9 A B C D E F
50                 0 0 0 0 0 0 0 0 0
60 0 0 0 0 0 0 0 0
>RBIT 56 67 1 0 1 0 1 0 1 1
>RBIT 56 67
   0 1 2 3 4 5 6 7 8 9 A B C D E F
50                 1 0 1 0 1 0 1 1 1 0
60 1 0 1 0 1 1 1 0
>
```

5.8.3.4 Display Interrupt Enable, Interrupts in Progress and Pending Status

---

RINT

Example: Display interrupt enable, pending status, and the interrupt(s) in progress for the 80515/535. (The interrupt enable/disable register bits are listed in the 1st row with corresponding pending status flags listed in the 2nd row.)

```
>RINT
  EAL  ES   ET2  ET1  ET0  EX6  EX5  EX4  EX3  EX2  EX1  EX0  EADC EXEN2
   1    1    1    1    1    0    0    0    0    0    1    1    0    0
  RI   TI   TF2  TF1  TF0  IEX6 IEX5 IEX4 IEX3 IEX2 IE1  IE0  IADC EXF2
   0    0    1    0    0    0    0    0    0    0    0    0    0    0

INTERRUPT LEVEL - 0  1  2  3
IN PROGRESS ITEM - TM0 TM1    TM2
>
```

5.9 Jump/Branch Command - J

J address

J is the command for Jump/Branch.

address is a hexadecimal address of the emulation processor's program memory where the program counter is to be set.

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

Example: Change the program counter from the current address to 5H.

```
>J5
>R
   A   B   C   D   E   H   L   M   P     S   PC
  02  03  00  00  00  00  00  07  00  AAAA 0005
>
```

CHAPTER 6

CONTROL SIGNAL COMMANDS

These commands enable or disable specific control signals going to the emulation processor. The available signal is either enabled or disabled at the input pin on the emulation processor. Only sense input to the processor is affected; circuits which generate the control signal in the target system are unaffected. The control characters and signal designation for hardware control signals (including pin numbers) and software control signals that are affected by the Disable and/or Enable commands are listed below:

Signal Designation	8085	8048	8031/8344	80515/80535	8052/80C152
H - Bus Hold Request	HLDA - 38 HOLD - 39				
I - Interrupt Request	RST 7.5 - 7 RST 6.5 - 8 RST 5.5 - 9 INTR - 10	$\overline{INT}$ - 6			
T - Trap Request and T1 Input	TRAP - 6	T1 - 39			
A - External Memory Access		EA - 7	$\overline{EA}$ - 31 *1	$\overline{EA}$ - 51 *2	$\overline{EA}$ - 31/56 *3
R - Reset			RST - 9 *4	$\overline{RESET}$ - 10 *5	$\overline{RST}$ - 9/13 *4
E - E - Bus Enable					EBEN - 12 *6
C - Specify Program Memory and Data Memory Space (within HUEM/UEM)			*7	*7	*7
M - Specify On-Board Internal Emulation Program Memory Space				*8	

*1 External Memory Access for NEW MICE-II 8031/8344

a) After power-up, MICE checks the $\overline{EA}$ pin and displays the target processor type currently being emulated. The "!" command may be used to verify the current target.

Example: *** MICE-II 8031 V#.# ***

```
>!
 EP8031
>
```

*2 External Memory Access for NEW MICE-II 80515/80535

a) When the "EA" command is input, the status of $\overline{EA}$ depends on the target's $\overline{EA}$ signal.
b) When the "DA" command is input, the status of $\overline{EA}$ is 0.

c) After power-on, MICE defaults to EA mode (i.e. the status of EA depends on the target's EA signal.)
d) MICE-II 80515 firmware 3.1 (CEP Rev: B) can successfully read external program memory in 80535 mode.

*3 External Memory Access for NEW MICE-II 8052/80C152

a) When the "EA" command is input, the status of EA depends on the target's EA signal.
b) When the "DA" command is input, the status of EA is 0.
c) After power-on, MICE defaults to EA mode (i.e. the status of EA depends on the target's EA signal.)

*4 Reset for NEW MICE-II 8031/8344 and 8052/80C152

During Upload, Download and A/I/J/M/O/R/T/Z commands, all interrupts and the RST will be disabled.

*5 Reset for NEW MICE-II 80515/80535

a) When the "ER" command is input, the status of RESET depends on the target's RESET signal.
b) When the "DR" command is input, the status of RESET is 1.
c) After power-on, and during Upload, Download and A/I/J/M/O/R/T/Z commands, all interrupts and the RESET will be disabled.

*6 E-Bus Enable for NEW MICE-II 80C152

a) EBEN is available with 80C152JB only.
b) When the "EE" command is input, the status of EBEN depends on the target's EBEN signal.
c) When the "DE" command is input, the status of EBEN is 0.
d) EA = 1 and EBEN = 0 is invalid combination.

*7 Specify Program Memory and Data Memory Space

This is a software control function; no pin signals are affected. Emulation memory (HUEM/SUEM) is configured by using the following commands.

DX - combined memory space
EX - segregated memory space

a) Key in the "DX" command to combine external program and data memory.
b) Key in the "EX" command to segregate external program and data memory.
c) After power-on, MICE defaults to DX mode (i.e. external program and data memory are combined.) However, New MICE-II 8031/8344 (firmware V3.4 and later), 8052/80C152 (firmware V3.0 and later) and 80515 (firmware V3.1 and later) are exceptional, they default to EX mode (i.e. external program and data memory are segregated).
d) Upload, Download and A/M/T/Z commands are not affected by DX and EX.

*8 Specify CEP On-Board Internal Emulation Program Memory Capacity

Key in the "EM #" command to enable 2, 4, 8 or 16 Kbytes of CEP on-board internal emulation program memory. The system default is for 8K bytes.

Notes for Control Signals:

1. To control emulation of the processor, MICE shares the use of READY input (pin-35) for NEW MICE-II 8085 and SS input (pin 5) for NEW MICE-II 8048, pulling the signal low to suspend emulation.

2. D and E commands cannot control interrupts for 8031/8344, 8052/80C152 and 80515/80535 microprocessors.

6.1 Disable/Display Control Signal Command - D

D[control-signal]

D	is the command for Disable. If no signals are specified, keying in "D<CR>" will display the control signals disabled.
control-signal	is a single alphabetic character indicating a control signal to the emulation processor.

The control signal specified is deactivated at the pin of the emulation processor. When resetting the processor using the X command (section 5.7), these signals are activated if MICE is connected to a power-applied target system.

Example: Deactivate the interrupt request (INT) control signal, thereby not processing any interrupts which may occur. Note that this command does not affect the interrupt mask list in the processor status register.

```
>DI
>
```

Example: Display the disabled control signals.

```
>D
DISABLE - INT
>
```

6.2 Enable/Display Control Signal Command - E

80515/80535 - E[A|R|X|M {2|4|8|16}]
all others - E[control-signal]

E — is the command for Enable. If no signals are specified, keying in "E<CR>" will display the control signals enabled.

control-signal — is a single alphabetic character indicating a control signal to the emulation processor.

The control signal specified is activated at the pin of the emulation processor. When resetting the processor with the X command (section 5.7), these signals are not activated unless MICE is connected to a power-applied target system.

Example: Activate the interrupt request control signal.

```
>EI
>
```

Example: Display the enabled control signals.

```
>E
ENABLE - HOLD
       - INT
       - TRAP
>
```

CHAPTER 7

EMULATION AND TRACE CONTROL COMMANDS

These commands perform free-running emulation or tracing for the emulation processor in real time. Program operation may be recorded with user defined trigger addresses to specify the start or end of tracing. Up to 2048 machine cycles can be recorded.

Status information displayed by these commands use the following column headings: "IFADDR ADDRESS DATA STATUS SPARE(4/8 BITS)"

where: **IFADDR** is the instruction fetch address.
ADDRESS is the hexadecimal value on the address bus.
DATA is the hexadecimal value on the data bus.
STATUS is the type of processor activity.
SPARE is the status of the headers on the personality (CEP) board.

The specific types of processor activity that can be displayed in the status column are indicated below. The single character mnemonics listed for each processor also serve as qualifiers for H/F/B/L and BPP commands, as well as being displayed as statuses for C/SZ commands.

CPU Statuses	8085	8048	8031/8344	80515/80535	8052/80C152
Instruction Fetch Cycle	S	S	S	S	S
Memory Read Cycle	R	R			
program Memory Read Cycle		F	F	F	F
Data Memory Read Cycle			R	R	R
Memory Write Cycle	W	W			
Data Memory Write Cycle		*1	W	W	W
Port Input	I	I*2			
Port Output	O	O			
Interrupt Acknowledge	A		A *3	A	A

*1 Instructions "IN A, P1" and "IN A, P2" will not display I.

*2 Instructions "OUTL P1, A" and "OUTL P2, A" will not display O.

*3 For NEW MICE-II 8031/8344, firmware V3.4 (hardware requirement: CEP8031 Rev: H1) and later, interrupt acknowledge "A" is now able to be specified as a breakpoint qualifier.

7.1 Go/Execution Command - G

G[address]

G is the command for Go/Execution.

address is a hexadecimal address of the emulation processor's program memory where emulation will begin.

The emulation processor's program counter is first set to the address indicated. MICE then starts real-time emulation of the target processor. If no address is specified, program emulation will start at the current program counter.

Example: Display register content (PC=0000H), and start emulation from address 6H.

```
>R
  A  B  C  D  E  H  L  M  P    S    PC
 02 03 00 00 00 00 00 C7 00 7FF0 0000
>G6
>
```

Example: Display register content, and resume emulation from the current PC.

```
>R
  A  B  C  D  E  H  L  M  P    S    PC
 02 03 00 00 00 43 21 C7 00 7FEE 000A
>G
>
```

Note: ?/!/D/E/L/BS* commands may be input without halting emulation, while the processor is executing the Go command, or while executing FR/BR commands after the trigger has been matched. (*BS is a BPP command.)

7.2 Halt/Breakpoint Set Command - H

H[0[1|2]|1 [addx[count[qualifierl]]]|2 [address-2]]

H is the command for Halt/Breakpoint Set.

0 is the command to clear a breakpoint.

1 is breakpoint 1.

2 is breakpoint 2. (Count, qualifier and wildcard do not apply.) addx is an up to 4 digit hexadecimal address setting for breakpoint 1, with a wildcard byte pair of "XX" or "XXXX"; or a wildcard nibble of "X" [only for the third digit for 8048 (e.g. X23)].

Count is a hexadecimal value from 1 to 4000H which specifies the number of times the indicated condition must be matched before emulation stops.

qualifier is a single alphabetic character indicating the type of processor activity to be selected.

address-2 is an up to 4 digit hexadecimal address setting for breakpoint 2. (No wildcard, count or qualifier is permitted.)

7.2.1 Halt

H

Input "H" during emulation or tracing to stop execution immediately and display the current address.

Example: Stop emulation and display the current address.

```
>G
>H
      IFADDR ADDRESS   DATA STATUS SPARE(8 BITS)
                7FEE     24   R       11111111
>
```

7.2.2 Set Breakpoint

H[1[addx[count [qualifier]]]|2[address-2]]

NEW MICE-II supports two real-time breakpoints. Breakpoint 1 can be used to specify a program address where emulation will stop. A count may also be specified to define the number of times the indicated condition is to be matched before emulation stops. If no count is specified, the default is one and emulation stops at the first breakpoint address matched. A qualifier can also be selected to choose the type of processor activity which must be matched along with the breakpoint address and count to stop emulation. If no qualifier is specified, all machine cycles are chosen. To enter a qualifier, a count must first be defined.

Breakpoint 2 can be used to specify a program address where emulation or tracing will stop. When setting breakpoint 2, set the breakpoint address only.

Example: Set breakpoint 1 to address AH and count to 5.

```
>H1 A 5
>
```

Example: Set breakpoint 2 to address 13H.

```
>H2 13
>
```

Breakpoints 1 and 2 only set breakpoint conditions. Unless the emulation processor is in execution, an emulation or trace command must be input to start the CPU. If H1 or H2 is matched, MICE will stop the emulation CPU and display the current address (at the specified breakpoint address) . Breakpoint 1 will reset automatically after the Trace command since it shares the same logic as the Forward/Backward Trace commands. Note that breakpoints are not active in Cycle Step and Instruction Step commands.

Note: For the 8048, 8031/8344, 8052/80C152 and 80515/80535, when the breakpoint is matched, the CPU will finish executing the current instruction cycle and stop at the opcode fetch of the next instruction.

Example: Set an emulation program at 0H; then set breakpoint 1 to AH, count to 1, qualifier for instruction fetch cycle (S) , and breakpoint 2 to 13H.

```
>J0
>H1 A 1 S
>H2 13
>G
PROGRAM  BROKE AT ADDRESS  000A
>
```

Example: Set the breakpoint address to XXEEH, the count to 1 and qualifier for memory write cycle(W).

```
>H1 XXEE 1 W
>G
PROGRAM  BROKE AT  ADDRESS 7FEE
>
```

7.2.3 Display Breakpoint

H[1|2]

Input H1 or H2 to display breakpoint 1 or breakpoint 2 messages respectively.

Example: Display breakpoint 1 and 2 messages. Note that the address, count and qualifier are displayed for breakpoint 1; and only address is displayed for breakpoint 2.

```
>H1
H1= XXEE 1 W
>H2
H2= 13
>
```

7.2.4 Clear Breakpoint

H 0[1|2]

NEW MICE-II can clear all breakpoints concurrently or separately.

Example: Clear breakpoint 1 only.

```
>H01
>H1
H1=
>
```

Example: Clear all breakpoints concurrently.

```
>H0
>H2
H2=
>
```

7.3 Forward Trace command - F

F[R]addX[count[qualifier]]

F is the command for Forward Trace.

R continues running the emulation processor after the trace stops.

addx is an up to 4 digit hexadecimal trigger address where MICE begins recording trace information. An optional wildcard byte pair of "XX" or "XXXX", or a wildcard nibble of "X" [only for the third digit for the 8048 (e.g. X23)], may be specified for this address.

count is a hexadecimal value from 1 to 4000H Which specifies the number of times the target condition must be matched before the trace records information.

qualifier is a single alphabetic character indicating the type of processor activity that must be matched with the trigger address.

Forward tracing starts real-time emulation of the target and begins recording target status information when the trigger condition is matched. Forward trace will stop when the trace buffer is full or breakpoint 2 is reached. Forward trace can record up to 2048 machine cycles.

When the trace stops, MICE halts the emulation CPU at the break address. The recorded information can be examined by using the List Trace Buffer command.

Note: For the 8048, 8031/8344, 8052/80C152 or 80515/80535, when the breakpoint is matched or the trace buffer filled, the CPU will finish executing the current instruction cycle and stop at the opcode fetch of the next instruction.

Option R allows the emulation CPU to continue running after the trace stops but breakpoint 2 will force the CPU to stop. Without option R, the CPU will stop whenever the trace stops. If only the trigger address is specified, all machine cycles after that trigger address encountered are recorded. If a count is specified, the trace will record information after the number of matches of the indicated trigger condition occur. If no count is specified, the trace will begin recording information as soon as the trigger condition is matched.

A single qualifier may be specified to choose the type of processor activity associated with the trigger address; default is for all machine cycles. To enter a qualifier, a count must first be defined. When specified, the trace will start recording only when the trigger address together with the qualifier occurs the number of times defined by count.

Example: Begin recording program status from the trigger address of 3H. In the following message, STEP indicates the last trace frame recorded. The trace count begins at zero, so the actual value is one more than the step number shown. The trace buffer is completely filled if STEP 07FF is displayed. (However, for NEW MICE-II 8048, if the data traced at step 7FEH is a one cycle instruction, emulation will stop at step 7FEH.)

```
>F3
 THE TRACE STOPS AT STEP 07FF
>
```

Depending on command specifications, the elapse time before the buffers fill may be long. Entering <ESC> terminates the trace, yet retains the information already recorded.

Example: Specify a trigger address of 18H which the program never reaches; the trace buffer is emptied. Note that if breakpoint 2 is encountered before the trigger address is reached, the message will be displayed without entering <ESC>.

```
>F18
<ESC>
FORWARD TRACE FAILS
>
```

Example: Begin tracing when memory location 2H is addressed and stop at breakpoint 2 before the trace buffer is filled.

```
>J0
>H2 000A
>F2
 THE TRACE STOPS AT STEP 000A
 PROGRAM BROKE AT ADDRESS 000A
>
```

Example: Begin recording status information the first time the program writes to an address in the range of 7F00H to 7FFFH. Continue running after the trace stops until the program addresses 0014H. Note that qualifiers are only associated with the trigger address; all machine cycles are recorded after the trace has begun.

```
>J0
>H2 14
>FR 7FXX 1 W
THE TRACE STOPS AT STEP 0051
PROGRAM BROKE AT ADDRESS 0014
>
```

7.4 Backward Trace Command - B

B[R]addX[count[qualifier]]

B is the command for Backward Trace.

R continues running the emulation processor after the trace stops.

addx is an up to 4 digit hexadecimal trigger address where emulation is to stop. An optional wildcard byte pair of "XX" or "XXXX", or a wildcard nibble of "X" [only for the third digit for the 8048 (e.g. X23)], may be specified for this address.

Count is a hexadecimal value from 1 to 4000H which specifies the number of times the target condition must be matched before the trace stops recording information.

qualifier is a single alphabetic character indicating the type of processor activity that must be matched with the trigger address.

Backward tracing starts real-time emulation of the target and immediately begins recording target status information until the trigger address or breakpoint 2 is reached. When the trace is matched, MICE halts the emulation CPU at the specified break address.

Note: For the 8048, 8031/8344, 8052/80C152 or 80515/80535, when the breakpoint is matched or the trace buffer filled, the CPU will finish executing the current instruction cycle and stop

at the opcode fetch of the next instruction.

Backward trace can record up to 2048 cycles and the recorded information can be examined using the List Trace Buffer command. If more than 2048 cycles elapse before backward tracing ends, only the last 2048 cycles are recorded. Option R allows the emulation CPU to continue running after the trace stops, however breakpoint 2 will force the CPU to stop and display a breakpoint match message.

Without option R, the CPU will stop whenever the trace stops. If only the trigger-address is specified, all machine cycles are recorded up to and including the trigger-address encountered. If a count is specified, backward tracing will record all information before the number of matches of the indicated trigger-address. If no count is specified, the trace will stop after the first match.

A single qualifier may be specified to choose the type of processor activity associated with the trigger-address. To enter any qualifier, a count must first be defined.

Example: Begin tracing immediately and stop when memory location 14H is addressed. Note that the trace will also stop when breakpoint 2 is encountered.

```
>J0
>H2 000A
>B14
THE TRACE STOPS AT STEP 000C
PROGRAM BROKE AT ADDRESS 000A
>
```

In the above example, the breakpoint message following the command line indicates that 0DH cycles have been recorded. The counter is a hexadecimal value from 0 to 7FFH. The trace count begins at zero, so the actual value is one more than the step number shown.

Again, depending on the specification, elapse time before the trigger address is encountered may be long. Entering an <ESC> terminates the trace, yet retains the information already recorded.

Example: Specify a trigger address of 18H which the program never reaches. The trace buffer is filled with the cycles immediately before <ESC> is received.

```
>B18
<ESC>
THE TRACE STOPS  AT STEP 07FF
>
```

Example: Begin tracing immediately and stop the 1st time the program reads from the address range of 7F00H to 7FFFH. Continue running after the trace stops, until the program addresses 0014H. Note that qualifiers are only associated with the trigger address; all machine cycles are recorded once the trace begins.

```
>J0
>H2 14
>BR 7FXX 1 R
 THE TRACE STOPS AT STEP 0012
 PROGRAM BROKE AT ADDRESS 0014
>
```

7.5 List/Display Trace Buffer Command - L

```
L[step[ address-1[ address-2[ qualifier(s)]]]|S[step]|Z[step]|N]
```

L	is the command for List/Display Trace Buffer.
Step	is a hexadecimal value from 0 to 7FFH indicating the initial cycle step to display.
address-1	is the starting address of the range to be listed.
address-2	is the ending address of the range to be listed.
qualifier(s)	are single characters indicating the type of processor activities to list.
S	lists the trace buffer using mnemonic code.
Z	lists the trace buffer displaying mnemonic code and all machine statuses.
N	displays the frame number where the last trace stopped.

Status information recorded during a trace command is displayed using this command. If a step is specified, NEW MICE-II lists the trace buffer from the specified step to the last recorded step. If no step is specified, NEW MICE-II will list all records in the trace buffer. An optional address range can be entered to specify the range to be listed. If no address range is

specified, then the display range is from 0000H to FFFFH.

The type of information to be listed is specified by entering qualifier(s). If no qualifier is entered, all machine cycles within the specified address range are listed. To enter a qualifier, an address range must first be defined. To enter an address range, a step must first be entered. The List command accepts multiple qualifiers; and displays only machine cycles that match the specified qualifiers. If the buffer is listed with the LS or LZ command, only "step" can be specified.

Example: List all records in the trace buffer. Note that the list operation can be terminated by entering an <ESC>. Note that FRAME is the sequence number of the trace step in hexadecimal; the range is 0 to 2047 (7FFH).

```
>L
   FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)*
   0000    0000    0000   31   S      11111111
   0001            0001   F0   R      11111111
   0002            0002   7F   R      11111111
   0003    0003    0003   21   S      11111111
   0004            0004   34   R      11111111
   0005            0005   12   R      11111111
   0006    0006    0006   E5   S      11111111
   0007            7FEF   12   W      11111111
   0008            7FEF   34   W      11111111
   0009    0007    0007   21   S      11111111
   000A            0008   21   R      11111111<ESC>
>
```

* The trace buffer header for the 8048 is listed as -
  ADDRESS DATA STATUS PORT 1 PORT 2 SPARE(4 BITS)

Example: List the trace buffer from the 6th step to the last step, and terminate the operation before the listing is completed.

```
>L5
 FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
 0005         0005    12   R      11111111
 0006   0006  0006    E5   S      11111111
 0007         7FEF    12   W      11111111
 0008         7FEF    34   W      11111111
 0009   0007  0007    21   S      11111111
 000A         0008    21   R      11111111<ESC>
>
```

Example: List the trace buffer write cycles in the range 5H to 8H starting at step 2.

```
>L2 5 8
   FRAME IFADDR ADDRESS DATA  STATUS SPARE(8 BITS)
   0005         0005    12    R      11111111
   0006   0006  0006    E5    S      11111111
   0009   0007  0007    21    S      11111111
   000A         0008    21    R      11111111
>
```

Example: List the trace buffer from 3H to FH, with qualifiers R and W.

```
>LO 3 F R W
   FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
   0004          0004    34   R      11111111
   0005          0005    12   R      11111111
   000A          0008    21   R      11111111
   000B          0009    43   R      11111111
   000D          000B    06   R      11111111
>
```

Example: List trace message with source code; the operation is terminated before the listing is completed. Note that the command also ends if the trace buffer is empty or if nothing is recorded within the specified range.

```
>LS
   FRAME IFADDR  ADDRESS DATA  SOURCE CODE
   0000   0000    0000    31    LXI   SP,7FF0
   0003   0003    0003    21    LXI   H,1234
   0006   0006    0006    E5    PUSH  H
   0007   0007    0007    21    LXI   H,4321<ESC>
>
```

Example: List trace message with mnemonic code and all machine statuses; then display the frame number of the last trace.

```
>LZ
   FRAME IFADDR ADDRESS DATA  STATUS SPARE(8 BITS) SOURCE CODE
   0000  0000    0000    31    S      11111111     LXI    SP,7FF0
   0001          0001    F0    R      11111111
   0002          0002    7F    R      11111111
   0003  0003    0003    21    S      11111111     LXI    H,1234
   0004          0004    34    R      11111111
   0005          0005    12    R      11111111
   0006  0006    0006    E5    S      11111111     PUSH   H<ESC>
>LN
THE TRACE STOPS  AT STEP 0012
>
```

CHAPTER 8

STEPPED EMULATION COMMANDS

These commands perform cycle-stepped or instruction-stepped emulation of the target processor in real-time. Note that breakpoints are not active for C/S commands. Refer to page 7-1 for a description of the displayed status information.

Note: A blank space (ASCII 20H) along with a handshaking code (ASCII 03H) is sent at the end of each displayed line immediately before the cursor for C/S commands. This code serves as an end of data message to the host computer; and is provided to help in implementing symbolic debuggers. (Refer to page 4-1.)

8.1 Cycle Step Command - C

C[count]

C is the command for Cycle Step.

count specifies the machine cycle interval between displayed messages. The range is 0-FFFFH. Note that an input value of "0" represents 10000H.

The Cycle Step command first stops the emulation processor, steps the program one cycle, and then halts the processor in WAIT state. **MICE displays the target status for the current cycle to be executed,** and awaits a <CR> or <LF> to advance the emulation processor to the next machine cycle. The new status is then displayed and MICE again waits. Enter <CR> or <LF> to repeat the entire sequence, or an <ESC> to terminate the single-cycle mode of emulation. Because the displayed status is still active, debug for associated hardware signals is simplified.

However, the 8048 stops the emulation processor at HALT state instead, **where the displayed cycle status has already been executed;** while the 8031/8344, 8052/80C152 and 80515/80535 keep the CPU running in place with a 'SJMP $' instruction, **where the displayed instruction (including all associated machine cycles) has already been executed.**

Note: During Cycle Step for the 8048, the whole instruction (1 or 2 cycles) is executed each time. For certain instruction types (e.g. RET, RETI), the status for the second cycle cannot be identified by NEW MICE-II. Activity for these instructions

will be displayed as "S" (1st cycle) and "F" (2nd cycle)

Example: Cycle step the following program.

```
>J0
>C
    IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
     0000    0000    31   S      11111111<CR>
             0001    F0   R      11111111<CR>
             0002    7F   R      11111111<CR>
     0003    0003    21   S      11111111<CR>
             0004    34   R      11111111<CR>
             0005    12   R      11111111<CR>
     0006    0006    E5   S      11111111<CR>
             7FEF    12   W      11111111<ESC>
>
```

Example: Cycle step the above program using a count of 3.

```
>J0
>C3
    IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
             0002    7F   R      11111111<CR>
             0005    12   R      11111111<CR>
             7FEE    34   W      11111111<CR>
             0009    43   R      11111111<CR>
     000C    000C    71   S      11111111<CR>
     000E    000E    E3   S      11111111<ESC>
>
```

8.2 Instruction Step Command - S

S[[S|R][count]|Z]

S — is the command for Instruction Step.

S(option) — displays messages in mnemonic form.

R — displays register contents in addition to mnemonic code.

count — is an hexadecimal value from 0~FFFFH specifying the step interval between status display. (Note that 0000H indicates 10000H steps.)

Z — displays cycle status in addition to mnemonic code.

8.2.1 Instruction Step

S

If count is not specified for instruction step, the default is one. In this mode, **MICE first displays the current instruction to be executed** and waits for input. (**However, remember that for the 8031/8344, 8052/80C152 and 80515/80535 the instruction displayed has already been executed.**) To cause the emulation processor to execute this instruction, a <CR> or <LF> should be entered; the new status is then displayed and MICE again waits. Enter a <CR> or <LF> to repeat the entire sequence, displaying the new status after each instruction. Enter an <ESC> to terminate the single step mode

of emulation. Note that breakpoints are not active for the S command.

Instruction step only displays "S" status 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 problem area, and then Cycle Step through the problem area for a more detailed display.

Example: Instruction step the previous program. Note that the processor does not begin emulation until the current status displays and either a <CR> or <LF> is entered.

```
>J0
>S
    IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
     0000   0000     31   S      11111111<CR>
     0003   0003     21   S      11111111<CR>
     0006   0006     E5   S      11111111<CR>
     0007   0007     21   S      11111111<ESC>
>
```

8.2.2 Instruction Step in Mnemonic Form

SS

Example: Instruction step with S option.

```
>J0
>SS
    IFADDR ADDRESS DATA     SOURCE CODE
     0000   0000     31     LXI   SP,7FF0<CR>
     0003   0003     21     LXI   H,1234<CR>
     0006   0006     E5     PUSH  H<ESC>
>
```

8.2.3 Instruction Step with Register Content

SR

The R option displays register content along with mnemonic information. All register contents associated with the current step in program execution can be observed; but remember that the contents displayed are prior to executing the instruction.

Example: Instruction step with R option.

```
>J0
>SR
     IFADDR ADDRESS DATA     SOURCE CODE
     A=00 B=AA C=06 D=AA E=AA H=43 L=21 M=07 P=54 S=7FEE PC=0000
     0000     0000     31      LXI    SP,7FF0<CR>
     A=00 B=AA C=06 D=AA E=AA H=43 L=21 M=07 P=54 S=7FF0 PC=0003
     0003     0003     21      LXI    H,1234<CR>
     A=00 B=AA C=06 D=AA E=AA H=12 L=34 M=07 P=54 S=7FF0 PC=0006
     0006     0006     E5      PUSH   H<ESC>
>
```

8.2.4 Instruction Step with Count

S[S|R] count

If an instruction count of one (default) is entered, the current instruction to be executed is displayed; NEW MICE-II does not step. If an instruction count other than one is entered, emulation immediately begins from the current program counter and continues until the count of the next instruction to be executed equals the number specified. The current status is then displayed and NEW MICE-II waits for a <CR> or <LF> before executing the next "count" instruction. Enter an <ESC> to terminate the multi-step mode of emulation. For large intervals, emulation may be terminated by entering an <ESC> before the specified number of instructions are executed.

Instruction Step executes "count-1" instructions before the first status line is displayed and then executes "count" instructions between subsequent displays.

Example: Multi-step the program and compare results with the display in the first example.

```
>J0
>S3
    IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
     0006    0006   E5   S      11111111<CR>
     000C    000C   71   S      11111111<CR>
     000F    000F   0D   S      11111111<CR>
     000D    000D   23   S      11111111<ESC>
>
```

8.2.5 Instruction Step with Cycle Status

SZ

The SZ option includes a memory and I/O access message that provides data for bus cycles R/W/I/O/A [e.g. 7FEF-W-12]. The first four digits (7FEF) indicate the memory address; the capital letter (W) represents the type of bus cycle; and the last two digits (12) are the data transferred.

Example: Instruction step the program displaying Cycle status together with mnemonic code.

```
>JO
>SZ
      IFADDR ADDRESS DATA     SOURCE CODE
       0000    0000    31     LXI    SP,7FF0<CR>
       0003    0003    21     LXI    H,1234<CR>
       0006    0006    E5     PUSH   H<CR>
              7FEF-W-12
              7FEE-W-34
       0007    0007    21     LXI    H,4321<ESC>
>
```

8.3 Application Notes For Stepped Emulation Commands

To make debugging easier, there are two available methods for slowing down the CPU's execution speed.

1. Use the C command with a "Ø" count to slow execution speed as follows:

NEW MICE-II	Speed
8085	1/120
8048	1/70
8031/8344	1/98
8052/80C152	1/98
80515/80535	1/108

2. Use the S command with a "Ø" count to slow execution speed as follows:

NEW MICE-II	Speed
8085	1/8330
8048	1/70
8031/8344	1/98
8052/80C152	1/98
80515/80535	1/108

3. Typing ahead with <CR> will keep the processor running.

   Example: >CØ
   <CR>
   <CR>
   <CR>

CHAPTER 9

UTILITY COMMANDS INVOLVING A SYSTEM

These commands are only applicable when NEW MICE-II is connected to a host system. Programs and data can be downloaded from a file in a host computer to memory in the target system. Conversely, information can also be uploaded from the target back to the host.

Both Intel and Tektronix loading formats are recognized by NEW MICE-II. Each of these formats are described in detail in the following pages.

Note: Only program memory (i.e. memory type "P") can be accessed when using Upload or Download commands for the 8031/8344 or 80515/80535. NEW MICE-II 8052/80C152 can upload or downnload from program memory and data memory. (Refer to section 2.6 for a detailed description of memory types).

9.1 Download Command (Intel Format) - :

:load-record

: is the command keyword for Download.

load-record is a string of ASCII data containing up to 128 bytes (256 bytes for 8048 firmware V3.3, 8031 firmware V3.4, 8052 firmware V3.0, 80515 firmware V3.1 and all later) of program information.

Each record transferred contains the record type, length, memory load address, and checksum in addition to data. Each transfer is limited to 128 bytes (256 bytes for 8048 firmware V3.3, 8031 firmware V3.4, 8052 firmware V3.0, 80515 firmware V3.1 and all later) of program data. The general format of a record, shown with spaces separating each field, is:

RECORD MARK	RECORD LENGTH	LOAD ADDRESS	RECORD TYPE	PROGRAM DATA	CHECK-SUM
:	##	aaaa	tt	dd...dd	cc

where:

: is the keyword used to signal start of record.

is a two ASCII hexadecimal value indicating the record length, the number of data bytes in the 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. (For record type 01 [next item], this field contains "0000".) Successive data bytes are stored in the following memory locations.

tt is a two ASCII hexadecimal value representing the record type:

tt	##
00 - data record	actual data length
01 - end of file record	00

If record type 2 or 3 is chosen by the user, it will be ignored during downloading.

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

tt	dd...dd
00	A pair of hex digits representing the ASCII code for each data byte, where the high order digit is the 1st digit of each pair.
01	none

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, take two at a time, modulo 256 should be zero.

When NEW MICE-II receives a ":", it reads the record length (##). The rest of the record is then read and verified. If the incoming checksum agrees with the computed checksum, NEW MICE-II stores the data into program memory of the target system and reprompts after the following acknowledgement response <ACK sequence> is sent:

ACK LF CR NUL NUL NUL NUL NUL NUL

If the checksum does not agree, NEW MICE-II also reprompts but the alternate negative acknowledgement response <NAK sequence> is sent:

NAK LF CR NUL NUL NUL NUL NUL NUL

Example: Download with an end-record into a target system using Intel format. Note that the characters are not echoed. A <CR> is not required at the end of the input line.

```
>:060000002300A8A917046B<ACK sequence>
>:00000001FF<ACK sequence>
>
```

In the above example, the first load record is

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

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

For the end-record,

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

where: 00H+00H+00H+01H+FFH = 00H

9.2 Download Command (Tektronix Format) - /

/load-record

/ is the command keyword for Download.

load-record is a string of ASCII data containing up to 128 bytes (255 bytes for 8048 firmware V3.3, 8031 firmware V3.4, 8052 firmware V3.0, 80515 firmware V3.1 and all later) of program information.

Each record transferred contains the record type, length, memory load address and checksum in addition to data. Each transfer is limited to 128 bytes (255 bytes for 8048 firmware V3.3, 8031 firmware V3.4, 8052 firmware V3.0, 80515 firmware V3.1 and all later) of program data. The general format of a record, shown with spaces separating each field, is:

RECORD MARK	LOAD ADDRESS	RECORD LENGTH	LOAD SUM	PROGRAM DATA	CHECK-SUM
/	aaaa	##	ss	dd...dd	cc

where:

/ is the keyword used to signal 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. Successive data bytes are stored in the following memory locations.

is a two ASCII hexadecimal value indicating record length, the number of data bytes in the record. A record length of zero indicates 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 NEW MICE-II receives a "/", input is read until a <CR> terminates the line. Checksums are then computed and compared. If the incoming checksum agrees with the computed checksum, NEW MICE-II stores the data into program memory of the target system and reprompts after the following acknowledgement response <ACK sequence> is sent:

ACK LF CR NUL NUL NUL NUL NUL NUL

If the checksum does not agree, NEW MICE-II also reprompts but the alternate negative acknowledgement response <NAK sequence> is sent:

NAK LF CR NUL NUL NUL NUL NUL NUL

Example: Download with an end-record into a target system using Tektronix format.

```
>/000006062300A8A9170436<CR>
<ACK sequence>
>/00000000<CR>
<ACK sequence>
>
```

In the above example, the load record is

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

where: ss = 0H+0H+0H+0H+0H+6H = 06H
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

9.3 Upload Command - U

U start-address end-address [format]

U	is the command for Upload.
start-address	is a logical hexadecimal address of the emulation processor's memory where data transfer begins.
end-address	is the last logical hexadecimal address of the emulation processor's program memory where data transfer ends.
format	is a single alphabetic character [I|T] indicating the load-record format to be used. I - Intel T - Tektronix

Memory contents defined by the memory range start-address through end-address are transferred to the host system using either Intel or Tektronix format. If the format is not specified, then Intel is used. Also, the end-address must be greater than or equal to the start-address or an error message is sent and the command ended.

NEW MICE-II first sends the current segment value record to the host system. It then reads data from memory and sends a maximum of 32 bytes (depending on data length) to the host system using one of the above formats. Data transmission continues in this manner until an end address is sent. NEW MICE-II then sends an end-of-record file and prompts for the next command. For 8048 firmware V3.3, 8031 firmware V3.4, 8052 firmware V3.0, 80515 firmware V3.1 and all later, handshaking code will be sent out after each record during uploading.

If anything other than an <ACK> is received when NEW MICE-II is waiting for an acknowledgement response, the same block is retransmitted. If after five retries transmission is still unsuccessful, the command is aborted, and an error message is sent to the host system before NEW MICE-II prompts for the next command. The command can also be aborted by the host system when NEW MICE-II receives an <ESC> character from the console.

Example: Upload from NEW MICE-II using Intel format. Note that load records are shown on separate lines for clarity. No <CR> or <LF> characters are generated by NEW MICE-II, and anything received other than an <ACK>, is treated the same as a <NAK>.

```
>U 0 6 I<CR>
:070000002300A8917046E8FF<ACK>
:00000001FF<ACK>
>
```

Example: Upload using Tektronix format.

```
>U 0 6 T<CR>
/000007072300A8917046E848<CR>
<NAK>
/000007072300A8917046E848<CR>
<ACK>
/00000000
<ACK>
>
```

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

Appendix A

Summary of MICE-II Commands

MICE-II UTILITY COMMANDS

? - Help Command
! - Attention Command

MEMORY, PORT AND REGISTER COMMANDS

M - Memory Display/Examine/Modify/Fill/Search Command
M[start-address[end-address[data-1[...data-8]
[S]]]]

T - Memory Checksum/Test/Transfer/Compare Command
T start-address end-address {S|M|address-3[V]}

A - Line Assembly command
A[start-address]

Z - Disassembly Command
Z[start-address[,end-address]]

I - Port Input Command
I port[count[duration]]

O - Port Output Command
O port data-1 [...data-8]

X - Reset/Initialization Command
X[address]

R - Register Display/Modify Command
R[register]

J - Jump/Branch Command
J address

CONTROL SIGNAL COMMANDS

D - Disable/Display Control Signal Command
D[control signal]

E - Enable/Display Control Signal Command
E[control signal]

EMULATION AND TRACE CONTROL COMMANDS

G - Go/Execution Command
G[address]

H - Halt/Breakpoint Set Command
H[0[1|2]|1[address-1[count[qualifier]]]
|2 [address-2]]

F - Forward Trace Command
F[R]trigger-address[count[qualifier]]

B - Backward Trace Command
B[R]trigger-address[count[qualifier]]

L - List/Display Trace Buffer Command
L[step[address-1[address-2[qualifier(s)]]]
|S[step]|Z[step]|N]

STEPPED EMULATION COMMANDS

C - Single Cycle/Step Command
C[count]

S - Instruction Step Command
S[[S|R][count]|Z]

UTILITY COMMANDS INVOLVING A SYSTEM

: - Download Command (Intel Format)
:load-record

/ - Download Command (Tektronix Format)
/load-record

U - Upload Command
Ustart-address end-address [I|T]

Appendix B

Hexadecimal-Decimal Conversion

To find the decimal equivalent of a hexadecimal number, first locate your hex number in the correct position (1st through 4th place digit) in the following table. Note the decimal equivalent for each hex digit in your number and then add up these decimal values.

To find the hexadecimal equivalent of a decimal number, locate the next lower decimal number in the table, write down the hex equivalent and its position (1st through 4th place digit). Subtract this decimal number from yours, take the difference and continue this process until conversion is completed.

BYTE				BYTE			
4th place digit		3rd place digit		2nd place digit		1st place digit	
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	61444	F	3840	F	240	F	15

Appendix C

ASCII Code Lists and Definitions

Table C-1
ASCII Code List

HEX	DEC	CHAR	HEX	DEC	CHAR
00	0	NUL	22	34	"
01	1	SOH	23	35	#
02	2	STX	24	36	$
03	3	ETX	25	37	%
04	4	EOT	26	38	&
05	5	ENQ	27	39	'
06	6	ACK	28	40	(
07	7	BEL	29	41	)
08	8	BS	2A	42	*
09	9	HT	2B	43	+
0A	10	LF	2C	44	,
0B	11	VT	2D	45	-
0C	12	FF	2E	46	.
0D	13	CR	2F	47	/
0E	14	SO	30	48	0
0F	15	SI	31	49	1
10	16	DLE	32	50	2
11	17	DC1	33	51	3
12	18	DC2	34	52	4
13	19	DC3	35	53	5
14	20	DC4	36	54	6
15	21	NAK	37	55	7
16	22	SYN	38	56	8
17	23	ETB	39	57	9

Appendix C

ASCII Code Lists and Definitions

Table C-1
ASCII Code List (Cont.)

HEX	DEC	CHAR	HEX	DEC	CHAR
18	24	CAN	3A	58	:
19	25	EM	3B	59	;
1A	26	SUB	3C	60	<
1B	27	ESC	3D	61	=
1C	28	FS	3E	62	>
1D	29	GS	3F	63	?
1E	30	RS	40	64	@
1F	31	US	41	65	A
20	32	SP	42	66	B
21	33	!	43	67	C
44	68	D	62	98	b
45	69	E	63	99	c
46	70	F	64	100	d
47	71	G	65	101	e
48	72	H	66	102	f
49	73	I	67	103	g
4A	74	J	68	104	h
4B	75	K	69	105	i
4C	76	L	6A	106	j
4D	77	M	6B	107	k
4E	78	N	6C	108	l
4F	79	O	6D	109	m
50	80	P	6E	110	n

Appendix C

ASCII Code Lists and Definitions

Table C-1
ASCII Code List (Cont.)

HEX	DEC	CHAR	HEX	DEC	CHAR
51	81	Q	6F	111	o
52	82	R	70	112	p
53	83	S	71	113	q
54	84	T	72	114	r
55	85	U	73	115	s
56	86	V	74	116	t
57	87	W	75	117	u
58	88	X	76	118	v
59	89	Y	77	119	w
5A	90	Z	78	120	x
5B	91	[	79	121	y
5C	92	\	7A	122	z
5D	93	]	7B	123	{
5E	94	^	7C	124	|
5F	95	-	7D	125	}
60	96	`	7E	126	~
61	97	a	7F	127	DEL

Appendix C

ASCII Code Lists and Definitions

Table C-2
ASCII-Code Definitions

ABB	CTRL	MEANING	HEX
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 Acknowledgement	15
SYN	V	Synchronous Idle	16

Table C-2
ASCII-Code Definitions (Cont.)

ABB	CTRL	MEANING	HEX
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 NEW MICE-II Driver Program

If for some reasons user wishes to write his own NEW MICE-II driver program, the following guides are provided and should be considered in writing such program.

D.1 System Setup

User's system must have an RS-232C series communication port.

Communication protocol refer to section 2.5.

D.2 MICE Command Transparency

STEP	D R I V E R A C T I O N	R E M A R K S
1	Sends 'CR' to try to initiate communication with MICE. If it receives no response, send the character which has just been input from keyboard. Receive all responses from MICE and display them until a prompt (>) followed by handshaking code appear on the screen.	;MICE responds with self-diagnostic results.
2	Pick up one key (keyboard input) say "char-X", and send it to MICE. Receive response from MICE and display it until MICE echo with message- "char-Y". where: If "char-X" is BS, receive two data If the second data is SP, "char-Y" is BS. else "char-Y" is handshaking code. If "char-X" is ESC, "char-Y" is handshaking code, else "char-Y" is the same as "char-X".	 ;BS (Back Space=08H) ;SP (Space bar=20H) ;ESC=1BH
3	If "char-X" is 'CR', Receive response from MICE and display received messages until a prompt (>) followed by handshaking code appear on the screen. At the same time, scan keyboard input during display and send any received data to MICE.	;CR=0DH
4	Goto Step 2.	

D.3 Upload

STEP	DRIVER ACTION	REMARKS
1	Send attention command to MICE. If displayed title includes version description, set flag.	;Command syntax: "! CR"
2	Send upload command to MICE. Receive echo messages until 'CR' is detected.	;Command syntax: "U start_adr end_adr CR"
3	If flag on, receive upload data until handshaking code is detected. else receive upload data counted by record length.	
4	Store record to file.	
5	Send ACK (CTRL-F) to MICE.	
6	If not end-record, goto Step 3. Otherwise close the file.	

D.4 Download

STEP	DRIVER ACTION	REMARKS
1	Get one record from file.	
2	Send one record to MICE and receive one character (e.g. char-Z) from MICE.	
3	If "char-Z" is ACK (Ctrl-F), receive until handshaking code is detected. If it is the end-record, close the file, else goto Step 2. If "char-Z" is NACK (Ctrl-U), receive until handshaking code is detected. If the same record is sent 5 times, abort download. else goto step 3. Else display "char-Z", receive and display until handshake code appears on the screen. Abort download.	;For next record. ;Retry. ;Error message from MICE.

Appendix E

NEW MICE Driver Programs

USD (Universal Symbolic Debugger) available on various host computer and its medium:

	Host Computer	Standard Medium
USD/PC	IBM/PC/XT/AT	5-1/4" DSDD floppy diskette
USD PC	IBM PS/2	3-1/2" 2S/HD floppy diskette
USD 9801	NEC PC-9801 family	5-1/4" DSDD floppy diskette
USD VMS	VAX	TU-80 Magtape, Files-11 format
USD/VMS	Micro VAX	TK-50 Cartridge tape, Files-11 format
USD/ULTRIX	Micro VAX	TK-50 Cartridge tape, Tar format
USD SUN	SUN microsystems	1/4" Cartridge tape, Tar format

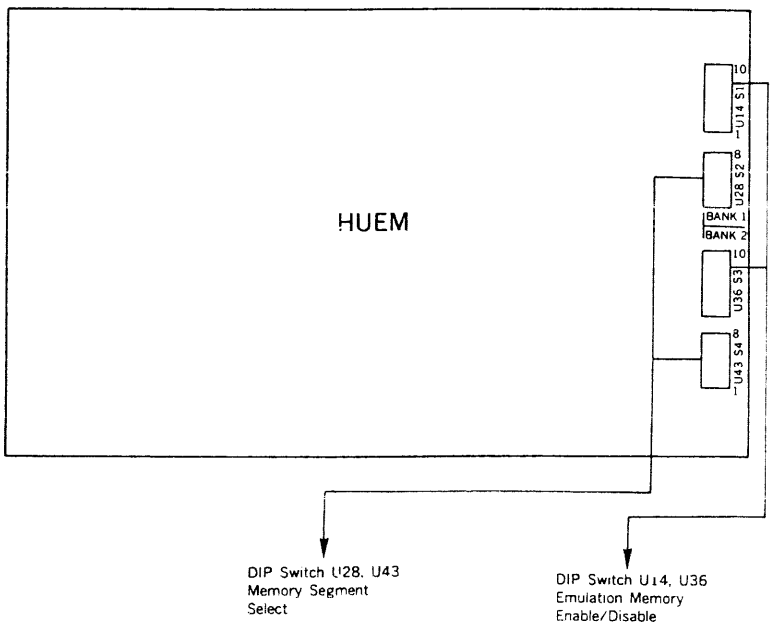
Appendix F

High Performance Universal Emulation Memory Board (HUEM)

The High Performance Universal Emulation Memory (HUEM) board is the bottom board in the conventional three board NEW MICE-II module. Each memory board supports 128K bytes of emulation memory.

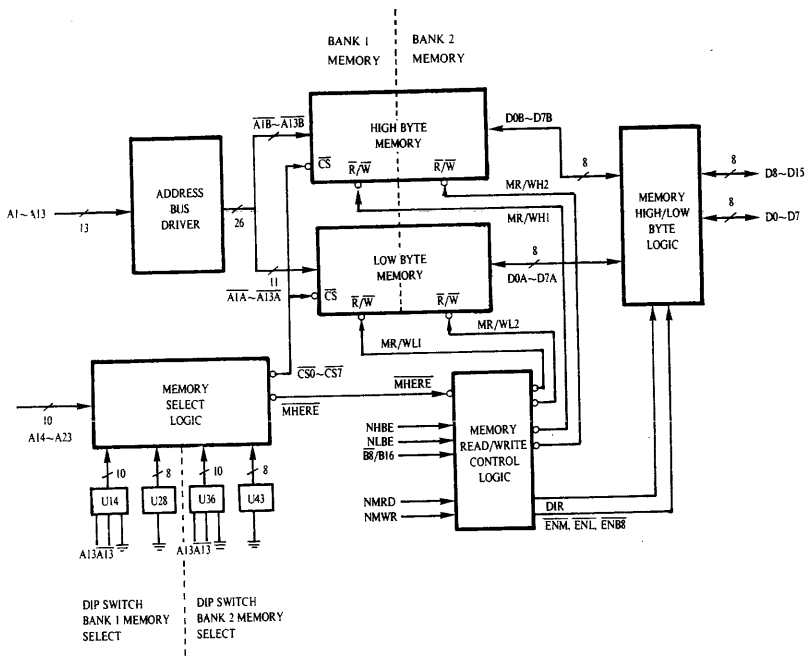
Appendix F

High Performance Universal Emulation Memory Board (HUEM) Placement Chart



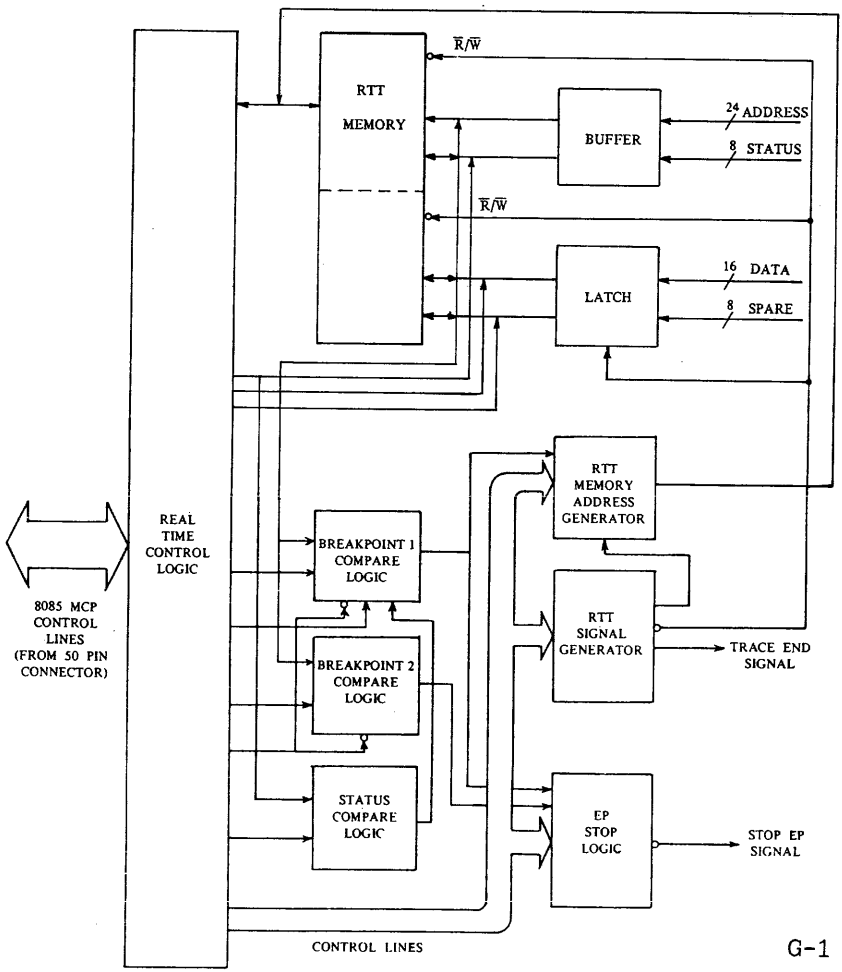
Appendix F

High Performance Universal Emulation Memory Board (HUEM) Block Diagram



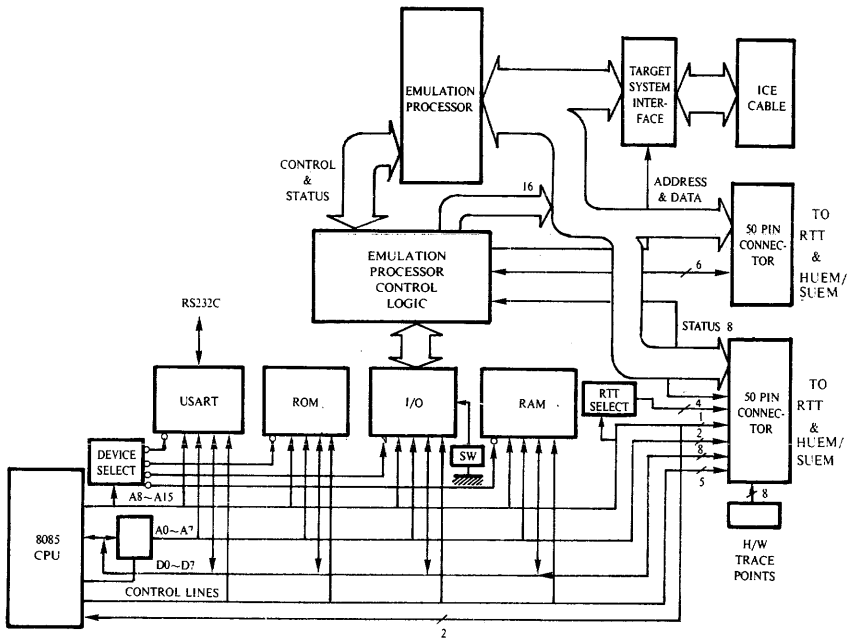
Appendix G

Realtime Trace Board (RTT)
Block Diagram



Appendix H

Control Emulation Processor Board (CEP) Block Diagram

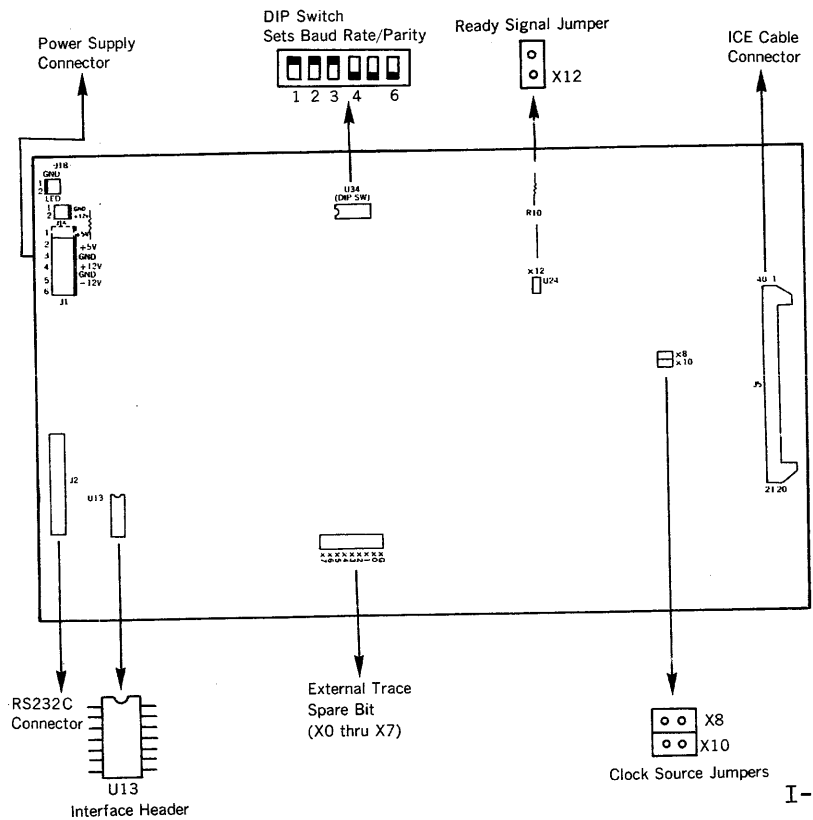


Appendix I

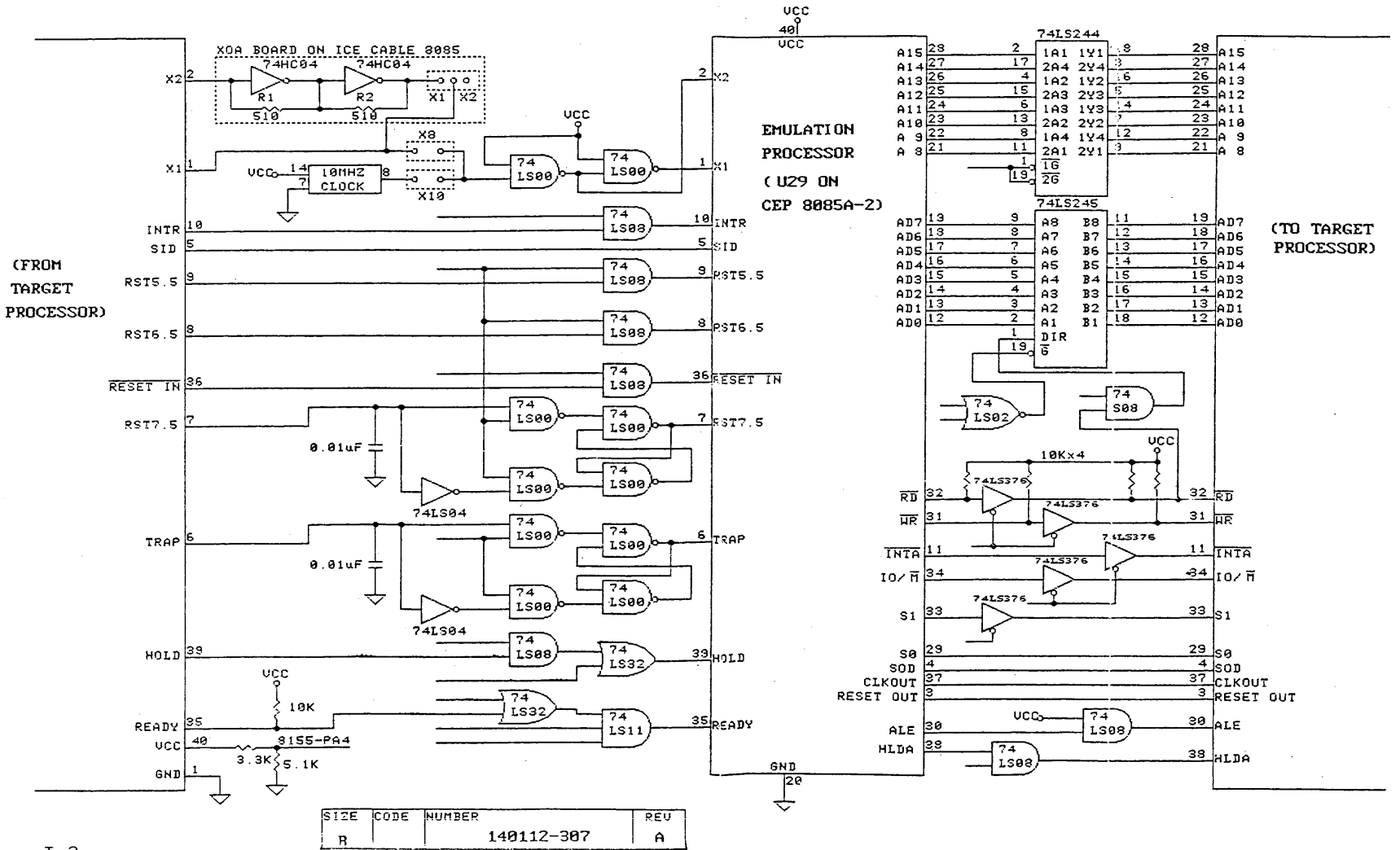
8085

I.1 Hardware

I.1.1 Control Emulation Processor Board, CEP-8085



I.1.2 Interface Diagram from Target Processor to CEP-8085



SIZE	CODE	NUMBER	REV
B		140112-307	A

I.2 Application Note for the 8085 Command Set

During I/O read and write 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 command specification must also duplicate the port address in the upper byte.

I.3 Assembly Mnemonic Code Summary, with Disassembly Examples

LOC	OBJ	LINE	LABEL	SOURCE CODE	
0000	CE00	0001	S0000	ACI	00
0002	8F	0002		ADC	A
0003	8E	0003		ADC	M
0004	87	0004		ADD	A
0005	C622	0005		ADI	22
0007	A0	0006		ANA	B
0008	E633	0007		ANI	33
000A	CD0000	0008		CALL	0000
000D	DC2222	0009		CC	2222
0010	FC2022	0010		CM	2220
0013	2F	0011		CMA	
0014	3F	0012		CMC	
0015	BA	0013		CMP	D
0016	BF	0014		CMP	A
0017	D40022	0015		CNC	2200
001A	C40011	0016		CNZ	1100
001D	F43333	0017		CP	3333
0020	EC3244	0018		CPE	4432
0023	FE33	0019		CPI	33
0025	E43344	0020		CPO	4433
0028	CC2211	0021		CZ	1122
002B	27	0022		DAA	

LOC	OBJ	LINE	LABEL	SOURCE CODE	
002C	09	0023		DAD	B
002D	0D	0024		DCR	C
002E	25	0025		DCR	H
002F	0B	0026		DCX	B
0030	F3	0027		DI	
0031	FB	0028		EI	
0032	76	0029		HLT	
0033	DB00	0030		IN	00
0035	3C	0031		INR	A
0036	0C	0032		INR	C
0037	13	0033		INX	D
0038	DA0000	0034		JC	0000
003B	FAFFFF	0035		JM	FFFF
003E	C34444	0036		JMP	4444
0041	D23333	0037		JNC	3333
0044	C24444	0038		JNZ	4444
0047	F24444	0039		JP	4444
004A	EA3333	0040		JPE	3333
004D	E23333	0041		JPO	3333
0050	CA0000	0042		JZ	0000
0053	3A9798	0043		LDA	9897
0056	1A	0044		LDAX	D
0057	2A9090	0045		LHLD	9090
005A	113300	0046		LXI	D,0033
005D	54	0047		MOV	D,H
005E	78	0048		MOV	A,B
005F	6A	0049		MOV	L,D
0060	0E33	0050		MVI	C,33
0062	0E44	0051		MVI	C,44
0064	00	0052		NOP	
0065	B1	0053		ORA	C
0066	B2	0054		ORA	D
0067	F633	0055		ORI	33
0069	D333	0056		OUT	33

LOC	OBJ	LINE	LABEL	SOURCE CODE	
006B	E9	0057		PCHL	
006C	D1	0058		POP	D
006D	E5	0059		PUSH	H
006E	17	0060		RAL	
006F	1F	0061		RAR	
0070	D8	0062		RC	
0071	C9	0063		RET	
0072	20	0064		RIM	
0073	07	0065		RLC	
0074	F8	0066		RM	
0075	D0	0067		RNC	
0076	F0	0068		RP	
0077	E8	0069		RPE	
0078	E0	0070		RPO	
0079	0F	0071		RRC	
007A	C7	0072		RST	0
007B	C8	0073		RZ	
007C	9A	0074		SBB	D
007D	9E	0075		SBB	M
007F	DE33	0076		SBI	33
0081	224400	0077		SHLD	0044
0083	30	0078		SIM	
0084	F9	0079		SPHL	
0085	324444	0080		STA	4444
0088	12	0081		STAX	D
0089	37	0082		STC	
008A	92	0083		SUB	D
008B	D644	0084		SUI	44
008D	EB	0085		XCHG	
008E	AA	0086		XRA	D
008F	AE	0087		XRA	M
0090	EE44	0088		XRI	44
0092	E3	0089		XTHL	

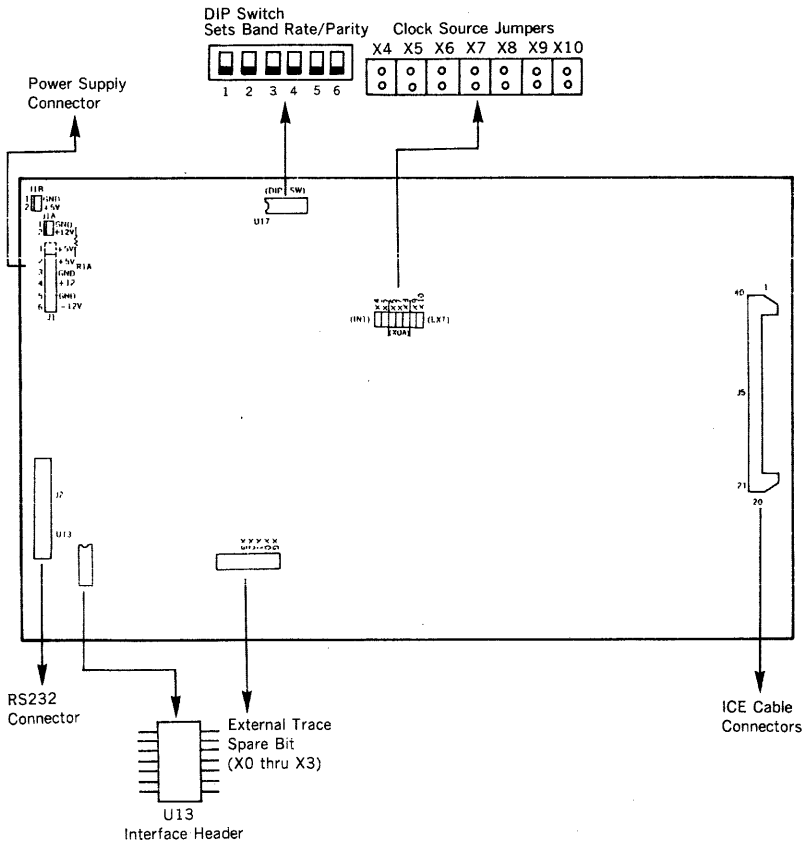
Appendix J

8048

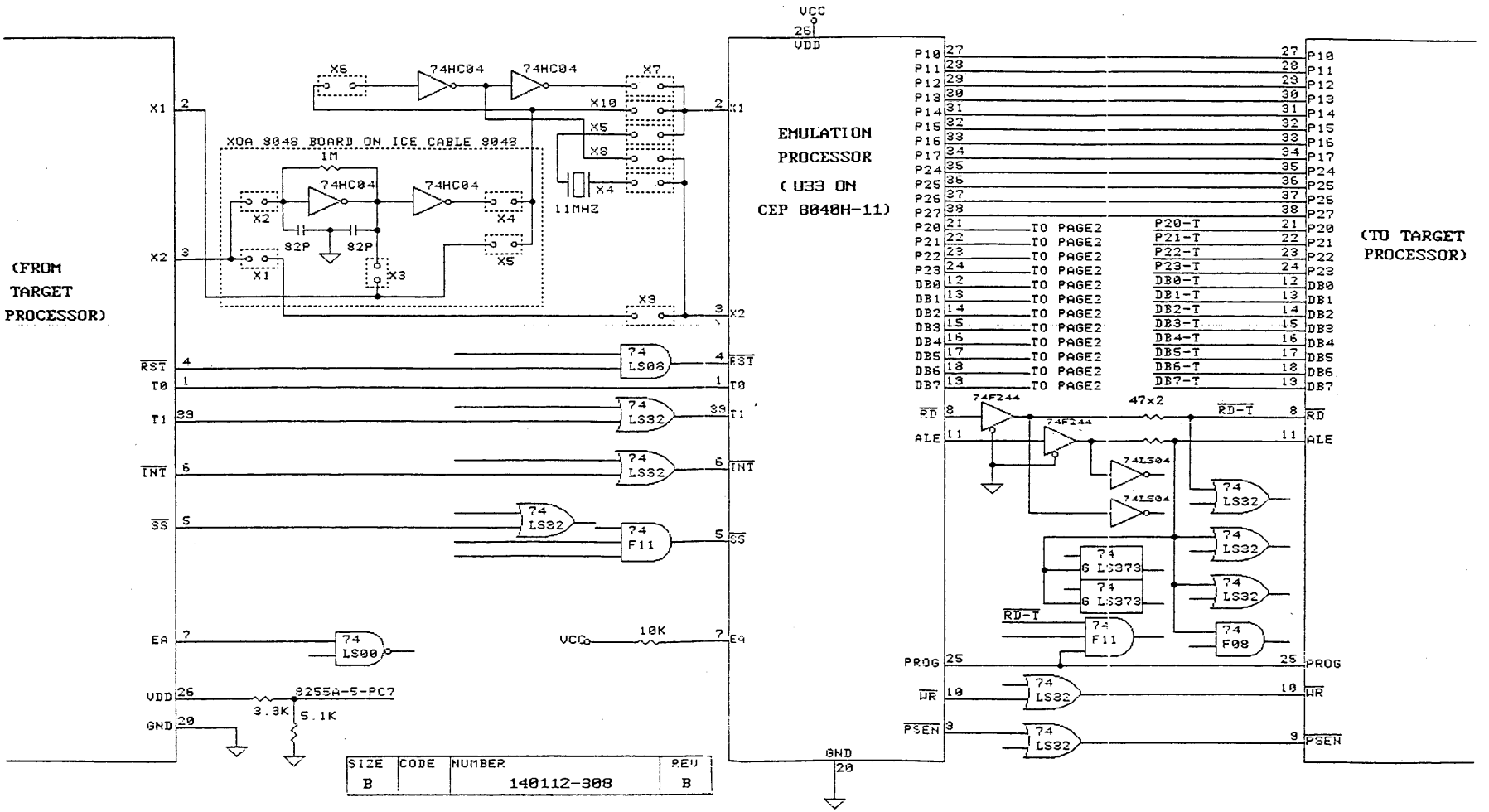
J.1 Hardware

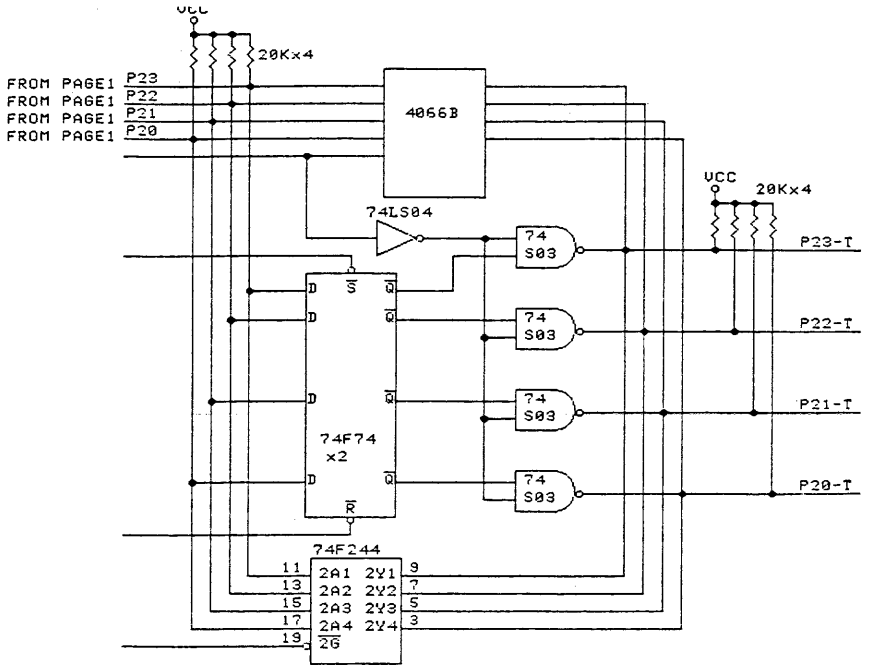
NEW MICE-II 8048 includes two boards, CEP-8048 and RTT. When replacing the CEP board on another MICE model with the CEP-8048, the memory board (HUEM/SUEM) may be removed if desired.

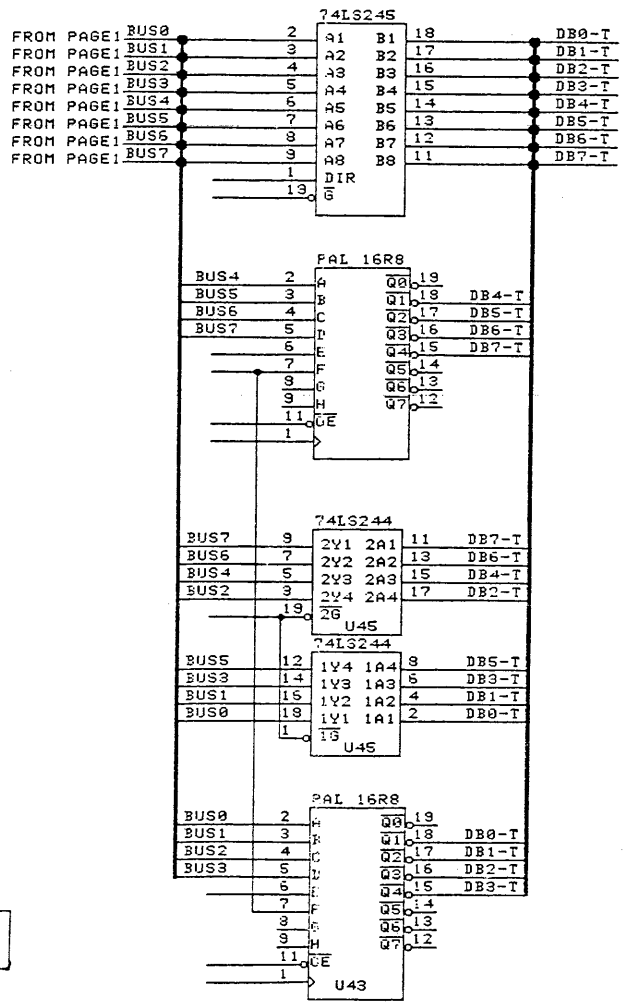
J.1.1 Control Emulation Processor Board, CEP-8048



J.1.2 Interface Diagram from Target Processor to CEP-8048







SIZE	CODE	NUMBER	REV
B		140112-308	B

J.2 Assembly Mnemonic Code Summary, with Disassembly Examples

LOC	OBJ	LINE	LABEL	SOURCE CODE	
000	6B	001		ADD	A,R3
001	61	002		ADD	A,@R1
002	0322	003		ADD	A,#22
004	7C	004	B004	ADDC	A,R4
005	71	005		ADDC	A,@R1
006	1333	006	B006	ADDC	A,#33
008	5B	007		ANL	A,R3
009	51	008		ANL	A,@R1
00A	5311	009		ANL	A,#11
00C	9822	010		ANL	BUS,#22
00E	9922	011		ANL	P1,#22
010	9C	012		ANLD	P4,A
011	1433	013		CALL	033
013	27	014		CLR	A
014	97	015		CLR	C
015	A5	016		CLR	F1
016	85	017		CLR	F0
017	37	018		CPL	A
018	A7	019		CPL	C
019	95	020		CPL	F0
01A	B5	021		CPL	F1
01B	57	022		DA	A
01C	07	023		DEC	A
01D	CC	024		DEC	R4
01E	15	025		DIS	I
01F	35	026		DIS	TCNTI
020	EB88	027		DJNZ	R3,088
022	05	028		EN	I
023	25	029		EN	TCNTI
024	75	030		ENTO	CLK
025	09	031		IN	A,P1
026	17	032		INC	A

```
LOC     OBJ      LINE     LABEL  SOURCE CODE
027     1B       033             INC    R3
028     11       034             INC    @R1
029     08       035             INS    A,BUS
02A     7244     036             JB3    044
02C     F644     037             JC     044
02E     B644     038             JF0    044
030     7644     039             JF1    044
032     0404     040             JMP    004
034     B3       041             JMPP   @A
035     E603     042             JNC    003
037     8603     043             JNI    003
039     2603     044             JNTO   003
03B     4633     045             JNT1   033
03D     9633     046             JNZ    033
03F     1606     047             JTF    006
041     3604     048             JT0    004
043     5604     049             JT1    004
045     C665     050             JZ     065
047     2355     051             MOV    A,#55
049     C7       052             MOV    A,PSW
04A     FC       053             MOV    A,R4
04B     F1       054             MOV    A,@R1
04C     42       055             MOV    A,T
04D     D7       056             MOV    PSW,A
04E     AC       057             MOV    R4,A
04F     BE44     058             MOV    R6,#44
051     A1       059             MOV    @R1,A
052     B144     060             MOV    @R1,#44
054     62       061             MOV    T,A
055     0C       062             MOVD   A,P4
056     3D       063             MOVD   P5,A
057     A3       064             MOVP   A,@A
058     E3       065             MOVP3  A,@A
059     81       066             MOVX   A,@R1
05A     91       067             MOVX   @R1,A
```

```
LOC     OBJ             LINE    LABEL   SOURCE CODE
05B     00              068             NOP
05C     4C              069             ORL     A,R4
05D     41              070             ORL     A,@R1
05E     4344            071             ORL     A,#44
060     8833            072             ORL     BUS,#33
062     89FF            073             ORL     P1,#FF
064     8D              074             ORLD    P5,A
065     02              075     B065    OUTL    BUS,A
066     39              076             OUTL    P1,A
067     83              077             RET
068     93              078             RETR
069     E7              079             RL      A
06A     F7              080             RLC     A
06B     77              081             RR      A
06C     67              082             RRC     A
06D     E5              083             SEL     MB0
06E     F5              084             SEL     MB1
06F     C5              085             SEL     RB0
070     D5              086             SEL     RB1
071     65              087             STOP    TCNT
072     45              088             STRT    CNT
073     55              089             STRT    T
074     47              090             SWAP    A
075     2C              091             XCH     A,R4
076     21              092             XCH     A,@R1
077     31              093             XCHD    A,@R1
078     DB              094             XRL     A,R3
079     D1              095             XRL     A,@R1
07A     D344            096             XRL     A,#44
07C     20              097             XCH     A,@R0
07D     33              098             ENFCR           ;*
07E     01              099             IOL             ;**
```

Notes: * This instruction applies to 80CX48 only.
** This instruction applies to 80C48 only.

Appendix K

8031/8344

K.1 Hardware

NEW MICE-II 8031/8344 comes with two or three boards, when external memory is used, SUEM or HUEM is also required.

NEW MICE-II 8031/8344 provides the following basic features:

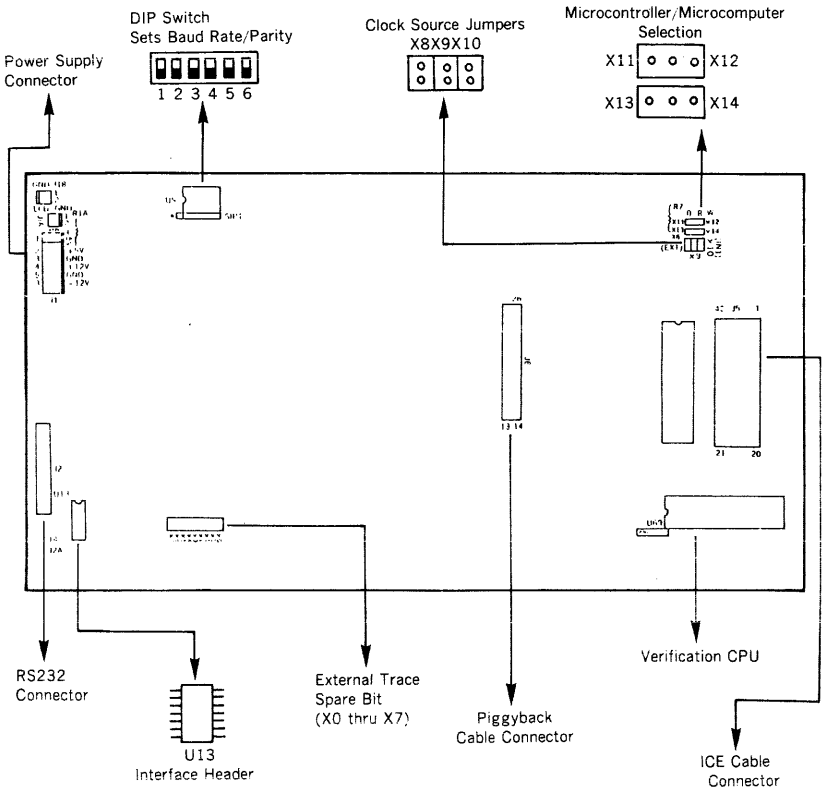
1. Full emulation support for the 8031, 8032 and 8344.

2. Display, modification, test and transfer of on-chip RAM.

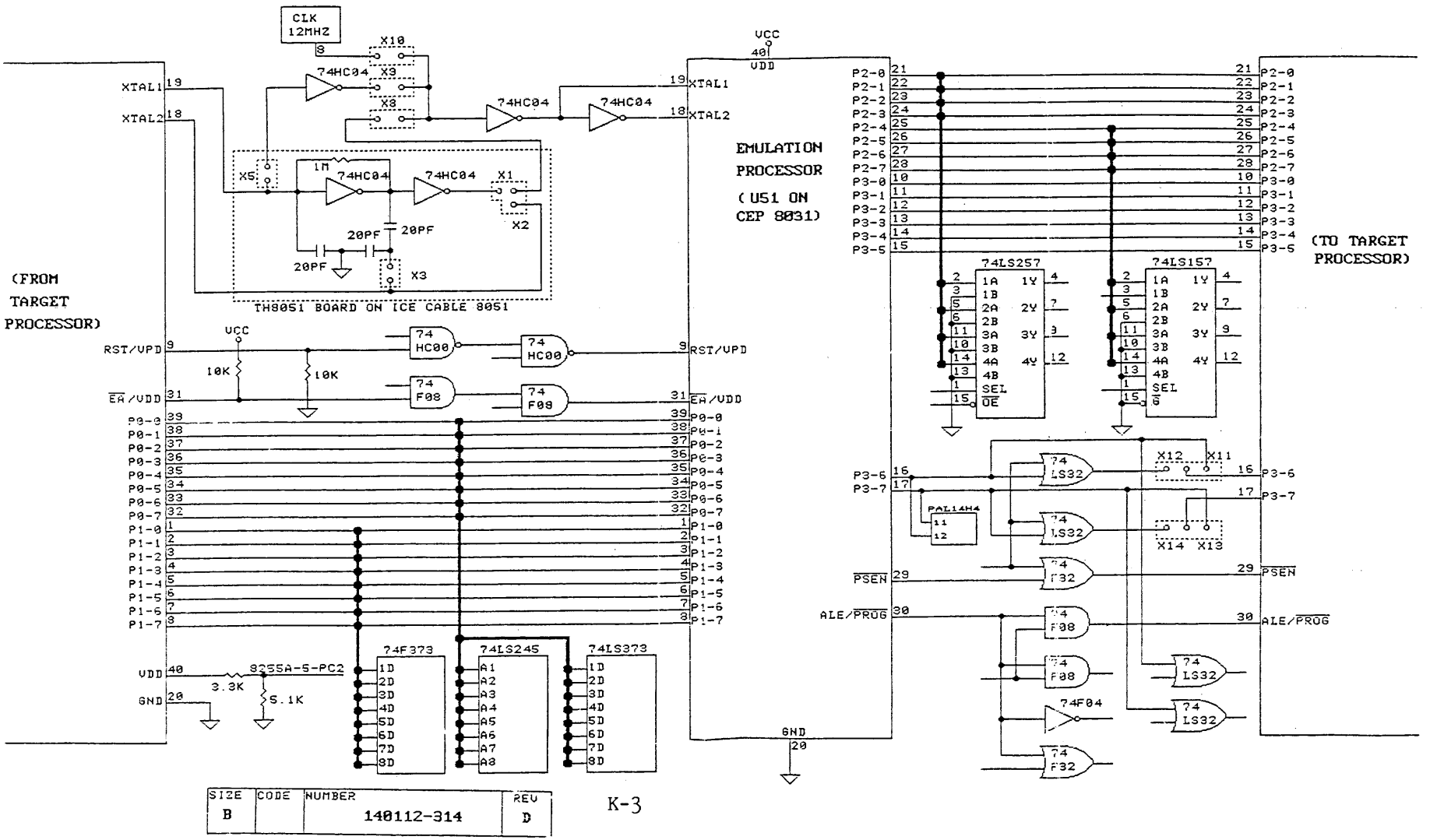
3. Display and modification for all special function and bank registers, as well as internal program verification by permitting:

   a) Display of entire on-chip internal program (EP)ROM for the 8051/8751.
   b) Modification/execution of on-chip (EP)ROM for the 8051/8751 by transferring on-chip program memory to the CEP emulation memory.

K.1.1 Control Emulation Processor Board, CEP-8031



K.1.2 Interface Diagram from Target Processor to CEP-8031



SIZE	CODE	NUMBER	REV
B		140112-314	D

K.2 Application Note for the 8031/8344 Command Set

The 8031/8344 families have no HALT or READY state. When the emulation CPU is stopped by any MICE command, it will execute a "SJMP $" instruction to prevent the program counter from changing to another location.

K.3 8031 Assembly Mnemonic Code Summary, with Disassembly Examples

LOC	OBJ	LINE	LABEL	SOURCE CODE
0000	3123	0001		ACALL 0123
0002	28	0002		ADD A,R0
0003	2512	0003		ADD A,12
0005	26	0004		ADD A,@R0
0006	2412	0005		ADD A,#12
0008	39	0006		ADDC A,R1
0009	3534	0007		ADDC A,34
000B	37	0008		ADDC A,@R1
000C	3434	0009		ADDC A,#34
000E	4134	0010		AJMP 0234
0010	5A	0011	B0010	ANL A,R2
0011	5556	0012		ANL A,56
0013	56	0013		ANL A,@R0
0014	5456	0014		ANL A,#56
0016	5278	0015		ANL 78,A
0018	539A78	0016		ANL 9A,#78
001B	8280	0017		ANL C,P0.0
001D	B089	0018		ANL C,/TCON.1
001F	B580EE	0019		CJNE A,P0,0010
0022	B49AFB	0020		CJNE A,#9A,0020
0025	BBBC08	0021		CJNE R3,#BC,0030
0028	B7DE15	0022		CJNE @R1,#DE,0040
002B	E4	0023		CLR A

```
LOC     OBJ       LINE     LABEL     SOURCE  CODE
002C    C292      0024               CLR     P1.2
002E    C3        0025               CLR     C
002F    C29B      0026               CLR     SCON.3
0031    F4        0027               CPL     A
0032    B3        0028               CPL     C
0033    B2A4      0029               CPL     P2.4
0035    D4        0030               DA      A
0036    14        0031               DEC     A
0037    84        0032               DIV     AB
0038    1C        0033               DEC     R4
0039    15AF      0034               DEC     AF
003B    16        0035               DEC     @R0
003C    DC12      0036               DJNZ    R4,0050
003E    D5DE1F    0037               DJNZ    DE,0060
0041    04        0038               INC     A
0042    0D        0039               INC     R5
0043    0588      0040               INC     TCON
0045    06        0041               INC     @R0
0046    A3        0042               INC     DPTR
0047    20AD26    0043               JB      IE.5,0070
004A    10B633    0044               JBC     P3.6,0080
004D    4031      0045               JC      0080
004F    73        0046               JMP     @A+DPTR
0050    30BFFD    0047     B0050     JNB     IP.7,0050
0053    507F      0048               JNC     00D4
0055    7080      0049               JNZ     FFD7
0057    60FE      0050     B0057     JZ      0057
0059    120200    0051               LCALL   0200
005C    020300    0052               LJMP    0300
005F    EE        0053               MOV     A,R6
0060    E5DC      0054     B0060     MOV     A,DC
0062    E7        0055               MOV     A,@R1
0063    74FE      0056               MOV     A,#FE
0065    FF        0057               MOV     R7,A
```

LOC	OBJ	LINE	LABEL	SOURCE	CODE
0066	A8BA	0058		MOV	R0,BA
0068	79DC	0059		MOV	R1,#DC
006A	F590	0060		MOV	P1,A
006C	8A76	0061		MOV	76,R2
006E	853254	0062		MOV	54,32
0071	8610	0063		MOV	10,@R0
0073	7513BA	0064		MOV	13,#BA
0076	F7	0065		MOV	@R1,A
0077	A624	0066		MOV	@R0,24
0079	7798	0067		MOV	@R1,#98
007B	A2D0	0068		MOV	C,PSW.0
007D	92E1	0069		MOV	ACC.1,C
007F	901234	0070		MOV	DPTR,#1234
0082	93	0071		MOVC	A,@A+DPTR
0083	83	0072		MOVC	A,@A+PC
0084	E2	0073		MOVX	A,@R0
0085	E0	0074		MOVX	A,@DPTR
0086	F3	0075		MOVX	@R1,A
0087	F0	0076		MOVX	@DPTR,A
0088	A4	0077		MUL	AB
0089	00	0078		NOP	
008A	4A	0079		ORL	A,R2
008B	4557	0080		ORL	A,57
008D	46	0081		ORL	A,@R0
008E	4476	0082		ORL	A,#76
0090	4268	0083		ORL	68,A
0092	437954	0084		ORL	79,#54
0095	72F2	0085		ORL	C,B.2
0097	A0FD	0086		ORL	C,/F8.5
0099	D08A	0087		POP	8A
009B	C09B	0088		PUSH	9B
009D	22	0089		RET	
009E	32	0090		RETI	
009F	23	0091		RL	A

LOC	OBJ	LINE	LABEL	SOURCE	CODE
00A0	33	0092		RLC	A
00A1	03	0093		RR	A
00A2	13	0094		RRC	A
00A3	D2EF	0095		SETB	E8.7
00A5	D3	0096		SETB	C
00A6	D2DB	0097		SETB	D8.3
00A8	80FE	0098	B00A8	SJMP	00A8
00AA	9B	0099		SUBB	A,R3
00AB	95AC	0100		SUBB	A,AC
00AD	97	0101		SUBB	A,@R1
00AE	9432	0102		SUBB	A,#32
00B0	C4	0103		SWAP	A
00B1	CC	0104		XCH	A,R4
00B2	C5BD	0105		XCH	A,BD
00B4	C6	0106		XCH	A,@R0
00B5	D7	0107		XCHD	A,@R1
00B6	6D	0108		XRL	A,R5
00B7	65CE	0109		XRL	A,CE
00B9	66	0110		XRL	A,@R0
00BA	6410	0111		XRL	A,#10
00BC	62FF	0112		XRL	FF,A
00BE	630FFF	0113		XRL	0F,#FF

Appendix L

80515/80535

L.1 Hardware

NEW MICE-II 80515/80535 comes with two or three boards, depending on the user's specified emulation requirements. When only internal memory is used, the CEP-80515 and RTT boards are all that is required. However, when external memory is used, HUEM or SUEM is required. When replacing the CEP board on another NEW MICE-II model with the CEP-80515, HUEM/SUEM may be removed if desired.

NEW MICE-II 80515/80535 provides the following basic features:

1. Full emulation support for 80515 and 80535.

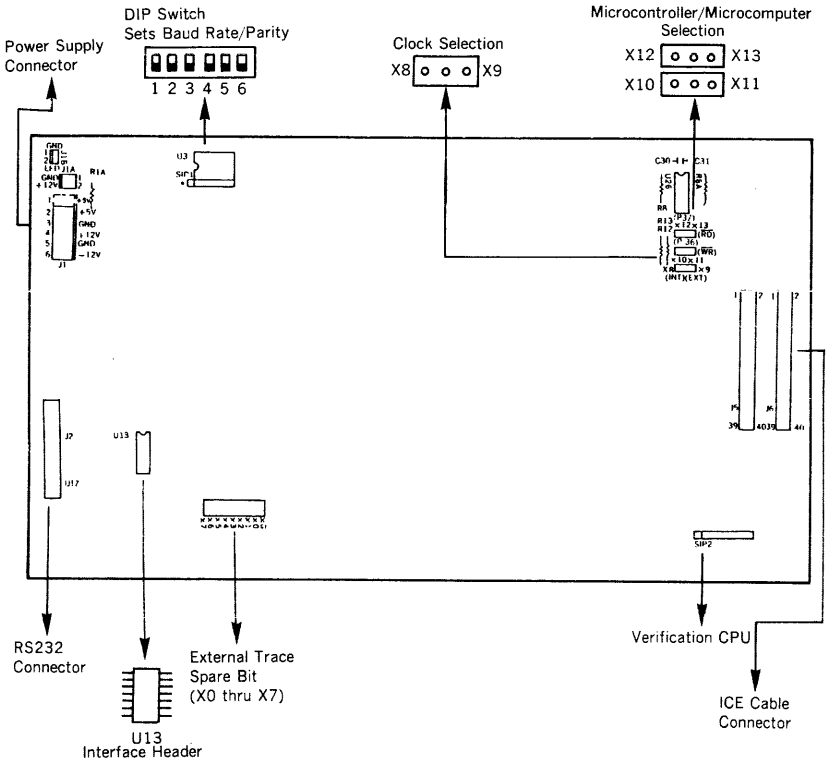
2. Display, modification, test and transfer of internal/external data memory.

3. Display and modification for all special function and bank registers.

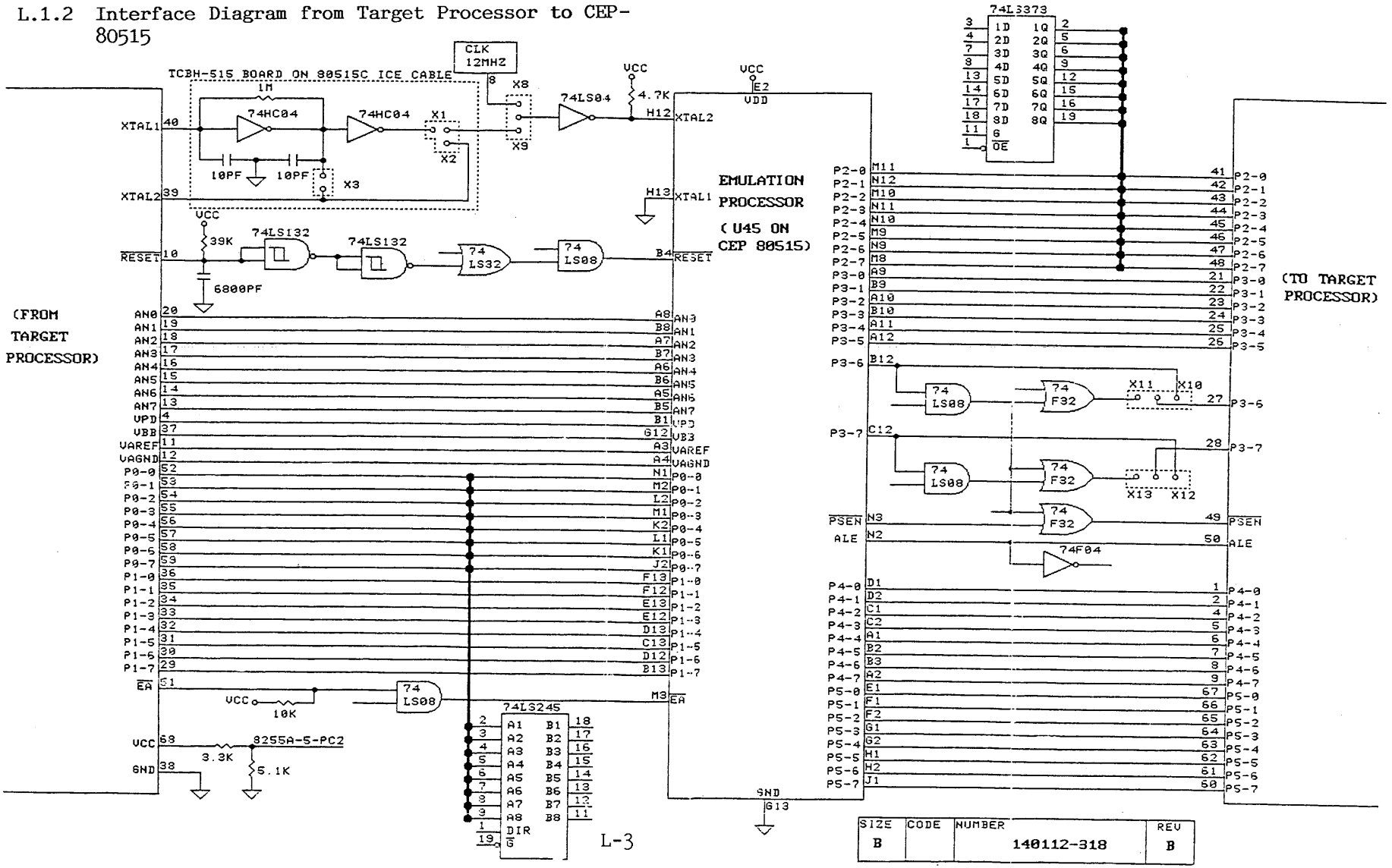
4. Internal program verification by permitting -

   a) Display of entire on-chip internal program ROM for the 80515.
   b) Modification/execution of the 80515 on-chip internal program (in ROM) by transferring it to CEP on-board internal emulation program memory.

L.1.1 Control Emulation Processor Board, CEP-80515



L.1.2 Interface Diagram from Target Processor to CEP-80515



L.2 Application Notes for the 80515/80535 Command Set

1. The 80515 family has no HALT or READY state. When the emulation CPU is stopped by any MICE command, it will execute a "SJMP $" instruction to prevent the PC from changing to another location; all the interrupts are disabled; and the timers, serial interface and watchdog timer are stopped.

2. If the emulation processor (EP) is stopped during the serial transmission or reception of an 8 or 9 bit character, the EP will wait for the transfer operation to be completed, thereby adversely affecting the internal logic for serial interface. The EP should only be stopped between the transmission or reception of two characters.

   TI or RI will not be at logical "1" unless the transfer operation has been completed. Serial transfer can be resumed by forcing TI or RI to "1" with the RBIT command; however note that the character where the transfer operation was stopped will be lost.

L.3 80515 Assembly Mnemonic Code Summary, with Disassembly Examples

Refer to section K.3.

Appendix M

8052/80C152

M.1 Introduction

NEW MICE-II 8052/80C152 basically is an extension to model 8051. It emulates more 8051 CPU families (e.g. 80C154, 83C154 and 8052) with the innovative design concept which the EPOD emulation device is applied.

M.2 Hardware

NEW MICE-II 8052/80C152 comes with three or four boards, depending on the user's specified emulation requirements. The CEP-8052, EPOD-8052 (Emulation POD) or EPOD-80C152 and RTT boards are all required, if only internal program memory is used (a maximum of 4K bytes is available when emulates the 8051/8751; 8K bytes for 8052/8752; 16K bytes for 83C154, 8K bytes for 83C152JA). A PVPOD-8052 is required when the programming and verification function are used. However, if external memory is used, you should also use either HUEM or SUEM. When replacing the CEP board on another MICE model with the CEP-8052, EPOD-8052/EPOD-80C152, HUEM/SUEM may be removed.

NEW MICE-II 8052/80C152 provides the following basic features:

1. Full emulation support for the 8051, 8751, 8052, 8752, 83C152JA, 83C154, 8031, 8032, 80C154, 80C152JA, 80C152JB, 80C51FA and 8344.

2. Display, modification, test and transfer of on-chip RAM.

3. Display and modification for all special functions and bank registers, as well as internal program verification by permitting:

   a) Display of entire on-chip internal program (EP)ROM for the 8051/8751, 8052/8752 and 83C154.*
   b) Modification/execution of on-chip (EP)ROM for the 8051/8751, 8052/8752 or 83C154 by transferring on-chip program memory to the CEP on-board emulation memory.*
   c) Programming and verification for 8751, 8752, 87C51FA and 8744.*

The 85C154VS must be used as the emulation processor for 8052/8752; for 8051/8751, use 80C51VS or 85C154VS as the emulation processor. For 8044/8744, 83C51FA/87C51FA, MICE cannot execute internal program memory (from 0-4 Kbytes).

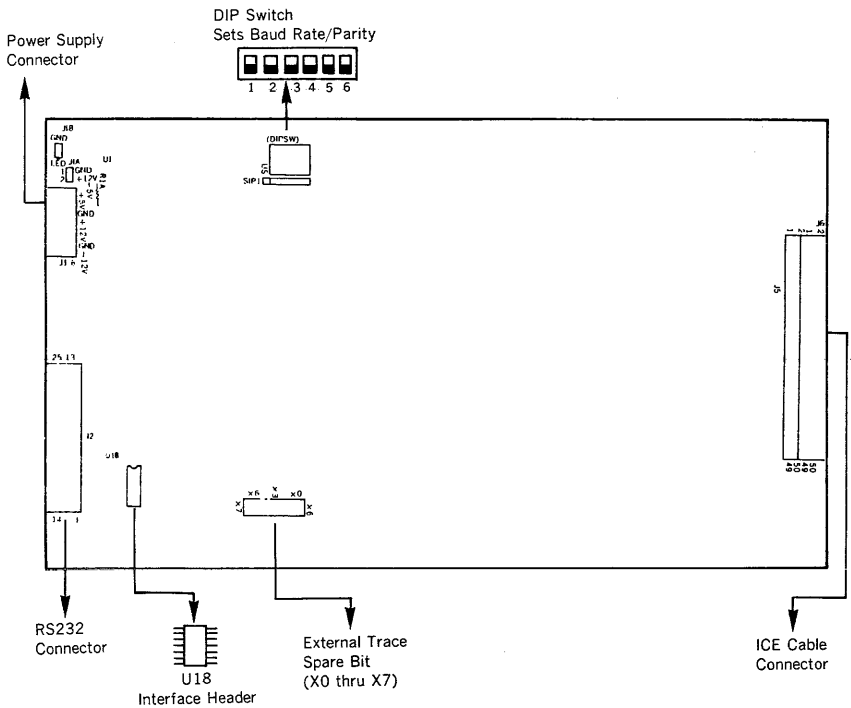
4. The MICE-8052 uses an 85C154VS (at EPOD-8052 U14) as the primary emulation CPU to emulate the 8051, 8751, 8031, 8052, 8752, 8032, 83C154 and 80C154. An 8344/8744/8044 may be used as the primary emulation CPU for 8344. An 80C51FA/83C51FA/87C51FA may be used as the primary emulation CPU for 80C51FA.

5. EPOD-80C152 must be used when EP is 83C152JA/80C152JA/80C152JB and F/W should be updated to V3.2 or later.

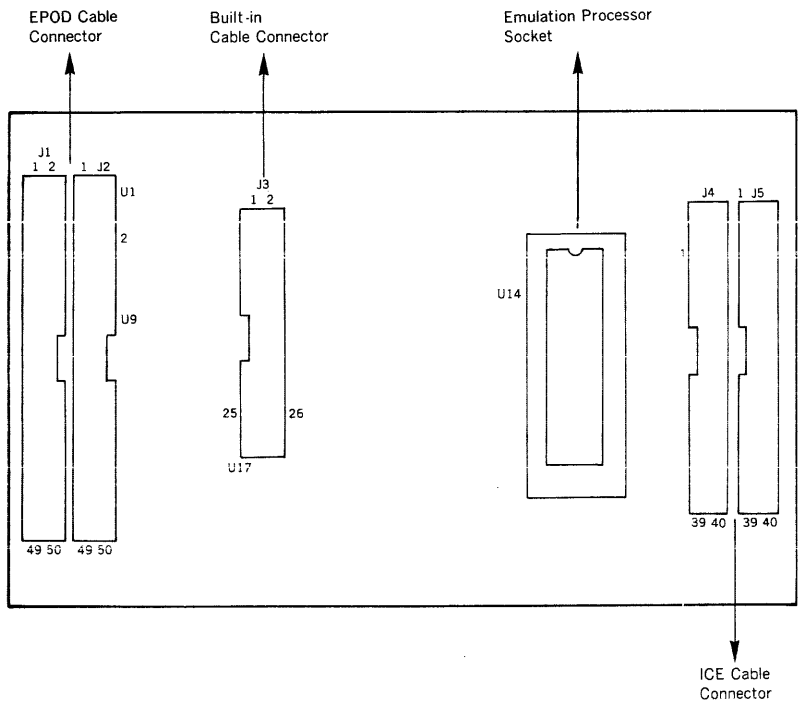
---

* These features supplied with the PVPOD-8052.

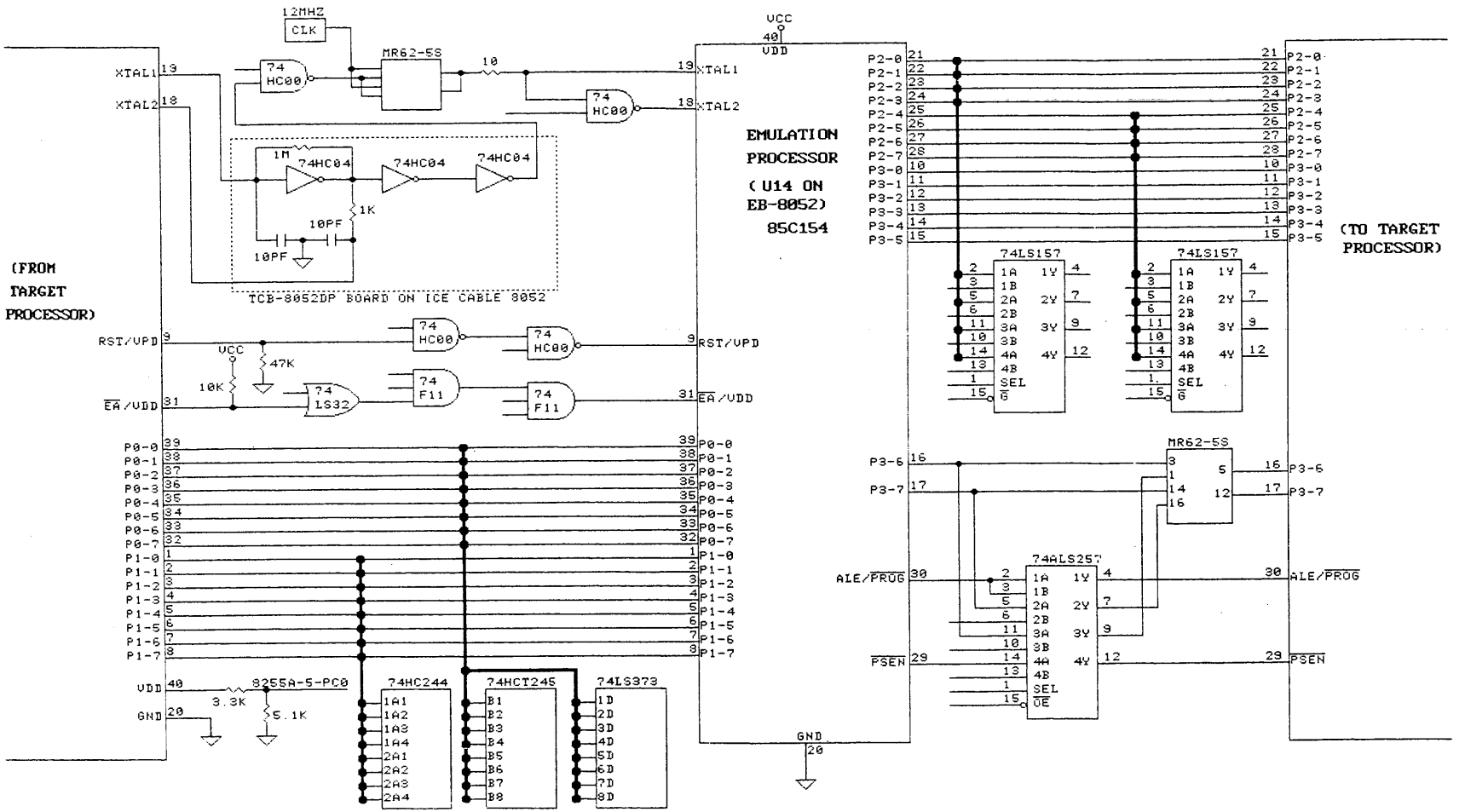
M.2.1 Control Emulation Processor Board, CEP-8052



M.2.2 Emulation Board, EB-8052

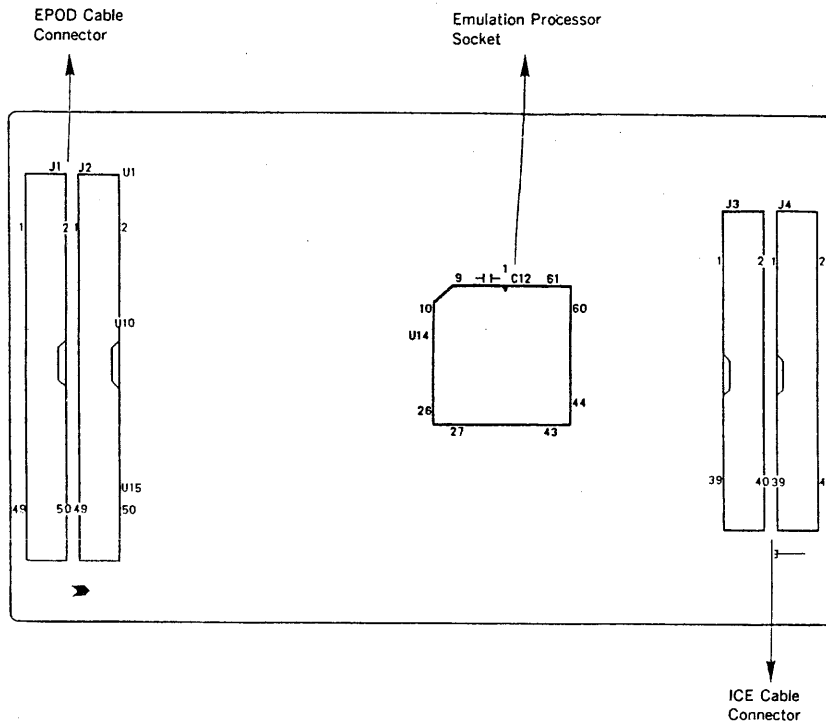


M.2.3 Interface Diagram from Target Processor to EB-8052

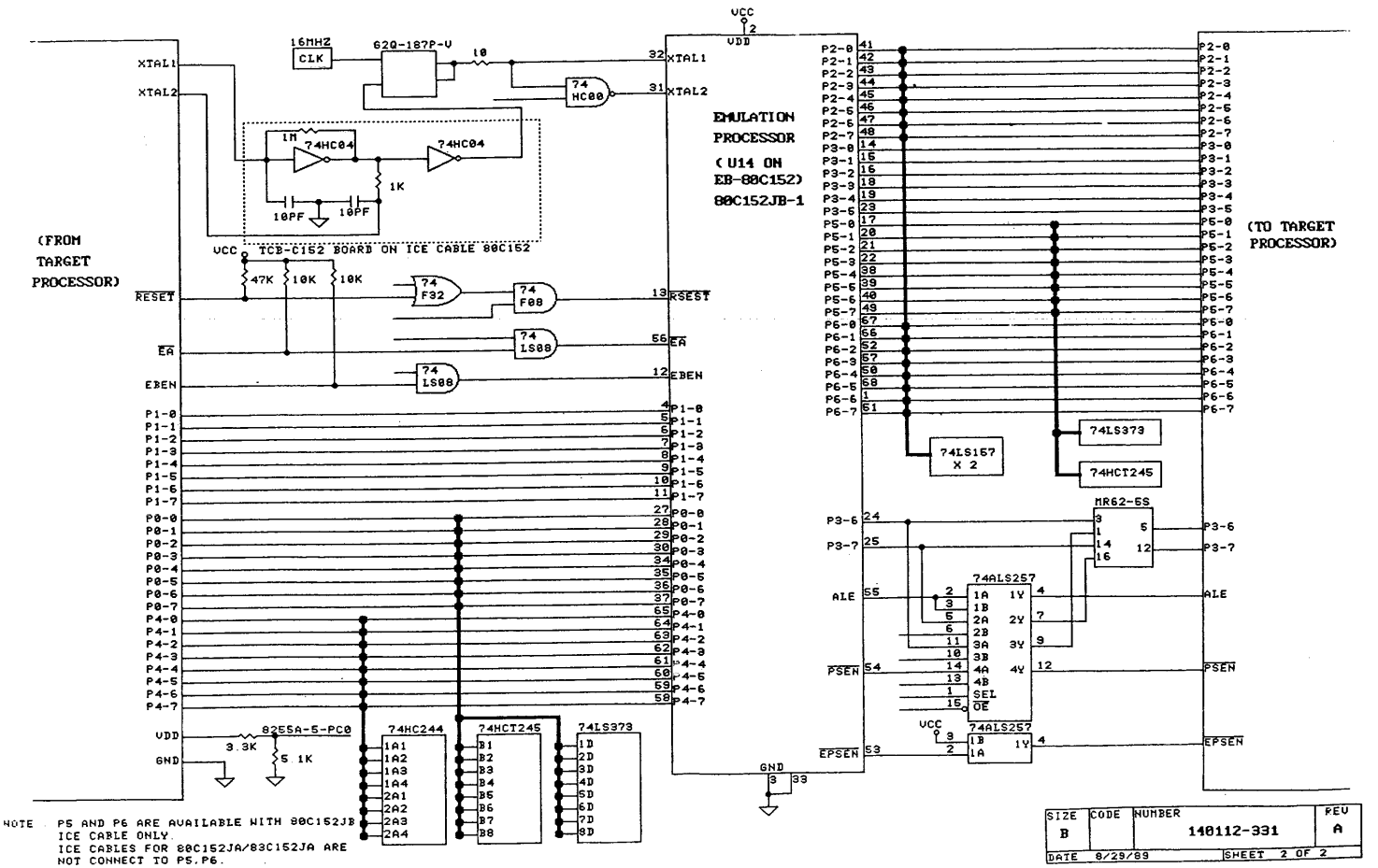


SIZE	CODE	NUMBER	REV
B		140112-325	A
DATE 8/15/88		SHEET 2 OF 2	

M.2.4 Emulation Board, EB-80C152



M.2.5 Interface Diagram from Target Processor to EB-80C152



M.3 Emulation Control Signals

Control characters for Disable and Enable commands are below with the control signal names and pin numbers.

M.3.1 A - external memory access

$\overline{EA}$ - pin 31 (EPOD-8052)/pin 58 (EPOD-80C152)

1. If this signal is enabled, the selected target microprocessor to be emulated depends on the target $\overline{EA}$ signal status. If 8051/8751/8052/8752, 83C152JA* or 83C154 (0-4KB, 0-8KB or 0-16KB internal program memory) is the target's processor, the target $\overline{EA}$ signal must be set to HIGH. If 8031/8032/8344/80C51FA/80C152JA/80C154 is the target's processor, then the $\overline{EA}$ signal must be set to LOW. Note that the piggyback connector** (EPOD-8052 only) must be attached between J3 and the emulation CPU on EB-8052 whenever the $\overline{EA}$ signal is enabled.

---

* For 80C152JB, the input status are EA = 1, EBEN = 1 to emulate 83C152JA.

** when the piggyback cable is connected; the EA pin signal is controlled by EA/DA commands and the target signal; EA pin signal is always LOW when disconnected (i.e. EA/DA commands and the target will have no control over the signal).

Command	J3	Target $\overline{EA}$	Target CPU
EA	ON	0	8031/8032/80154
EA	ON	1	8051/8751/8052/8752/83C154
DA	x	x	8031/8032/8344/80C154
x	OFF	x	8031/8032/8344/80C154

x: don't care

2. If this signal is disabled, the target defaults to 8031/8032/8344/80C51FA/80C154; and the status of the piggyback cable is "don't care".

3. After power-on, MICE defaults to EA mode (i.e. the $\overline{EA}$ signal is controlled by the target).

4. When power-up, MICE checks the $\overline{EA}$ pin and displays the target processor type currently being emulated. If the CPU on EPOD-8052 U14 is 85C154, the EA/DA commands can change the target processor type. The "!" command may be used to verify the current target. The "!M" command is used to set emulated processor type.

```
Example: *** MICE-II 83154 V#.# ***
         >!
         EP83154 V#.#
         >DA
         >!
          EP80154 V#.#
         >EA
         >!M 51
         >!
          EP8051 V#.#
         >DA
         >!
          EP8031 V#.#
```

5. If EPOD-80C152 connects with MICE, the default type after power-up is 80C152JB. The "!M" command is used to set emulated processor type (8XC152JA, 80C152JB).

```
Example: *** MICE-II 80C152JB V#.# ***
         >!
         EP80C152JB V#.#
         >!M JA
         >!
          EP83C152JA V#.#
         >DA
         >!
          EP80C152JA V#.#
```

M.3.2 R - reset

$\overline{RST}$ - pin 9 (EPOD-8052)/pin 13 (EPOD-80C152)

After EP is stopped, all interrupts, the RST and all timers will be disabled.

M.3.3 E-EBEN (EPOD-80C152)

EBEN - pin 12
Only EP is 80C152JB

EBEN	$\overline{EA}$	Program Fetch via	$\overline{PSEN}$	$\overline{EPSEN}$	Comments
0	0	P0, P2	Active	Inactive	Addresses 0-0FFFFH
0	1	N/A	N/A	N/A	Invalid Combination
1	0	P5, P6	Inactive	Active	Addresses 0-0FFFFH
1	1	P5, P6 P0, P2	Inactive Active	Active Inactive	Addresses 0-1FFFFH Addresses > 2000H

Program Memory Fetches

M.3.4 Specify Program Memory and Data Memory Space

This is a software control function ineffective to any pin signals. Emulation memory (HUEM/SUEM) is configured by using the following commands.

DX - combined memory space
EX - segregated memory space

1. Key in the "DX" command to combine external program and data memory.

2. Key in the "EX" command to segregate external program and data memory.

3. After power-on, MICE defaults to EX mode (i.e. external program and data memory are segregated.)

4. Upload, Download and A/M/T/Z commands are not affected by DX and EX.

M.3.5 Port Settings for Microcontroller/Microcomputer Selection

By setting of port3 pins 6 & 7, MICE permits operation for standard microcontroller I/O functions or for microcomputer Read/Write function to external data memory. The default setting is DP.

Selection can be made to use port3 pin 6 for I/O or $\overline{WR}$ and port3 pin 7 for I/O or $\overline{RD}$ by using DP/EP command as follows :

Command	port3 pins 6 & 7
DP	I/O
EP	$\overline{RD}/\overline{WR}$

Microcontroller/Microcomputer Selection

M.3.6 Specify high byte address when MOVX @Ri executed

If an 8-bit address is being used (MOVX @Ri), the contents of the Port2 SFR remain at Port2 pins throughout target memory; but throughout HUEM/SUEM A8~A15 may be set to reflect the contents of the Port2 SFR or may be set to all 0 by using ES or DS command as indicated below, the default setting is DS.

Command	A8~A15
ES	P2
DS	0

Status of HUEM/SUEM A8~A15 during MOVX @Ri

M.4 Specify CEP On-Board Internal Emulation Program Memory Capacity

---

! [M[51[S|E]|52[S|E]|154]] (EPOD-8052)
! [M[JA|JB[E|D]]] (EPOD-80C152)

If emulation CPU (on EPOD-8052 U14) is 85C154VS and the piggyback cable was connected, user can key in the "!M" command to enable 4, 8 or 16 Kbytes of CEP on-board internal emulation program memory. The system default is for 16K bytes. To check which type now emulated, enter the !M command.

If EPOD-80C152 be used, user can key in the "!M" emulation CPU type (80C152JB, 80C152JA, 83C152JA)

! is the command keyword for attention.

S|E is an option when user selects to emulate 8051 or 8052. MICE defaults to single-chip mode. Expand mode option is used when target system's address latcher failed to latch correct address in single-chip mode.

D|E is an option when user selects to emulate 80C152JB and EA=EBEN=1. When EPSEN active will access to emulation memory on CEP U73. User can use disable option to force MICE to access to target memory. MICE defaults to enable mode. In disable mode, don't enable emulation memory (on HUEM or SUEM) during 0H~1FFFH, otherwise, it won't work properly.

Example: Display the type used and select emulation type. (EPOD-8052)

```
      *** MICE-II 83154 V#.# ***
>!M
 83154
>!M 51                    ;Select to emulate 8051
>!M
 8051 SINGLE CHIP MODE
>!M 51 E
>!M
 8051 EXPAND MODE
>DA
>!M
 8031
>
```

Notes: 1) Only single chip mode can be chosen, if target external memory is not used (i.e. use port 0 and port 2 as I/O port instead).

2) When executing "!M" command to emulate target 8051 with emulation CPU 85C154VS, if the target doesn't function properly, user may correct the problem by changing the emulation CPU to 80C51VS.

Example: Display the type used and select emulation type. (EPOD-80C152)

```
      *** MICE-II 80C152JB V#.# ***

  >!
   EP80C152JB V#.#
  >!M
   80C152JB ENABLE EMULATION MEMORY (Through P5, P6)
  >!M JB D
   80C152JB DISABLE EMULATION MEMORY (Through P5, P6)
  >!M JA
  >!
   EP83C152JA V#.#
  >DA
  >!
   EP80C152JA V#.#
```

M.5 Register Command

```
R [S[ adr]|register|BIT[.symbol| start-adr[ end-adr[ data-1[ ...data-8]]]]|INT]
```

NEW MICE-II 8052 supports display/modification of registers and direct-addressable bits. Display of interrupt enable and pending status are also supported.

R — is the Register command. If no other parameters are specified, contents of the ACC, B, DPTR, PSW, SP, current bank registers and the PC are displayed. Note that the PSW displays both hex and binary value for the current bank registers.

S — displays additional contents of all Special Function Registers.

adr — is a hexadecimal pointer for a SFR register in the emulation CPU. The permissible address range is 80H~FFH.

register — are alphabetic characters indicating Special Function Registers, or bank registers, whose content is to be displayed or modified. Register mnemonics and addresses are listed in section M.5. (Note that all of the registers listed are 8-bit registers.)

BIT	indicates a specified operation for direct-addressable bits.
.symbol	is the mnemonic for a bit-addressable register location. All mnemonics must be prefixed by a period ".". (Refer to Special Function Register Bit Address, figure 7-4 for the 8051 family or figure 18-7 for the 8044 family, Intel Microcontroller Handbook, 1984.)
start-adr	is a hexadecimal pointer for a bit-addressable register in the emulation CPU indicating a memory location where the operation is to begin. The permissible address range is 0-FFH.
end-adr	is a hexadecimal pointer for a bit-adressable register in the emulation CPU indicating the last memory location of the specified range. The permissible address range is 0-FFH.
data-1...8	are binary values of "0" or "1". Note that if the address range (from start to end-address) is smaller than the block size (data1 - data8), then block fill is still executed but excess trailing data is ignored.
INT	displays interrupt enable and pending status.

M.5.1 Display/Modify Registers

R[S[a1]|register]

Example: Display contents for ACC, B, PSW, SP, DPTR, current bank registers and the PC. (The numbers enclosed in parentheses are register addresses.)

```
>R
 ACC  B  PSW  SP DPH DPL R0 R1 R2 R3 R4 R5 R6 R7    PSW     PC
 (E0)(F0)(D0)(81)(83)(82)                          CAFBBO-P
  12  23  34  45  56  67 31 32 33 34 35 36 37 38 00110100 0234
>
```

Example: Display SFR registers for 8051/8031.

```
>RS
 PCON    TCON    TMOD    TH0     TL0     TH1
 (87)    (88)    (89)    (8C)    (8A)    (8D)
  10      0E      00      00      00      00
 TL1     SCON    SBUF    IE      IP
 (8B)    (98)    (99)    (A8)    (B8)
  00      00      F7      40      44
 >
```

Example: Display SFR registers for 8052/8032.

```
>RS
  PCON    TCON    TMOD    TH0     TL0     TH1     TL1     SCON
  (87)    (88)    (89)    (8C)    (8A)    (8D)    (8B)    (98)
   10      0E      00      00      00      00      00      00
  SBUF    IE      IP      T2CON   CAP2H   CAP2L   TH2     TL2
  (99)    (A8)    (B8)    (C8)    (CB)    (CA)    (CD)    (CC)
   F7      40      44      00      00      00      00      00
>
```

Example: Display SFR registers for 83C154/80C154.

```
>RS
 PCON    TCON    TMOD    TH0     TL0     TH1     TL1     SCON    SBUF
 (87)    (88)    (89)    (8C)    (8A)    (8D)    (8B)    (98)    (99)
  10      08      00      00      00      00      00      00      FF
  IE      IP    T2CON   CAP2H   CAP2L    TH2     TL2    IOCON
 (A8)    (B8)    (C8)    (CB)    (CA)    (CD)    (CC)    (F8)
  40      40      00      00      00      00      00      00
>
```

Example: Display SFR registers for 83C152JA/80C152JA/ 80C152JB.

```
>R S
 GMOD    TFIFO   PCON    TCON    TMOD    TH0     TL0     TH1     TL1
 (84)    (85)    (87)    (88)    (89)    (8C)    (8A)    (8D)    (8B)
  00      FF      00      00      00      00      00      00      00
 DCON0   DCON1   BAUD    ADR0    SCON    SBUF    SARH0   SARL0   IFS
 (92)    (93)    (94)    (95)    (98)    (99)    (A3)    (A2)    (A4)
  76      00      00      00      00      FF      01      00      01
 ADR1     IE     SARH1   SARL1   SLOTTM  ADR2     IP     DARH0   DARL0
 (A5)    (A8)    (B3)    (B2)    (B4)    (B5)    (B8)    (C3)    (C2)
  00      60      FF      FF      20      00      E0      01      00
 BKOFF   ADR3    IEN1    DARH1   DARL1   TCDCNT  AMSK0   TSTAT   BCRH0
 (C4)    (C5)    (C8)    (D3)    (D2)    (D4)    (D5)    (D8)    (E3)
  9B      00      C0      FF      FF      00      00      84      00
 BCRL0   PRBS    AMSK1   RSTAT   BCRH1   BCRL1   RFIFO   MYSLOT  IPN1
 (E2)    (E4)    (E5)    (E8)    (F3)    (F2)    (F4)    (F5)    (F8)
  00      94      00      00      FF      FF      FF      00      C0
```

Example: Display SFR registers for 8344.

```
>RS
  PCON    TCON    TMOD    TH0     TL0     TH1     TL1     SCON    SBUF
  (87)    (88)    (89)    (8C)    (8A)    (8D)    (8B)    (98)    (99)
   FF      0C      00      00      00      00      00      FF      FF
   IE      IP      STS     SMD     RCB     RBL     RBS     RFL     STAD
  (A8)    (B8)    (C8)    (C9)    (CA)    (CB)    (CC)    (CD)    (CE)
   60      E4      00      00      D9      E1      0F      D9      9B
  DMA     NSNR    SIUST   TCB     TBL     TBS     FIFO1   FIFOM   FIFOH
  (CF)    (D8)    (D9)    (DA)    (DB)    (DC)    (DD)    (DE)    (DF)
   58      00      01      BB      BC      3F      93      99      F9
>
```

Example: Display SFR registers for 80C51FA.

```
>RS
  PCON    TCON    TMOD    TH0     TL0     TH1     TL1     SCON
  (87)    (88)    (89)    (8C)    (8A)    (8D)    (8B)    (98)
   30      00      00      00      00      00      00      00
  SBUF     IE     SADDR    IP     SADEN   T2CON   T2MOD   CAP2H
  (99)    (A8)    (A9)    (B8)    (B9)    (C8)    (C9)    (CB)
   00      00      00      80      00      00      FE      00
  CAP2L   TH2     TL2     CCON    CMOD    CCAPM0  CCAPM1  CCAPM2
  (CA)    (CD)    (CC)    (D8)    (D9)    (DA)    (DB)    (DC)
   00      00      00      20      38      80      80      80
  CCAPM3  CCAPM4   CH       CL     CCAP0H  CCAP0L  CCAP1H  CCAP1L
  (DD)    (DE)    (F9)    (E9)    (FA)    (EA)    (FB)    (EB)
   80      80      00      00      00      00      00      00
  CCAP2H  CCAP2L  CCAP3H  CCAP3L  CCAP4H  CCAP4L
  (FC)    (EC)    (FD)    (ED)    (FE)    (EE)
   00      00      00      00      00      00
```

Example: Modify the contents of the accumulator.

```
>R ACC<CR>
 ACC (E0) 12 5A<CR>
 B   (F0) 23<ESC>
>
```

Notes: 1) If emulation processor (U14 in EPOD-8052) is 83C154, 85C154VS, 80C154, don't modify .WDT or IOCON bit 7 to 1 when .TF1 bit is on, because when .TF1 and .WDT both are set to high, EP will generate a software reset.

2) When modify PCON SFR content, don't turn POWER DOWN MODE or IDLE MODE on, because it will let MICE lose control of EP and display "TARGET CAN'T STEP!".

M.5.2 Display/Modify Bit-Addressable Locations

```
RBIT[.symbol| start-address[ end-address[ data-1[ ...data-8]]]]
```

Example: Display all direct-addressable bits for the 8051/8031. The dash symbol "-" is used where no bit-addressable register exists for the indicated address.

```
>RBIT
    0 1 2 3 4 5 6 7 8 9 A B C D E F
00  0 0 1 0 0 0 1 0 1 0 0 0 1 0 1 0
10  1 0 1 0 1 0 1 0 0 0 1 0 0 0 1 0
20  1 0 0 0 1 0 1 0 1 0 0 0 1 0 1 0
30  0 0 1 0 1 0 1 0 1 0 1 0 0 0 0 0
40  1 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0
50  1 0 0 0 1 0 1 0 1 0 1 0 0 0 1 0
60  1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0
70  0 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0
80  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
90  1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
A0  1 1 1 1 1 1 1 1 0 0 0 0 0 - - 0
B0  1 1 1 1 1 1 1 1 0 0 0 0 0 - - -
C0  - - - - - - - - - - - - - - - -
D0  0 0 0 0 0 0 0 0 - - - - - - - -
E0  0 0 0 0 0 0 0 0 - - - - - - - -
F0  0 0 0 0 0 0 0 0 - - - - - - - -
>
```

Example: Display all direct-addressable bits for 8052/8032.

```
>RBIT
    0 1 2 3 4 5 6 7 8 9 A B C D E F
00  0 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1
10  0 1 1 1 1 0 1 1 0 1 1 1 0 1 1 1
20  0 0 1 1 0 0 1 1 0 0 1 1 0 1 1 1
30  1 0 1 1 1 1 0 1 1 0 0 0 0 0 0 1
40  0 0 0 1 0 1 0 1 0 0 1 0 0 0 0 1
50  0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1
60  0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0
70  0 1 0 1 0 0 1 1 0 1 1 1 0 1 1 0
80  0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
90  1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
A0  0 1 0 0 0 0 0 0 0 0 0 0 0 0 - 0
B0  1 1 1 1 1 1 1 1 0 0 0 0 0 0 - -
C0  - - - - - - - - 0 0 0 0 0 0 0 0
D0  0 0 0 0 0 0 0 0 - - - - - - - -
E0  0 0 0 0 0 0 0 0 - - - - - - - -
F0  0 0 0 0 0 0 0 0 - - - - - - - -
>
```

Example: Display all direct-addressable bits for 83C154/ 80C154.

```
>RBIT
   0 1 2 3 4 5 6 7 8 9 A B C D E F
00 0 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1
10 0 1 1 1 1 0 1 1 0 1 1 1 0 1 1 1
20 0 0 1 1 0 0 1 1 0 0 1 1 0 1 1 1
30 1 0 1 1 1 1 0 1 1 0 0 0 0 0 0 1
40 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0 1
50 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1
60 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0
70 0 1 0 1 0 0 1 1 0 1 1 1 0 1 1 0
80 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
90 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
A0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 - 0
B0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 - 0
C0 - - - - - - - - 0 0 0 0 0 0 0 0
D0 0 0 0 0 0 0 0 0 - - - - - - - -
E0 0 0 0 0 0 0 0 0 - - - - - - - -
F0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>
```

Example: Display all direct-addressable bits for 80C51FA.

```
>R BIT
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00  1  0  1  0  0  0  0  1  0  0  1  0  0  0  0  0
10  1  0  0  0  0  1  0  1  0  1  0  1  0  1  0  0
20  0  1  0  0  1  0  0  0  0  1  0  1  0  0  0  0
30  1  0  1  0  1  0  1  1  0  0  0  0  1  1  0  0
40  0  1  0  0  0  0  0  0  0  1  0  0  0  0  0  1
50  0  1  1  0  0  0  0  0  1  0  1  0  1  0  0  1
60  0  1  0  0  0  0  0  0  0  0  0  0  0  0  1  1
70  0  0  1  0  1  1  1  0  0  0  0  1  0  0  1  0
80  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
90  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0
A0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
B0  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  -
C0  -  -  -  -  -  -  -  -  0  0  0  0  0  0  0  0
D0  0  0  0  0  0  0  0  0  0  0  0  0  0  -  0  0
E0  0  0  0  0  0  0  0  0  -  -  -  -  -  -  -  -
F0  0  0  0  0  0  0  0  0  -  -  -  -  -  -  -  -
>
```

Example: Display all direct-addressable bits for 80C152JA/ 83C152JA/80C152JB

```
>R BIT
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00  1  1  1  1  0  1  1  1  0  1  0  1  1  0  1  1
10  0  0  0  1  0  0  1  1  1  0  0  1  0  0  0  1
20  0  0  0  1  1  1  0  0  1  0  0  1  1  0  0  0
30  1  0  1  1  1  0  0  1  1  1  1  0  1  1  1  1
40  1  0  1  1  1  1  1  1  1  1  0  1  0  0  1  0
50  0  1  1  1  0  0  1  1  0  1  1  1  1  0  1  1
60  1  1  0  0  0  0  1  0  1  0  0  0  1  1  1  0
70  1  0  0  1  0  1  1  1  1  0  1  1  0  1  1  1
80  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0
90  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0
A0  1  1  1  1  1  1  1  1  0  0  0  0  0  -  -  0
B0  1  1  1  1  1  1  1  1  0  0  0  0  0  -  -  -
C0  1  1  1  1  1  1  1  1  0  0  0  0  0  0  -  -
D0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1
E0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
F0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  -  -
>
```

Example: Display all direct-addressable bits for the 8344.

```
>RBIT
   0 1 2 3 4 5 6 7 8 9 A B C D E F
00 1 0 1 1 0 0 1 1 0 0 0 1 0 1 1 1
10 1 0 0 0 1 0 0 1 1 0 0 1 1 0 1 1
20 1 0 0 0 0 1 1 1 1 1 1 0 1 0 0 0
30 0 0 1 1 1 0 1 0 1 0 1 0 1 0 1 1
40 1 1 0 0 1 0 0 0 1 1 0 1 1 0 0 1
50 1 0 0 0 0 1 1 0 0 0 1 0 1 0 1 0
60 1 1 1 1 0 0 0 1 0 1 0 1 1 0 0 0
70 0 0 0 0 1 1 1 0 0 0 0 0 1 0 1 0
80 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
90 1 1 1 1 1 1 1 1 - - - - - - - -
A0 0 1 0 0 0 0 0 0 0 0 0 0 0 - - 0
B0 1 1 1 1 1 1 1 1 0 0 0 0 0 - - -
C0 - - - - - - - - 0 0 0 0 0 0 0 0
D0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
E0 0 0 0 0 0 0 0 0 - - - - - - - -
F0 0 0 0 0 0 0 0 0 - - - - - - - -
>
```

Example: Modify the bit register .EA, displaying the affected area both before and after register modification.

```
>RBIT A0 AF
   0 1 2 3 4 5 6 7 8 9 A B C D E F
A0 0 0 0 0 0 0 0 0 0 0 0 0 0 - - 0
>RBIT .EA
 AF 0 1<CR>
 B0 1 <ESC>
>RBITA0 AF
   0 1 2 3 4 5 6 7 8 9 A B C D E F
A0 0 0 0 0 0 0 0 0 0 0 0 0 0 - - 1
>
```

Example: Fill register bit range 56H-67H with binary values, displaying the affected area both before and after the fill operation.

```
>RBIT 56 67
   0 1 2 3 4 5 6 7 8 9 A B C D E F
50                 0 0 0 0 0 0 0 0 0
60 0 0 0 0 0 0 0 0
>RBIT 56 67 1 0 1 0 1 1 0 1
>RBIT 56 67
   0 1 2 3 4 5 6 7 8 9 A B C D E F
50                 1 0 1 0 1 1 0 1 1 0
60 1 0 1 1 0 1 1 0
>
```

M.5.3 Display Interrupt Enable and Pending Status

RINT

Example: Display interrupt enable and pending status for the 8051/8031. (The interrupt enable/disable register bits are listed in the 1st row corresponding pending status flags are listed in the 2nd row.)

```
>RINT
  EA  ES  EX0   EX1  ET0  ET1
   0   0   0     0    0    0
  RI  TI  IE0   IE1  TF0  TF1
   0   0   0     0    0    0
>
```

Example: Display interrupt enable and pending status for the 8052/8032/83C154/80C154.

```
 >RINT
   EA  ES  EX0  EX1  ET0  ET1  ET2
    0   0   0    0    0    0    0
   RI  TI  IE0  IE1  TF0  TF1  TF2
    0   0   0    0    0    0    0
 >
```

Example: Display interrupt enable and pending status for the 83C152JA/80C152JA/80C152JB.

```
>R INT
   EA    ES    EX0  EX1  ET0  ET1
    0     0     0    0    0    0
   RI    TI    IE0  IE1  TF0  TF1
    0     0     0    0    0    0
```

Example: Display interrupt enable and pending status for the 80C51FA.

```
>R INT
   EA    ES    EX0   EX1   ET0   ET1   ET2   CR
    0     0     0     0     0     0     0     0
   RI    TI    IE0   IE1   TF0   TF1   TF2   CF
    0     0     0     0     0     0     0     0
>
```

Example: Display interrupt enable and pending status for the 8344.

```
>RINT
  EA  ES  EX0  EX1  ET0  ET1
   0   0   0    0    0    0
  RI  SI  IE0  IE1  TF0  TF1
   0   0   0    0    0    0
>
```

M.6 Available Registers

The following is a list of accessable registers for the 8051/8751/8031.

Mnemonic	Register	Address
ACC	Accumulator	0E0H
B	B Register	0F0H
PSW	Program Status	0D0H
SP	Stack Pointer	81H
DPH DPTR	Data Pointer High Byte	83H
DPL	Data Pointer Low Byte	82H
IP	Interrupt Priority Control	0B8H
IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
TCON	Timer/Counter Control	88H
TH0	Timer/Counter 0 (high byte)	8CH
TL0	Timer/Counter 0 (low byte)	8AH
TH1	Timer/Counter 1 (high byte)	8DH
TL1	Timer/Counter 1 (low byte)	8BH
SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

In addition to the above registers, the 8032/8052/8752 also include the following registers.

Mnemonic	Register	Address
T2CON	Timer/Counter 2 Control	0C8H
TH2	Timer/Counter 2 (high byte)	0CDH
TL2	Timer/Counter 2 (low byte)	0CCH
CAP2H	Timer/Counter 2 Capture (high byte)	0CBH
CAP2L	Timer/Counter 2 Capture (low byte)	0CAH

In addition to the above registers of 8052, the 80C51FA also include the following registers.

Mnemonic	Register	Address
SADDR	Serial Address Compare	0A9H
SADEN	Serial Address Mask	0B9H
CCON	PCA Counter Control	0D8H
CMOD	PCA Counter Mode	0D9H
CH	PCA Timer/Counter (high byte)	0F9H
CL	PCA Timer/Counter (low byte)	0E9H
CCAPM0	PCA Modules Compare/Capture 0	0DAH
CCAPM1	PCA Modules Compare/Capture 1	0DBH
CCAPM2	PCA Modules Compare/Capture 2	0DCH
CCAPM3	PCA Modules Compare/Capture 3	0DDH
CCAPM4	PCA Modules Compare/Capture 4	0DEH
CCAP0H	PCA Compare/Capture 0 (high byte)	0FAH
CCAP0L	PCA Compare/Capture 0 (low byte)	0EAH
CCAP1H	PCA Compare/Capture 1 (high byte)	0FBH
CCAP1L	PCA Compare/Capture 1 (low byte)	0EBH
CCAP2H	PCA Compare/Capture 2 (high byte)	0FCH
CCAP2L	PCA Compare/Capture 2 (low byte)	0ECH
CCAP3H	PCA Compare/Capture 3 (high byte)	0FDH
CCAP3L	PCA Compare/Capture 3 (low byte)	0EDH
CCAP4H	PCA Compare/Capture 4 (high byte)	0FEH
CCAP4L	PCA Compare/Capture 4 (low byte)	0EEH

In addition to the above registers of 8052, the 80C154/ 83C154 also include the following registers.

Mnemonic	Register	Address
IOCON	I/O control register	0F8H

The following are accessable registers for the 83C152JA/ 80C152JA/80C152JB.

Mnemonic	Register	Address
ACC	Accumulator	0E0H
ADR0	GSC Match Address 0	095H
ADR1	GSC Match Address 1	0A5H
ADR2	GSC Match Address 2	0B5H
ADR3	GSC Match Address 3	0C5H
AMSK0	GSC Address Mask 0	0D5H
AMSK1	GSC Address Mask 1	0E5H
B	B Register	0F0H
BAUD	GSC Baud Rate	094H
BCRL0	DMA Byte Count 0 (Low)	0E2H
BCRH0	DMA Byte Count 0 (High)	0E3H
BCRL1	DMA Byte Count 1 (Low)	0F2H
BCRH1	DMA Byte Count 1 (High)	0F3H
BKOFF	GSC Backoff Timer	0C4H
DARL0	DMA Destination ADDR 0 (Low)	0C2H
DARH0	DMA Destination ADDR 0 (High)	0C3H
DARL1	DMA Destination ADDR 1 (Low)	0D2H
DARH1	DMA Destination ADDR 1 (High)	0D3H
DCON0	DMA Control 0	092H
DCON1	DMA Control 1	093H
DPH DPTR	Data Pointer (High)	083H
DPL	Data Pointer (Low)	082H
GMOD	GSC Mode	084H
IE	Interrupt Enable Register 0	0A8H
IEN1	Interrupt Enable Register 1	0C8H
IFS	GSC Interframe Spacing	0A4H
IP	Interrupt Priority Register 0	0B8H
IPN1	Interrupt Priority Register 1	0F8H
MYSLOT	GSC Slot Address	0F5H
PCON	Power Control	087H
PRBS	GSC Pseudo-Random Sequence	0E4H
PSW	Program Status Word	0D0H

RFIFO	GSC Receive Buffer	0F4H
RSTAT	Receive Status (DMA & GSC)	0E8H
SARLO	DMA Source ADDR 0 (Low)	0A2H
SARHO	DMA Source ADDR 0 (High)	0A3H
SARL1	DMA Source ADDR 1 (Low)	0B2H
SARH1	DMA Source ADDR 1 (High)	0B3H
SBUF	Local Serial Channel (LSC) Buffer	099H
SCON	Local Serial Channel (LSC) Control	098H
SLOTTM	GSC Slot Time	0B4H
SP	Stack Pointer	081H
TCDCNT	GSC Transmit Collision Counter	0D4H
TCON	Timer Control	088H
TFIFO	GSC Transmit Buffer	085H
TH0	Timer 0 (High)	08CH
TH1	Timer 1 (High)	08DH
TL0	Timer 0 (Low)	08AH
TL1	Timer 1 (Low)	08BH
TMOD	Timer MODE	089H
TSTAT	Transmit Status (DMA & GSC)	0D8H

The following are accessable registers for the 8344/8744/8044.

Mnemonic	Register	Address
B	B Register	F0H
ACC	Accumulator	E0H
FIFOH	FIFO	DFH
FIFOM	FIFO	DEH
FIFOL	FIFO	DDH
TBS	Transmit Buffer Start	DCH
TBL	Transmit Buffer Length	DBH
TCB	Transmit Control Byte	DAH
SIUST	SIU State Counter	D9H
NSNR	Send Count Receive Count	D8H
PSW	Program Status Word	D0H
DMA	DMA Count	CFH
STAD	Station Address	CEH
RFL	Receive Field Length	CDH
RBS	Receive Buffer Start	CCH
RBL	Receive Buffer Length	CBH
RCB	Receive Control Byte	CAH
SMD	Serial Mode	C9H
STS	Status Register	C8H
IP	Interrupt Priority Control	B8H
IE	Interrupt Enable Control	A8H
TH1	Timer High 1	8DH
TH0	Timer High 0	8CH
TL1	Timer Low 1	8BH
TL0	Timer Low 0	8AH
TMOD	Timer Mode	89H
TCON	Timer Control	88H
DPH DPTR	Data Pointer High	83H
DPL	Data Pointer Low	82H
SP	Stack Pointer	81H

M.7 Processor clock control and clock rate detection

K [I|T]

K is the clock selection. If no parameter is specified, inputting "K<CR>" will display the current clock selection and clock rate for the emulation processor.

I is internal clock and detects the current clock rate.

T select the external target clock/crystal and detects the current clock rate.

After power on, MICE defaults to the internal clock.

Example: Display clock selections.

```
>K                                  ;Display clock selection.
CLOCK SOURCE IS INTERNAL : 12.000 MHZ.  ;Internal clock 12MHZ.
>KT                                 ;Change to external clock/crystal.
CLOCK SOURCE IS EXTERNAL : ##.### MHZ.  ;External clock rate.
>
```

Notes: 1) The adaptor (TCB-8052DP) on the ICE-CABLE (8052DP) supports the XTAL input or clock input (to X1, and X2 N.C.) without jumpers setting, this adaptor supports the XTAL input from 6MHZ to 16 MHZ input. To support XTAL below 6MHZ, user may care the Target X1 input without capacitance and X2 input with 68PF approximate.

2) After changing CLOCK source, MICE stops the

processor and forces a reset.

3) When inputting "K<cr>" (without I or T option), the first time after power on or reset (X command), will detect the current clock rate and stop the processor. Subsequent input of the same command will only display current clock status and clock rate (detected during initial "K<cr>") without stopping the processor if it is in running mode.

4) If I or T option is selected, MICE will always detects the current clock rate and stop the processor.

5) The maximum clock rate can be detected is 16.384MHZ.

M.8 Download memory type selection

W [P|X]

NEW MICE-II 8052 supports download to data memory (uses MOVX instruction and memory is segregated) or program memory area. MICE default is downloading to program memory area. User may identify the type of download before using download command.

W	is the write type command for download. If no parameter is specified, current setting is displayed.
P	selects program memory area to download data.
X	selects data memory area to download data.

Note: If SUEM has been used and transmission data bit is 8, W command will be ineffective when download range is overlaid.

M.9 Upload command

U start-address end-address [T|I]|[P|X]

U	is the command keyword for Upload.
start-address	is a hexadecimal address of the target processor's memory where data transfer begins.
end-address	is the last hexadecimal address of the target processor's memory where data transfer ends.
T	indicates the Tektronix record format.
I	indicates the Intel record format (as default setting).
P	selectes upload from program memory area (as default setting).
X	selectes upload form data memory area. (Uses MOVX instruction and memory is segregated.)

M.10 Application Notes for the Command Set

The 8051/8344 families have no HALT or READY state. When the emulation CPU is stopped by any MICE command, it will execute a "SJMP $" instruction to prevent the program counter from changing to another location.

M.11 8051 Assembly Mnemonic Code Summary, with Disassembly Examples

```
LOC     OBJ             LINE    LABEL     SOURCE CODE
0000    3123            0001              ACALL  0123
0002    28              0002              ADD    A,R0
0003    2512            0003              ADD    A,12
0005    26              0004              ADD    A,@R0
0006    2412            0005              ADD    A,#12
0008    39              0006              ADDC   A,R1
0009    3534            0007              ADDC   A,34
000B    37              0008              ADDC   A,@R1
000C    3434            0009              ADDC   A,#34
000E    4134            0010              AJMP   0234
0010    5A              0011    B0010     ANL    A,R2
0011    5556            0012              ANL    A,56
0013    56              0013              ANL    A,@R0
0014    5456            0014              ANL    A,#56
0016    5278            0015              ANL    78,A
0018    539A78          0016              ANL    9A,#78
001B    8280            0017              ANL    C,P0.0
001D    B089            0018              ANL    C,/TCON.1
001F    B580EE          0019              CJNE   A,P0,0010
0022    B49AFB          0020              CJNE   A,#9A,0020
0025    BBBC08          0021              CJNE   R3,#BC,0030
0028    B7DE15          0022              CJNE   @R1,#DE,0040
002B    E4              0023              CLR    A
```

LOC	OBJ	LINE	LABEL	SOURCE	CODE
002C	C292	0024		CLR	P1.2
002E	C3	0025		CLR	C
002F	C29B	0026		CLR	SCON.3
0031	F4	0027		CPL	A
0032	B3	0028		CPL	C
0033	B2A4	0029		CPL	P2.4
0035	D4	0030		DA	A
0036	14	0031		DEC	A
0037	84	0032		DIV	AB
0038	1C	0033		DEC	R4
0039	15AF	0034		DEC	AF
003B	16	0035		DEC	@R0
003C	DC12	0036		DJNZ	R4,0050
003E	D5DE1F	0037		DJNZ	DE,0060
0041	04	0038		INC	A
0042	0D	0039		INC	R5
0043	0588	0040		INC	TCON
0045	06	0041		INC	@R0
0046	A3	0042		INC	DPTR
0047	20AD26	0043		JB	IE.5,0070
004A	10B633	0044		JBC	P3.6,0080
004D	4031	0045		JC	0080
004F	73	0046		JMP	@A+DPTR
0050	30BFFD	0047	B0050	JNB	IP.7,0050
0053	507F	0048		JNC	00D4
0055	7080	0049		JNZ	FFD7
0057	60FE	0050	B0057	JZ	0057
0059	120200	0051		LCALL	0200
005C	020300	0052		LJMP	0300
005F	EE	0053		MOV	A,R6
0060	E5DC	0054	B0060	MOV	A,DC
0062	E7	0055		MOV	A,@R1
0063	74FE	0056		MOV	A,#FE

LOC	OBJ	LINE	LABEL	SOURCE	CODE
0065	FF	0057		MOV	R7,A
0066	A8BA	0058		MOV	R0,BA
0068	79DC	0059		MOV	R1,#DC
006A	F590	0060		MOV	P1,A
006C	8A76	0061		MOV	76,R2
006E	853254	0062		MOV	54,32
0071	8610	0063		MOV	10,@R0
0073	7513BA	0064		MOV	13,#BA
0076	F7	0065		MOV	@R1,A
0077	A624	0066		MOV	@R0,24
0079	7798	0067		MOV	@R1,#98
007B	A2D0	0068		MOV	C,PSW.0
007D	92E1	0069		MOV	ACC.1,C
007F	901234	0070		MOV	DPTR,#1234
0082	A5D4D2	0071		MOV	PSW.2,PSW.4*
0085	93	0071		MOVC	A,@A+DPTR
0086	83	0072		MOVC	A,@A+PC
0087	E2	0073		MOVX	A,@R0
0088	E0	0074		MOVX	A,@DPTR
0089	F3	0075		MOVX	@R1,A
008A	F0	0076		MOVX	@DPTR,A
008B	A4	0077		MUL	AB
008C	00	0078		NOP	
008D	4A	0079		ORL	A,R2
008E	4557	0080		ORL	A,57
0090	46	0081		ORL	A,@R0
0091	4476	0082		ORL	A,#76
0093	4268	0083		ORL	68,A
0095	437954	0084		ORL	79,#54
0098	72F2	0085		ORL	C,B.2
009A	A0FD	0086		ORL	C,/F8.5
009C	D08A	0087		POP	8A
009E	C09B	0088		PUSH	9B

LOC	OBJ	LINE	LABEL	SOURCE CODE	
00A0	22	0089		RET	
00A1	32	0090		RETI	
00A2	23	0091		RL	A
00A3	33	0092		RLC	A
00A4	03	0093		RR	A
00A5	13	0094		RRC	A
00A6	D2EF	0095		SETB	E8.7
00A8	D3	0096		SETB	C
00A9	D2DB	0097		SETB	D8.3
00AB	80FE	0098	B00A8	SJMP	00A8
00AD	9B	0099		SUBB	A,R3
00AE	95AC	0100		SUBB	A,AC
00B0	97	0101		SUBB	A,@R1
00B1	9432	0102		SUBB	A,#32
00B3	C4	0103		SWAP	A
00B4	CC	0104		XCH	A,R4
00B5	C5BD	0105		XCH	A,BD
00B7	C6	0106		XCH	A,@R0
00B8	D7	0107		XCHD	A,@R1
00B9	6D	0108		XRL	A,R5
00BA	65CE	0109		XRL	A,CE
00BC	66	0110		XRL	A,@R0
00BD	6410	0111		XRL	A,#10
00BF	62FF	0112		XRL	FF,A
00C1	630FFF	0113		XRL	0F,#FF

* This instruction is reserved for 8051/8052/8031/8032/8344, only defined by 83C154/80C154.

Appendix N

Error Messages

---

1. ADDRESS xxxx RAM ERROR!

   Condition: Memory addressed in Memory Test (T) command failed.

   Recovery: Verify that specified address is write enabled RAM; and then see if the address bus and/or data bus are open or shorted.

2. BACKWARD TRACE FAILS!

   Condition: Trace buffer is empty when the breakpoint is matched, or <ESC> is input, or the processor is stopped without stepping any machine cycles.

   Recovery: See if the READY signal is always inactive, or the HOLD signal is always active. Then see is the trigger address or a specified breakpoint (H1~H6) was matched immediately.

3. BC TABLE FULL

   Condition: Instructions branched to and called in the disassembly command exceed 900.

4. **BUS REQUESTING!**

   Condition: Target bus requesting signal always active.

   Recovery: Verify that target has a bus request signal.

5. **COMMAND TOO LONG!**

   Condition: Instruction string in assembly command exceeds 50 characters.

6. **ERROR!**

   Condition: Command syntax error.

7. **ERROR CODE!**

   Condition: Instruction error in disassembly command.

8. **ERROR CODE, TRY AGAIN!**

   Condition: Instruction error in assembly command.

9. **FORWARD TRACE FAILS!**

   Condition: Trace buffer is empty when the breakpoint is matched, or <ESC> is input, or the processor is stopped without stepping any machine cycles.

   Recovery: See if the trigger address was invalid, or a specified breakpoint was matched before the trace started.

10. Hx NOT SET!

    Condition: Specified breakpoint not set prior to executing BT command.

11. MEMORY SEARCH FAILURE!

    Condition: Data string not found within specified address range for Memory Search (M) command.

12. MEMORY VERIFICATION FAILURE!

    Condition: Data cannot be written to address memory in Modify (M) command.

    Recovery: Verify that specified address is valid and write enabled.

13. MEMORY WRITE FAILURE!

    Condition: Data cannot be written to addressed memory in assembly command.

    Recovery: Verify that specified address is valid and write enabled.

14. NO BPP CARD!

    Condition: BPP commands cannot be executed without a BPP card.

15. PROGRAM HALT!

Condition: Emulation processor halted by instruction.

Recovery: Use an interrupt or reset to recover; then perform a trace to see where the HALT instruction occurred.

16. TARGET CAN'T STEP! *

Condition: Emulation processor does not respond after applying power or inputting the software reset command "r" (see Section 2.10). There is a problem with emulation processor that prevents a program from executing; there are several possibilities which can generate this message:

1) There is no clock.
2) There is a problem with the address line.
3) Target does not support the "ready" signal.
4) There is a problem with the RTT, CEP or HUEM/UEM board.
5) Other component failure.

Recovery: Check to be sure the emulation processor clock signal is correct and verify that the clock source selection switch is properly adjusted (see CEP Placement Chart in the appropriate appendix for switch location).

17. TARGET IS NOT READY!

Condition: Target memory or input/output ready signal is inactive.

Recovery: Use a valid physical location in the command specification.

18. Ux -- FAILURE! *

Condition: MICE selftest indicates location number for failure device.

Recovery: Contact your nearest Microtek distributor for assistance.

19. WARNING: ONLY 6 BYTES ARE VALID

Condition: Define byte (DB) in assembly command exceeds 6 bytes.

20. WHAT? *

Condition: RS-232C communication with parity or framing error.

Recovery: Check interface setting for communications at U13.

* Potentially fatal error; if problem cannot be resolved, contact your nearest Microtek distributor for assistance.

APPENDIX O

LIMITED WARRANTY; service

Microtek International Inc. ("Microtek") **warrants** NEW MICE-II (the "Product") and the user's manual for the Product, to be free from physical defects for a period of twelve (12) months from the date of the original retail purchase. This warranty applies only to the original retail purchaser who bought this Product from an authorized Microtek representative. This warranty is void if the Product is damaged by improper or abnormal use or by accident, if the Product is altered or modified in any way, or if any attempt is made to repair the Product without authorization from Microtek.

If you find a Product to be defective, contact your authorized Microtek representative or Microtek. When you receive authorization to do so, return the Product as directed; include proof of purchase and purchase date. Microtek will correct physical defects in Products under warranty at no charge to you by repairing or, at its option, replacing such Products. THIS IS THE SOLE AND EXCLUSIVE REMEDY AVAILABLE FOR BREACH OF WARRANTY OR UNDER ANY OTHER LEGAL THEORY WITH RESPECT TO MICROTEK PRODUCTS. Product repairs not covered by warranty, and Product updates, are provided at a set rate.

ALL OTHER WARRANTIES AND REPRESENTATIONS, ORAL OR WRITTEN, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, ARE EXCLUDED AND DO NOT APPLY. Except as set forth in this limited warranty, the purchaser assumes the risk as to these Products' quali-

ty, accuracy, design, and performance.

It is solely the purchaser's responsibility to determine the suitability of these Products for each particular application. Microtek Products are in all events not suitable, and are not authorized, for use in connection with life support devices or systems, or in other devices or systems potentially injurious to life or health.

IN NO EVENT WILL MICROTEK BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY BREACH OF WARRANTY OR UNDER ANY OTHER LEGAL THEORY, even if advised of the possibility of such damages. Microtek is thus not liable for lost profits or goodwill; downtime; damage or destruction of any program, data, equipment, or other property; costs of recovering, reprogramming, or reproducing any program, data, or equipment; personal injury or loss; or any other damages.

Except as required by law, no representative, agent, or employee of Microtek is authorized to commit Microtek to warranties, representations, or obligations inconsistent with or in addition to those set forth in this limited warranty.

Please complete and return the registration section of the warranty card (provided in the packing box) to Microtek.

BPP SECTION

For 8-Bit Intel and Siemens Series

8085
8048
8031/8344
8052/80C152
80515/80535

Second Edition
June 1989

Microtek International, Inc.
6 Industry Road East 3
Science-Based Industrial Park
Hsinchu, Taiwan, 30077, R.O.C.
TEL: (035)772155
TLX: 32169 MICROTEK
FAX: (035)772598

TABLE OF CONTENTS

SECTION 1

INTRODUCTION

Microtek's BreakPoint Processor (BPP) is a comprehensive breakpoint control unit for all NEW MICE-II emulators. The BPP is an optional single card PCB that features sophisticated breakpoint logic. It includes up to 120 new breakpoint constructs for more flexibility in target system debug and development. The BPP also includes external hardware triggering and an execution interval timer. This powerful triggering system helps you to quickly solve the most difficult software and hardware bugs.

Breakpoint conditions designating precise events can be logically joined with **ARM/AND/OR** connectives, permitting you to -

* begin or end tracing
* activate another event group
* specify delayed trigger count
* initiate event counter
* use external signal as trace trigger
* send trigger signal to external device
* record execution time between two events
* break emulation

Flexible trigger constructs can be used to define single events, multiple activities or external hardware signals. These powerful triggers make it easy to specify complex processor and hardware sequences. Data breakpoints can be set for address, data, status, count and delay. While external hardware breakpoints can specify up to two triggers (each with any combination of two signals).

This breakpoint system combines a large number of trigger events, permitting you to begin target trace or break emulation on the following activities -

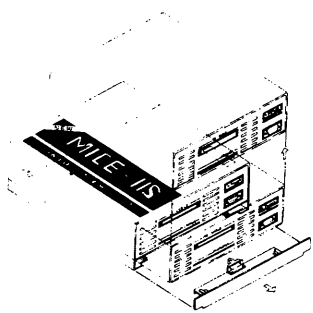
* Address Bus
* Data Bus
* Delay Count
* Event Counter (trigger match)
* Microprocessor Status Lines (Fetch, Read, Write, Input, Output and Interrupt Acknowledge)
* Logically Joined Breakpoints
* Sequence Levels
* Logical State Input

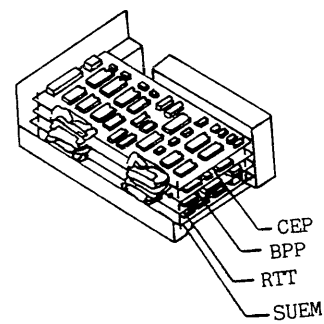
1.1 Installation

The BPP is a multilayer board added to the NEW MICE-II module.

To install the BPP board, first remove the cover to the NEW MICE-II case. With the power off, disconnect the two ribbon cable connectors; then remove the CEP and RTT boards. Replace the original brass spacers with the new ones provided with the BPP board. Reassemble all boards back into the MICE case, inserting the BPP board between the CEP and RTT boards. Then attach the new ribbon cable connectors (supplied with the BPP) firmly to each board.

Note: We recommend leaving the cover of the NEW MICE-II case off when using four boards, otherwise overheating may cause intermittent problems in operation.





1.2 Electrical and Environmental Specifications

Electrical Characteristics

Power	+5 VDC	(For BPP board)
Maximum current	1.3 Amp	

NEW MICE-II equips one built-in power supply.

Power Consumption: AC100-120V~:1.3A MAX
AC200-240V~:0.65A MAX
47-63HZ

Voltage Requirements: AC100-120V~ or
AC200-240V~
Factory-set

Environmental Characteristics

Operating temperature	0°~50°C	(32°~122°F)
Storage temperature	-10°~65°C	(14°~149°F)
Relative humidity	20~80%	

SECTION 2

BPP FUNCTIONS

The BPP includes four breakpoints (H3-H6), two triggers for transmitting a sync signal to an external device (BS1-BS2), and a Breakpoint Trigger Trace command (BT).

NEW MICE-II's other breakpoints (H1/H2) can be combined with the BPP commands; and H2~H6 can also serve as individual breakpoints for B/F/G commands (note that H1 can not be used for B/F commands).

2.1 Halt Breakpoints and Breakpoint Commands

Breakpoints:

H3/H4 are data breakpoints.

H5/H6 are external hardware breakpoints.

Commands:

BS is the Breakpoint Sync command. It outputs a sync signal (to an external device) when the specified trigger condition is matched.

BT is the Breakpoint Trace command. It begins tracing from the current program address up to the specified trigger condition (Backward Trace).

?B is the command to display BPP syntax as illustrated below:

```
>?B

H ?|[0][3|4|5|6]
H3 addx[ data[ <dbwc>][ q[ cnt]]]
H4 addx[ data[ <dbwc>][ q[ d1y]]]
H5 xx
H6 xx

BS 1|2[ addx[ data[<dbwc>][ q]]]

BT A[ B[ C[ D]]]
BT <A B C>
BT <A B>[ C[ D]]
BT (A B)[ C[ D]]
BT (A B C)[ D]
A-D ARE ANY OF H3~6
>
```

2.2 Breakpoint Syntax

For H3/H4

addx - is an up to 4 digit hex address with an optional byte pair of "XX".
data - is an up to 2 digit hex data.
<dbwc> - is an up to 2 digit hex setting; interpreted as 8 binary data bits, where "0" indicates "don't care".
q - is a qualifier for the type of processor activity, where "X" indicates "don't care".
cnt - is a trigger count; the range is 1~4000H (for nth match).
dly - is a cycle count delay; the range is 2~FFFFH.

For a detailed description refer to section 2.3.

For H5/H6

H5 [**xx**] - are 2 spare bits (X1 and X0).
H6 [**xx**] - are 2 spare bits (X3 and X2).

Spare bits are monitored signals input from stick headers X0~3 on the CEP board. They can be set to match signal input at HIGH, LOW or "don't care" (1/0/X). For a detailed description refer to section 2.4.

2.3 Data Breakpoints (H3/H4)

```
H3 [addx[ data [<dbwc>][ qualifier[ count]]]]
H4 [addx[ data [<dbwc>][ qualifier[ delay]]]]
```

H3 is the command for a Data Breakpoint with an optional event count.

H4 is the command for a Data Breakpoint with an optional cycle delay.

addx is the trigger address where emulation is to stop. This can be an up to 4 digit hex address with an optional byte pair of "XX".

data is an up to 2 digit hex data to be matched with the trigger address.

<dbwc> is an up to 2 digit hex setting indicating a "**data bit wildcard**". To enter any <dbwc>, a **data** value must first be defined. The data bit wildcard is interpreted as 8 binary data bits, where "0" indicates "don't care". The default value for the dbwc is FFH.

qualifier is a single or double alphabetic character indicating the type of processor activity to be matched with the trigger address. (If a **count** or **delay** is to be input, an "X" may be used in the status specification to indicate "don't care".)

For a complete listing of qualifiers applicable for a particular processor, refer to the appropriate appendix in the NEW MICE-II User's Manual.

count is a hexadecimal value from 1~4000H that specifies the number of times the target condition must be matched before the trace stops. To enter a count, a **qualifer** must first be defined.

delay is a hexadecimal value from 2~FFFFH that specifies a cycle count delay. After all other trigger conditions have been matched, the trace will continue until the specified cycle count has expired. To enter a delay, a **qualifier** must first be defined.

The breakpoints H3 and H4 are latched on the rising edge of a $\overline{\text{STROBE}}$ signal from the CEP board. When the breakpoint is activated, emulation stops at the end of the next bus cycle. (See Appendix C for a detailed description of break operation.) If a trigger count or cycle delay is defined, then the specified value must also be matched before the breakpoint is activated. This method is useful for tracing data after all other trigger conditions have been matched.

When the specified break condition is matched, a sync signal is output. Note that **delay** is not calculated for

the sync signal; as soon as all other trigger conditions are matched, the sync signal is transmitted without waiting for the specified delay to transpire. A count, however, specifies the number of times a sync signal will be transmitted. Everytime the trigger condition is matched and the count incremented, and a sync signal is output.

The following sample program for an 8085 target is used as the basis for all the examples listed in this user's manual. This program includes commonly used instructions for performing various bus cycles. Without a target system connected, it executes in a simple loop.

```
>Z
LOC     OBJ             LINE    LABEL     SOURCE CODE
0000    31F07F          0001              LXI   SP,7FF0
0003    213412          0002    B0003     LXI   H,1234
0006    DB00            0003              IN    00
0008    00              0004              NOP
0009    00              0005              NOP
000A    D300            0006              OUT   00
000C    E5              0007              PUSH  H
000D    212143          0008              LXI   H,4321
0010    0E06            0009              MVI   C,06
0012    71              0010    B0012     MOV   M,C
0013    23              0011              INX   H
0014    E3              0012              XTHL
0015    0D              0013              DCR   C
0016    C21200          0014              JNZ   0012
0019    E1              0015              POP   H
001A    C30300          0016              JMP   0003
     DISASSEMBLY COMPLETED
>
```

Example 1: Use the "H?" command to display all breakpoint settings; then set H3 at address 0000H and begin program execution.

```
>H?
H1=
H2=
H3= XXXX 12
H4=
H5=
H6=
>H0                    ;Clear H1-H6 settings.
>H3 0                  ;Set H3 at 0000H, for any data and any status.
>H3
H3= 0
>R                     ;Examine the registers prior to program execution.
  A  B  C  D  E  H  L  M  P    S   PC
 00 00 00 00 00 FF FF 07 54 7FF0 0000
>G                     ;Breakpoint matched at 0000H.
PROGRAM BROKE AT BREAKPOINT-3
>C                     ;Cycle command steps the program to address 0001H.
   IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
            0001    F0   R      11111111
            0002    7F   R      11111111 <ESC>
>
```

Note: For the 8085, if H3/H4 has been set and the trigger condition is matched at the current cycle, the CPU will stop at the next cycle (see Appendix C).

Example 2: Set H3 for any address and data to 12. (Note that without a dbwc, all binary bits of the specified data value are qualified.)

```
>J0                                  ;Reset the PC to 0000H.
>H3 XXXX 12                          ;Use H3 to break at data 12.
>G
PROGRAM BROKE AT BREAKPOINT-3
>C
   IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
    0006    0006    DB   S      11111111
            0007    00   R      11111111 <ESC>
>
```

Example 3: Set H3 with address and data of "don't care"; then set the status to Port Input with a count of 3.

```
>J0                              ;Reset the PC to 0000H.
>H3 XXXX FF <0> I 3              ;H3 set to break on 3rd input cycle.
>H?                              ;Display H1-H6 settings.
H1=
H2=
H3= XXXX FF <0> I 3
H4=
H5=
H6=
>G
PROGRAM BROKE AT BREAKPOINT-3
>C
   IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
    0008    0008    00   S      11110011
    0009    0009    00   S      11110011
    000A    000A    D3   S      11110011 <ESC>
>
```

Example 4: Set H4 to any address XXXX with data of "don't care"; then set the status to Port Output with a delay of 2.

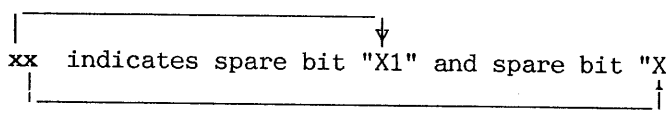
```
>J0                              ;Reset the PC to 0000H.
>H0                              ;Clear H1-H6 settings.
>H4 XXXX FF <0> O 2              ;H4 set for address XXXXH, output status
                                 ; and a delay of 2 cycles.

>G
PROGRAM BROKE AT BREAKPOINT-4
>C
   IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
    000D    000D    21   S      11110011
            000E    21   R      11110010 <ESC>
>
```

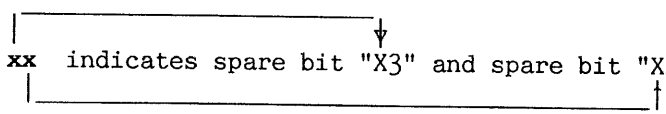
2.4 External Hardware Breakpoints (H5/H6)

H5 [xx]
H6 [xx]

H5 is the command for an External Hardware Breakpoint set by monitored signals at X1 and X0 on the CEP board.



H6 is the command for an External Hardware Breakpoint set by monitored signals at X3 and X2 on the CEP board.



The breakpoints H5 and H6 use two external signal groups on the CEP board (X1/X0 and X3/X2 respectively). Mini-jumpers provided with the BPP can be used to connect from these test points to the target. The monitored signals are called spare bits and support concurrent trace for address, data and status bus. Any of these bits can be monitored to provide hardware status based on machine cycles. Hardware breakpoints are latched when the specified logic state occurs.

The spare bits input from stick headers X0~3 on the CEP board can be set to match signal input at HIGH, LOW or "don't care" (1/0/X), where

1 = high level
0 = low level
X = don't care (either 1 or 0)

Note: Input signals to X0~3 must meet specifications for the Schmitt trigger IC - 74LS132.

To demonstrate External Hardward Breakpoints in the following examples, jumpers are connected to the CEP board of a NEW MICE-II 8085. The spare bits X0, X1, X2, X3 are configured as shown below:

X0 is connected to U29-29 (CEP 8085 S0 signal)
X1 is connected to U29-33 (CEP 8085 S1 signal)
X2 is connected to U29-34 (CEP 8085 IO/$\overline{M}$ signal)
X3 is connected to U29-34 (CEP 8085 IO/$\overline{M}$ signal)

The following table on 8085 bus cycles is a reference for checking the status of spare bits.

8085 BUS Cycle Status			
IO/$\overline{M}$	S1	S0	Status
0	1	1	Opcode Fetch
0	1	0	Memory Read
0	0	1	Memory Write
1	1	0	I/O Read
1	0	1	I/O Write
1	1	1	Interrupt Acknowledge

Example 1: Use H5 to break emulation on an Memory write cycle by setting the input trigger at 01 for X1/X0.

```
>J0                          ;Reset the PC to 0000H.
>H0                          ;Clear H1-H6 settings.
>H5 01                       ;Break condition for X0=1, X1=0.
>H?                          ;Display H1-H6 settings.
H1=
H2=
H3=
H4=
H5= 01
H6=
>G
PROGRAM BROKE AT BREAKPOINT-5
>C
   IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
            0000    00   O      11111101
    000C    000C    E5   S      11110011
            7FEF    12   W      11110001
            7FEE    34   W      11110001 <ESC>
>
```

Example 2: Use H6 to break emulation on an opcode fetch cycle by setting the input trigger at 00 for X3/X2.

```
>J0                              ;Reset the PC to 0000H.
>H0                              ;Clear H1-H6 settings.
>H6 00                           ;Break condition for X2=0, X3=0.
>H?
H1=
H2=
H3=
H4=
H5=
H6= 00
>G
PROGRAM BROKE AT BREAKPOINT-6
>C
   IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
    0000    0000    31   S      11110011
            0001    F0   R      11110010
            0002    7F   R      11110010
    0003    0003    21   S      11110011 <ESC>
>
```

SECTION 3

BREAKPOINT TRACE LOGIC

3.1 Breakpoint Trace - BT

Refer to the following sections for description of syntax.

BT is the command for Breakpoint Tracing.

Breakpoint Tracing begins at the current address (Backward Trace) and continues until all the specified breakpoints are matched. H3-H6 must be set prior to command execution, and may be specified as a single breakpoint or in logical combinations (ARM/AND/OR) with the BT command. Note that H1 and H2 (specified prior to execution, and not within the BT command definition) can also be used to add another logical OR to the trigger construct (refer to the last example in section 3.1.1). The BPP performs the following types of tracing:

Event Count Trace
Delay Trigger Trace
ARM Trigger Trace
AND Trigger Trace
OR Trigger Trace

3.1.1 Event Count Trace (with H3)

The address, data and qualifier of a specific program event can be defined along with a **count** in the H3 breakpoint. When the Event Count trigger is enabled, it is added as a logical OR to the breakpoint parameters defined in B/F commands. When enabled and then defined in the BT command, it can be used alone or in any legal combination with other breakpoints. (Also refer to the example in section 3.1.3.)

Example: Set break conditions for H3 at any address, any data, Write cycle and with a count of 2; then execute a trace using the BT command.

```
>J0                          ;Reset the PC to 0000H.
>H0                          ;Clear H1-H6 settings.
>H3 XXXX FF <0> W 2          ;Set the H3 event count trigger.
>H?                          ;Display H1-H6 settings.
H1=
H2=
H3= XXXX FF <0> W 2
H4=
H5=
H6=
>BTH3                        ;Execute a Breakpoint Trace with H3.
THE TRACE STOPS AT STEP 0011
>L0 0 FFFF W                 ;List all Write cycles between 0 and FFFFH.
 FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
  000F        7FEF    12   W      11110001
  0010        7FEE    34   W      11110001
>
```

Example: Set the break condition for H1 at address 0H, then execute a trace again using the BT command.

```
>J0                              ;Reset the PC to 0000H.
>H1 0                            ;Set H1 to address 0000H.
>H?                              ;Display H1-H6 settings.
H1= 0
H2=
H3= XXXX FF <0> W 2
H4=
H5=
H6=
>BTH3                            ;Execute a Breakpoint Trace with H3.
THE TRACE STOPS AT STEP 0008
 PROGRAM BROKE AT ADDRESS 0000
>L                               ;List all traced data.
 FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
  0000  0000     0000    31   S       11110011
  0001           0001    F0   R       11110010
  0002           0002    7F   R       11110010
  0003  0003     0003    21   S       11110011
  0004           0004    34   R       11110010
  0005           0005    12   R       11110010
  0006  0006     0006    DB   S       11110011
  0007           0007    00   R       11110010
  0008           0000    FF   I       11111110
>
```

3.1.2 Delay Trigger Trace (with H4)

The address, data and qualifier of specific program activity can be defined along with a **delay** in the H4 breakpoint. When the Delay Trigger is enabled, it is added as a logical OR to the breakpoint parameters defined in the B/F commands. When enabled and then defined in the BT command, it can be used alone or in any legal combination with other breakpoints. (Also refer the example in section 3.1.3.)

Example: Set the break conditions for H4 at address 4321, data FFH, Write cycle and with a delay of 3 machine cycles; then execute tracing using the BT command.

```
>J0                                  ;Reset the PC to 0000H.
>H0                                  ;Clear H1-H6 settings.
>H4 4321 FF <0> W 3                  ;Set H4 address 4321H, write cycle and
                                     ; trace delayed 3 cycles.
>H?                                  ;Display H1-H6 settings.
H1=
H2=
H3=
H4= 4321 FF <0> W 3
H5=
H6=
>BTH4                                ;Execute a Breakpoint Trace with H4.
THE TPACE STOPS AT STEP 001A
>L                                   ;List all traced data.
 FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
  0000  0000    0000   31   S       11110011
  0001          0001   F0   R       11110010
  0002          0002   7F   R       11110010
  0003  0003    0003   21   S       11110011
```

```
0004          0004   34   R      11110010
0005          0005   12   R      11110010
0006  0006    0006   DB   S      11110011
0007          0007   00   R      11110010
0008          0000   00   I      11111110
0009  0008    0008   00   S      11110011
000A  0009    0009   00   S      11110011
000B  000A    000A   D3   S      11110011
000C          000B   00   R      11110010
000D          0000   00   O      11111101
000E  000C    000C   E5   S      11110011
000F          7FEF   12   W      11110001
0010          7FEE   34   W      11110001
0011  000D    000D   21   S      11110011
0012          000E   21   R      11110010
0013          000F   43   R      11110010
0014  0010    0010   0E   S      11110011
0015          0011   06   R      11110010
0016  0012    0012   71   S      11110011
0017          4321   06   W      11110001    ;Π4 matched.
0018  0013    0013   23   S      11110011-
0019  0014    0014   E3   S      11110011 |-->;Trace delayed for 3
001A          7FEE   34   R      11110010-|   ; cycles then stopped.
>
```

3.1.3 Breakpoint Interval Timer

A timer is also supported for trigger trace mode which displays the interval from the initial trigger until emulation stops. (The interval will only display when an Armed breakpoint is encountered.) The timer can run for up to 35.39 minutes before the counter is reset. Note that resolution for the interval is down to micro-seconds. Time interval message syntax is as follows:

INITIAL TRIGGER TO BREAKPOINT INTERVAL-- aa:bb:ccc:ddd

where: **aa** = minute
bb = second
ccc = millisecond
ddd = microsecond

Example: Display the execution interval for an Armed Trigger Trace. (Refer to section 3.2 for a description of the Arm Trigger.)

```
>J0                              ;Reset the PC to 0000H.
>H0                              ;Clear H1-H6 settings.
>H3 XXXX 34 W                    ;Set H3 address to XXXXH.
>H4 4321 FF <0> W 3              ;Set H4 address to 4321H.
>H?
H1=
H2=
H3= XXXX 34 W
H4= 4321 FF <0> W 3
H5=
H6=
>BTH3 H4
THE TRACE STOPS AT STEP 001A
INITIAL TRIGGER TO BREAKPOINT INTERVAL--00:00:000:006
```

```
>LB
 FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
  000B  000A     000A    D3   S      11110011
  000C           000B    00   R      11110010
  000D           0000    00   O      11111101
  000E  000C     000C    E5   S      11110011
  000F           7FEF    12   W      11110001
  0010           7FEE    34   W      11110001    ;□3 matched.
  0011  000D     000D    21   S      11110011
  0012           000E    21   R      11110010
  0013           000F    43   R      11110010
  0014  0010     0010    0E   S      11110011
  0015           0011    06   R      11110010
  0016  0012     0012    71   S      11110011
  0017           4321    06   W      11110001    ;□4 matched.
  0018  0013     0013    23   S      11110011-
  0019  0014     0014    E3   S      11110011 |->;Trace delayed for 3
  001A           7FEE    34   R      11110010-|   ; cycles then stopped.
>
```

3.2 Armed Trigger Trace - BT

BT A[B[C[D]]]

A[B[C[D]]] Breakpoints A-D (which represent any of H3~6) can be logically joined in any Armed Trigger sequence. (The only exception being that they cannot be repeated).

There are up to 64 different Armed Trigger combinations. Sequential tracing is executed as follows -

if trigger A is matched, then trigger B is enabled;
if trigger B is matched, then trigger C is enabled;
and so on until the final breakpoint is matched.

Example 1: Execute a trigger trace using the single breakpoint of H5 set at (X1=0,X0=1).

```
>J0                              ;Reset the PC to 0000H.
>H0                              ;Clear H1-H6 settings.
>H5 01                           ;Break condition for X0=1, X1=0.
>H?                              ;Display H1-H6 settings.
H1=
H2=
H3=
H4=
H5= 01
H6=
>BTH5                            ;Execute the Breakpoint Trace with H5.
THE TRACE STOPS AT STEP 000D
>L
 FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
  0000  0000    0000    31   S      11110011
```

```
 0001          0001    F0   R      11110010
 0002          0002    7F   R      11110010
 0003  0003    0003    21   S      11110011
 0004          0004    34   R      11110010
 0005          0005    12   R      11110010
 0006  0006    0006    DB   S      11110011
 0007          0007    00   R      11110010
 0008          0000    00   I      11111110
 0009  0008    0008    00   S      11110011
 000A  0009    0009    00   S      11110011
 000B  000A    000A    D3   S      11110011
 000C          000B    00   R      11110010
 000D          0000    00   O      11111101  ;H5 matched.
>
```

Example 2: Execute an armed trigger trace with a single sequence of H5-->H6.

```
>J0                          ;Reset the PC to 0000H.
>H0                          ;Clear H1-H6 settings.
>H5 01                       ;Break condition for X0=1, X1=0.
>H6 00                       ;Break condition for X2=0, X3=0.
>H?
H1=
H2=
H3=
H4=
H5= 01
H6= 00
>BTH5 H6                     ;Execute the armed trigger trace.
THE TRACE STOPS AT STEP 000E
INITIAL TRIGGER TO BREAKPOINT INTERVAL--00:00:000:000
```

```
>L
 FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
  0000  0000    0000    31   S      11110011
  0001          0001    F0   R      11110010
  0002          0002    7F   R      11110010
  0003  0003    0003    21   S      11110011
  0004          0004    34   R      11110010
  0005          0005    12   R      11110010
  0006  0006    0006    DB   S      11110011
  0007          0007    00   R      11110010
  0008          0000    00   I      11111110
  0009  0008    0008    00   S      11110011
  000A  0009    0009    00   S      11110011
  000B  000A    000A    D3   S      11110011
  000C          000B    00   R      11110010
  000D          0000    00   O      11111101  ;H5 matched.
  000E  000C    000C    E5   S      11110011  ;H6 matched.
>
```

Example 3: Execute an armed trigger trace using a sequence of two triggers and a final breakpoint, i.e. H5-->H6-->H3.

```
>J0                          ;Reset the PC to 0000H.
>H0                          ;Clear H1-H6 settings.
>H3 XXXX 34 <0> W 2          ;Set H3 event count trigger.
>H5 01
>H6 00
>H?                          ;Display H1-H6 settings.
H1=
H2=
H3= XXXX 34 <0> W 2
H4=
H5= 01
```

```
H6= 00
>BTH5 H6 H3                    ;Execute the armed trace.
THE TRACE STOPS AT STEP 0011
INITIAL TRIGGER TO BREAKPOINT INTERVAL--00:00:000:002
>L
 FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
  0000  0000    0000   31   S      11110011
  0001          0001   F0   R      11110010
  0002          0002   7F   R      11110010
  0003  0003    0003   21   S      11110011
  0004          0004   34   R      11110010
  0005          0005   12   R      11110010
  0006  0006    0006   DB   S      11110011
  0007          0007   00   R      11110010
  0008          0000   00   I      11111110
  0009  0008    0008   00   S      11110011
  000A  0009    0009   00   S      11110011
  000B  000A    000A   D3   S      11110011
  000C          000B   00   R      11110010
  000D          0000   00   O      11111101  ;H5 matched.
  000E  000C    000C   E5   S      11110011  ;H6 matched.
  000F          7FEF   12   W      11110001
  0010          7FEE   34   W      11110001  ;H3 matched.
  0011  000D    000D   21   S      11110011
>
```

* Breakpoints H3 and H4 execute one additional cycle past the final trigger address. See timing diagram in Appendix C.

Example 4: Execute an armed trigger trace using a sequence of three triggers and a final breakpoint, i.e. H5-->H6-->H3-->H4.

```
>J0                                ;Reset the PC to 0000H.
>H0                                ;Display H1-H6 settings.
>H3 XXXX 34 <0> W 2
>H4 4321 FF <0> W 3
>H5 01
>H6 00
>H?
H1=
H2=
H3= XXXX 34 <0> W 2
H4= 4321 FF <0> W 3
H5= 01
H6= 00
>BTH5 H6 H3 H4                     ;Execute the armed trace.
THE TRACE STOPS AT STEP 001A
INITIAL TRIGGER TO BREAKPOINT INTERVAL--00:00:000:009
>L0 0 FFFF I O W                   ;List all INPUT/OUTPUT/WRITE
                                   ; cycles traced data.
 FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
  0008          0000    00   I      11111110
  000D          0000    00   O      11111101
  000F          7FEF    12   W      11110001
  0010          7FEE    34   W      11110001  ;H3 matched.
  0017          4321    06   W      11110001  ;H4 matched.
>
```

3.3 AND Trigger Trace - BT

```
BT(A B C)[ D]
BT(A B)[ C[ D]]
```

(A B C)[D] The AND trigger (breakpoints A, B and C) can be used as a single breakpoint construct, or can serve as a trigger construct for breakpoint D.

(A B)[C [D]] The AND trigger (breakpoints A and B) can be used as a single breakpoint construct, or can serve as a trigger construct for breakpoint C or for the armed trigger construct of C-->D.

Breakpoints A-D can be combined in a two or three level AND construct. Armed breakpoints can also be specified following the AND trigger. There are up to 38 different AND trigger combinations. Note that parantheses "( )" must be used in the syntax when inputting an AND trigger construct.

Example 1: Run a trace with an AND trigger used as a single breakpoint construct with H4, H5 and H6. (Note that breakpoints specified in an AND construct can be matched in any order.)

```
>J0                              ;Reset the PC to 0000H.
>H?                              ;Display H1-H6 settings.
H1=
H2=
H3= XXXX 34 <0> W 2
H4= 4321 FF <0> W 3
H5= 01
H6= 00
>BT(H5 H6 H4)                    ;Execute the trace with AND trigger.
THE TRACE STOPS AT STEP 001B
>L                               ;List all traced data.
 FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
  0000  0000    0000   31   S      11110011
  0001          0001   F0   R      11110010
  0002          0002   7F   R      11110010
  0003  0003    0003   21   S      11110011
  0004          0004   34   R      11110010
  0005          0005   12   R      11110010
  0006  0006    0006   DB   S      11110011
  0007          0007   00   R      11110010
  0008          0000   00   I      11111110
  0009  0008    0008   00   S      11110011
  000A  0009    0009   00   S      11110011
  000B  000A    000A   D3   S      11110011
  000C          000B   00   R      11110010
  000D          0000   00   O      11111101  ;H5 matched.
  000E  000C    000C   E5   S      11110011  ;H6 matched.
  000F          7FEF   12   W      11110001
  0010          7FEE   34   W      11110001
  0011  000D    000D   21   S      11110011
```

```
  0012            000E    21   R      11110010
  0013            000F    43   R      11110010
  0014  0010      0010    0E   S      11110011
  0015            0011    06   R      11110010
  0016  0012      0012    71   S      11110011
  0017            4321    06   W      11110001  ;H4 matched.
  0018  0013      0013    23   S      11110011
  0019  0014      0014    E3   S      11110011
  001A            7FEE    34   R      11110010
  001B            7FEF    12   R      11110010
>
```

* H5, H6 and H4 all matched; breakpoint H4 stops emulation at the next cycle.

Example 2: Perform a trace using an AND construct as the trigger for an armed breakpoint.

```
>J0                                ;Reset the PC to 0000H.
>H?                                ;Display H1-H6 settings.
H1=
H2=
H3= XXXX 34 <0> W 2
H4= 4321 FF <0> W 3
H5= 01
H6= 00
>BT(H4 H5 H6) H3                   ;Execute the logical trace.
THE TRACE STOPS AT STEP 001E
INITIAL TRIGGER TO BREAKPOINT INTERVAL--00:00:000:149
>L0 0 FFFF I O W
 FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
  0008          0000    00   I      11111110
  000D          0000    00   O      11111101  ;H5 matched.
  000F          7FEF    12   W      11110001  ;H6 matched.
```

```
  0010          7FEE    34   W       11110001  ;H3 matched.
  0017          4321    06   W       11110001  ;H4 matched.
  001C          7FEF    43   W       11110001
  001D          7FEE    22   W       11110001
>
```

Example 3: Run a trace with an AND trigger and an armed breakpoint sequence.

```
>J0                              ;Reset the PC to 0000H.
>H?                              ;Display H1-H6 settings.
H1=
H2=
H3= XXXX 34 <0> W 2
H4= 4321 FF <0> W 3
H5= 01
H6= 00
>BT(H6 H5) H4 H3                 ;Execute the logical trace.
THE TRACE STOPS AT STEP 001E
INITIAL TRIGGER TO BREAKPOINT INTERVAL--00:00:000:149
>L0 0 FFFF I O W
 FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
  0008          0000    00   I       11111110
  000D          0000    00   O       11111101  ;H5 matched.
  000F          7FEF    12   W       11110001  ;H6 matched.
  0010          7FEE    34   W       11110001
  0017          4321    06   W       11110001  ;H4 matched.
  001C          7FEF    43   W       11110001  ;H3 matched.
  001D          7FEE    22   W       11110001
>
```

3.4 OR Trigger Trace - BT

```
BT<A B C>
BT<A B>[ C[ D]]
```

`<A B C>` The OR trigger (breakpoints A, B or C) can be used as a single breakpoint construct.

`<A B>[C [D]]` The OR trigger (breakpoints A or B) can be used a single breakpoint construct, or can serve as a trigger for breakpoint C or for the armed trigger construct C-->D.

Breakpoints A-D can be combined in a two or three level OR construct. Armed breakpoints can also be specified after the OR trigger. If an armed trigger is used with the OR construct, the ORed pairs must be either data (H3,H4) or external breakpoints (H5,H6). There are up to 18 different OR trigger combinations. Note that angular brackets "< >" must be used in the syntax when inputting an OR trigger construct.

Example 1: Execute a trace using a two level OR construct.

```
>J0                          ;Reset the PC to 0000H.
>H?                          ;Display H1-H6 settings.
H1=
H2=
H3= XXXX 34 <0> W 2
H4= 4321 FF <0> W 3
H5= 01
H6= 00
>BT<H3 H5>                   ;Execute a trace with data breakpoint pair.
THE TRACE STOPS AT STEP 000D
>L
 FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
  0000  0000    0000    31   S      11110011
  0001          0001    F0   R      11110010
  0002          0002    7F   R      11110010
  0003  0003    0003    21   S      11110011
  0004          0004    34   R      11110010  ;H3 matched.
  0005          0005    12   R      11110010
  0006  0006    0006    DB   S      11110011
  0007          0007    00   R      11110010
  0008          0000    00   I      11111110
  0009  0008    0008    00   S      11110011
  000A  0009    0009    00   S      11110011
  000B  000A    000A    D3   S      11110011
  000C          000B    00   R      11110010
  000D          0000    00   O      11111101  ;H5 matched.
>
```

Example 2: Execute a trace using an OR construct as the trigger for an armed breakpoint.

```
>J0                              ;Reset the PC to 0000H.
>H?                              ;Display H1-H6 settings.
H1=
H2=
H3= XXXX 34 <0> W 2
H4= 4321 FF <0> W 3
H5= 01
H6= 00
>BT<H5 H6> H3                    ;Execute the trace.
THE TRACE STOPS AT STEP 0011
INITIAL TRIGGER TO BREAKPOINT INTERVAL--00:00:000:140
>L
 FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
  0000  0000    0000    31   S      11110011  ;H6 matched.
  0001          0001    F0   R      11110010
  0002          0002    7F   R      11110010
  0003  0003    0003    21   S      11110011
  0004          0004    34   R      11110010
  0005          0005    12   R      11110010
  0006  0006    0006    DB   S      11110011
  0007          0007    00   R      11110010
  0008          0000    00   I      11111110
  0009  0008    0008    00   S      11110011
  000A  0009    0009    00   S      11110011
  000B  000A    000A    D3   S      11110011
  000C          000B    00   R      11110010
  000D          0000    00   O      11111101
  000E  000C    000C    E5   S      11110011
  000F          7FEF    12   W      11110001
  0010          7FEE    34   W      11110001  ;H3 matched.
  0011  000D    000D    21   S      11110011
>
```

Example 3: Run a trace using an OR trigger and an armed breakpoint sequence.

```
>J0                                ;Reset the PC to 0000H.
>H?                                ;Display H1-H6 settings.
H1=
H2=
H3= XXXX 34 <0> W 2
H4= 4321 FF <0> W 3
H5= 01
H6= 00
>BT<H3 H4> H5  H6                 ;Execute and OR trigger and an armed
THE TRACE STOPS AT STEP 0011      ; breakpoint of H5→H6.
INITIAL TRIGGER TO BREAKPOINT INTERVAL--00:00:000:000
>L
 FRAME IFADDR ADDRESS DATA STATUS SPARE(8 BITS)
  0000  0000    0000    31   S      11110011
  0001          0001    F0   R      11110010
  0002          0002    7F   R      11110010
  0003  0003    0003    21   S      11110011
  0004          0004    34   R      11110010  ;H3 matched.
  0005          0005    12   R      11110010
  0006  0006    0006    DB   S      11110011
  0007          0007    00   R      11110010
  0008          0000    00   I      11111110
  0009  0008    0008    00   S      11110011
  000A  0009    0009    00   S      11110011
  000B  000A    000A    D3   S      11110011
  000C          000B    00   R      11110010
  000D          0000    00   O      11111101  ;H5 matched.
  000E  000C    000C    E5   S      11110011  ;H6 matched.
  000F          7FEF    12   W      11110001
  0010          7FEE    34   W      11110001
  0011  000D    000D    21   S      11110011
>
```

SECTION 4

SYNC SIGNAL OUTPUT

4.1 External Hardware Trigger Output - BS

BS 1|2[addx[data[<dbwc>][qualifier]]]

BS — is the command for External Hardware Trigger Output.

1 — specifies a positive signal pulse of 140~180ns. (Executing BS1 clears H3 because the same hardware circuit is used to support both of these breakpoints.)

2 — specifies a positive signal pulse that lasts for the duration of the trigger condition. (Executing BS2 clears H4 because the same hardware circuit is used to support both of these breakpoints.)

addx — is the trigger address that causes output of a sync signal. This can be an up to four digit hex address with an optional byte pair of "XX".

data — is an up to 2 digit hex data to be matched with the trigger address.

<dbwc> is an up to 2 digit hex setting indicating a "data bit wildcard". To enter any <dbwc>, a **data** value must first be defined. The data bit wildcard is interpreted as 8 binary data bits, where "0" indicates "don't care". The default value for the dbwc is FFH.

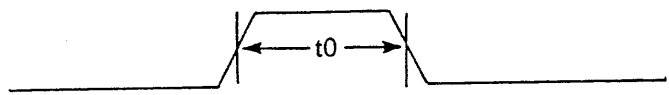
qualifier is a single or double alphabetic character indicating the type of processor activity to be matched with the trigger address. For a complete listing of the qualifiers applicable for a particular processor, refer to the appendices in the NEW MICE-II User's Manual.

The BS command can be used to trigger a sync signal to an external device (e.g. oscilloscope or logic analyzer). Connection is made on the BPP board at BSYNC1 (J3-1) or BSYNC2 (J3-2).

When the specified parameters are matched, a sync signal is transmitted. If no other breakpoints are specified before the BS command, emulation will continue after the trigger condition is met. To stop emulation before execution has been completed, input a Halt command.

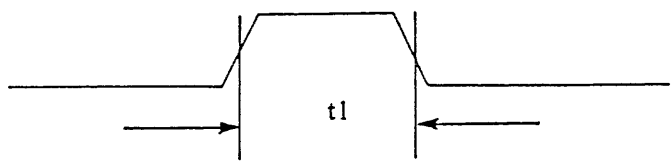
4.2 Output Signal Timing

1) BSYNC 1 Output Timing



When the BS1 setting is matched, the BPP will issue a positive signal pulse as indicated in the figure above. The pulse width of t0 = 140~180ns.

2) BSYNC 2 Output Timing



When the BS2 setting is matched, the BPP will issue a positive signal pulse as indicated in the figure above. The pulse width of t1 is determined by the following conditions:

a) No "wait state" inserted

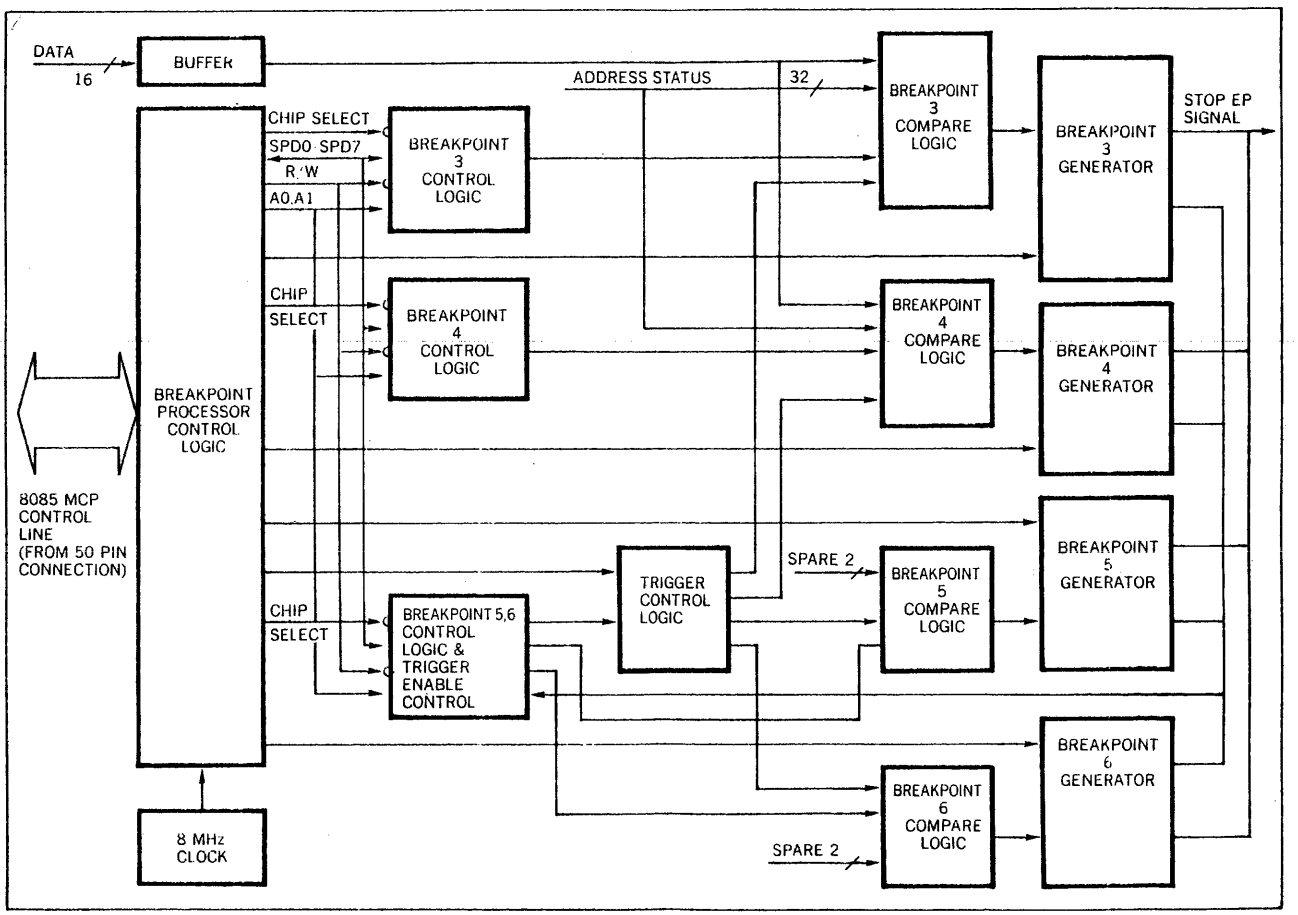
Pulse width of t1 = duration the trigger condition is matched on the bus (which depends on bus timing for the particular target system).

b) "Wait state" inserted

Pulse width of t1 = duration the trigger condition is valid on the bus, plus timing for wait state.

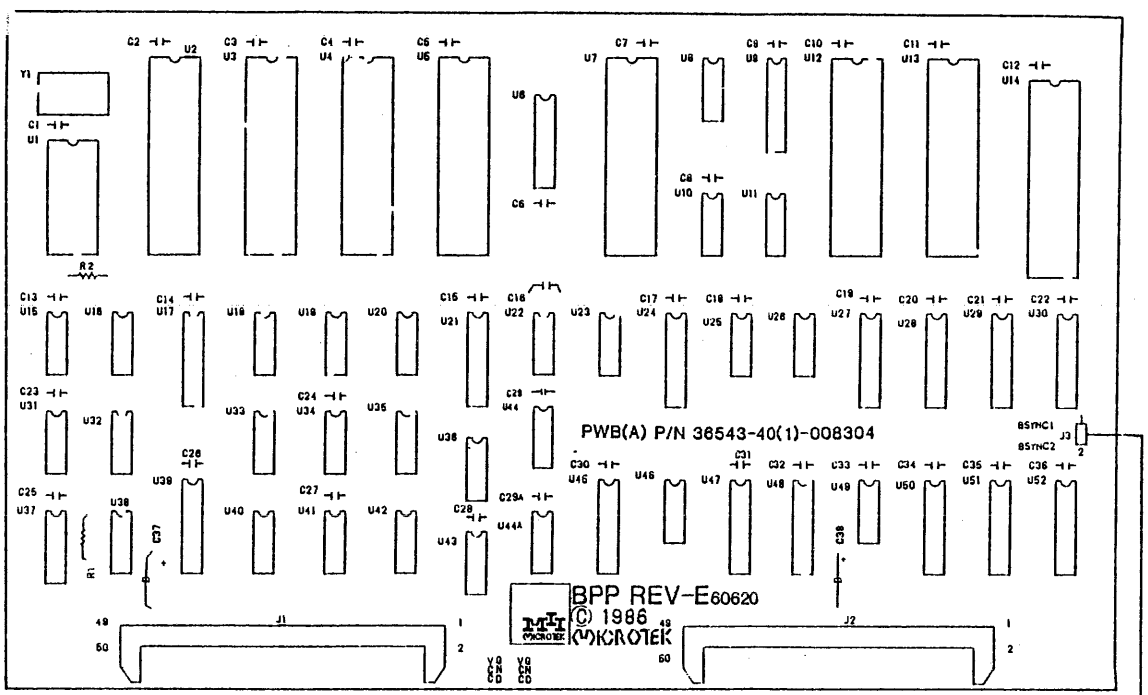
APPENDIX A

BPP BLOCK DIAGRAM



APPENDIX B

BPP PLACEMENT CHART



DWG NO.140104-213 SILK SCREEN

Connection For External
Hardware Trigger Output

APPENDIX C

TIMING FOR BREAK OPERATION (H1-H6)

C.1 Emulation Status

NEW MICE-II performs emulation in two different ways, depending on the type of processor.

1) Wait State

   The emulation processor bus is in wait state whenever it stops. In this state all signals on the bus are stable and can be tested with a logic probe or oscilloscope. Wait state is used in NEW MICE-II 8085 and 8048. In wait state, the bus indicates the current address for the 8085, and the next address for the 8048.

2) Run State

   Whenever user's program execution is stopped, the emulation processor is in free-run state, executing "SJMP $" instructions. In this state the signals at the Input/Output pins are stable and can be tested, but the signals on the Address/Data bus are changing dynamically and cannot be tested. Run state is used for NEW MICE-II 8031/8344, 8052/80C152 and 80515/80535.

The type of emulation performed determines timing for breakpoint operation. The following sections describe break operation for H1-H6. (The examples for wait state are based on the 8085 and those for run state are based on the 8031.)

C.2 Timing For Breakpoints H1/H2

1) Wait State

Emulation stops at stable bus state, before the current cycle has been completed. (The bus can be tested with a oscilloscope or logic probe at this time.) Entering a C command displays the current cycle address; but inputting an R command instead will display a different address if the status of the break cycle is other than S (instruction fetch). This is illustrated in the following program segment.

```
ADDRESS  DATA  STATUS   SOURCE CODE
   :       :      :          :
  1000     32     S      STA 3000      ;Break condition 1
  1001     00     R                    ;Break condition 2
  1002     30     R
  3000     AF     W
  1003     77     S      MOV M,A       ;A=0AF
   :       :      :          :
```

Note that no branch instruction is executed in the preceding example. In break condition 1, emulation is stopped at an instruction fetch cycle. Entering a C or R command displays address 1000H. In break condition 2, however, emulation is stopped at a cycle where bus status is other than instruction fetch. Entering a C command displays 1001H; but inputting an R command finishes the instruction and displays PC=1003H. (See figure C-1 under section C.5.)

2) Run State

When the break condition is matched, emulation stops at the end of the current instruction. Entering a C command will execute the next instruction and then display the current cycle status. However, entering an R command will display the current register content without causing any instructions to be executed. This is illustrated in the following example.

```
ADDRESS   DATA   STATUS   SOURCE CODE
   :        :        :          :
 0000       EE       S      MOV   A,R6
 0001       55       F
 0001       55       S      ANL   A,56      ;*
 0002       56       F                       ;*
 0003       F7       S      MOV   @R1,A
 0004       00       F      NOP
 0004       00       S      NOP
   :        :        :          :
```

* The specified break condition (H1 or H2) may be matched at any cycle of this instruction.

Entering a C command will execute the next instruction at address 0003H-0004H and then display the current cycle status; but entering the R command instead will simply display the current register content, where the PC=0003H. (See figure C-2, section C.5.)

C.3 Timing For Breakpoints H3/H4

The comparison operation for data breakpoints takes too long for emulation to be stopped at the current cycle. Also note that executing the C or R command after emulation has been stopped by H3 or H4 produces the same results as described above for H1/H2.

1) **Wait State**

   When the break condition is matched, emulation stops at stable bus state for the next cycle. (See figure C-1, section C.5.)

2) **Run State**

   When the break condition is matched, emulation is stopped at the end of the current instruction. (See figure C-3, section C.5.)

C.4 Timing For Breakpoints H5/H6

1) Wait State

When H5 or H6 is matched, a signal is immediately (appx. 100ns) generated by MICE to stop emulation at stable bus state. Therefore, if the trigger occurs "early enough" in a bus cycle (e.g. before T2 for 8085), emulation will stop at valid state for the current cycle; but if the trigger occurs "too late" (e.g. after T2 for 8085), then emulation will be stopped at valid state for the next cycle.

2) Run State

When H5 or H6 is matched, a signal is immediately (appx. 100ns) generated by MICE to stop emulation. Therefore, if the trigger occurs "early enough" (i.e. anywhere before the rising edge of ALE for 8031/8344, during the last cycle of an instruction), emulation will be stopped at the end of the current instruction; but if the trigger occurs "too late" (i.e. after the rising edge of ALE for 8031/8344, during the last cycle of an instruction), then emulation will be stopped at the end of the next instruction.

The break position for the emulation states described above is determined by the timing of H5/H6 and by the specific point in cycle execution that can be defined as "early enough" or "too late" (which depends on the type of emulation processor used, refer to section C.6).

C.5 Timing Diagrams For Break Operation

The following diagrams show the timing of break condition match and emulation halt during bus cycle execution.

1) Wait State

There are 4 bus cycles illustrated in the following diagram. (Note that this diagram is applicable for the Wait State example under section C.2.)

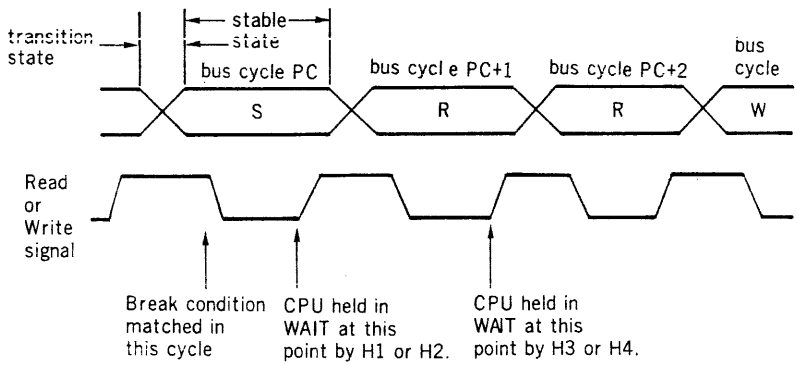


Figure C-1 Wait State Timing Diagram

S — is an opcode fetch cycle (i.e. the 1st byte of an instruction; no branch instruction is assumed in this example).

R — is a memory read cycle or the subsequent byte read for an instruction.

W	is a memory write cycle.
PC	is the program counter. (The increment per bus cycle is PC=PC+1; the PC is 1000 in this example.)
stable state	is a full bus cycle.
transition state	is end of a bus cycle and start of the next one.

2) Run State

a) H1/H2

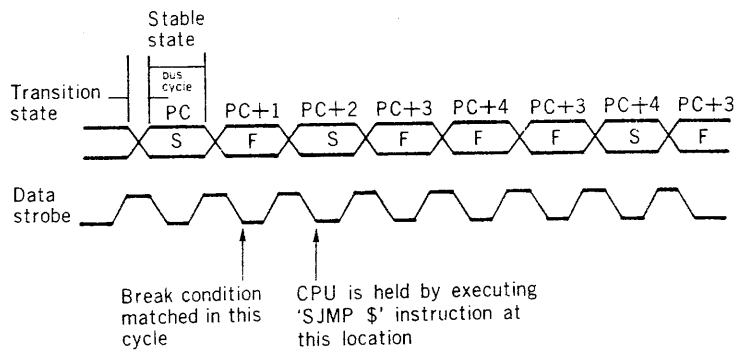


Figure C-2 Run State Timing Diagram for H1/H2

b) H3/H4

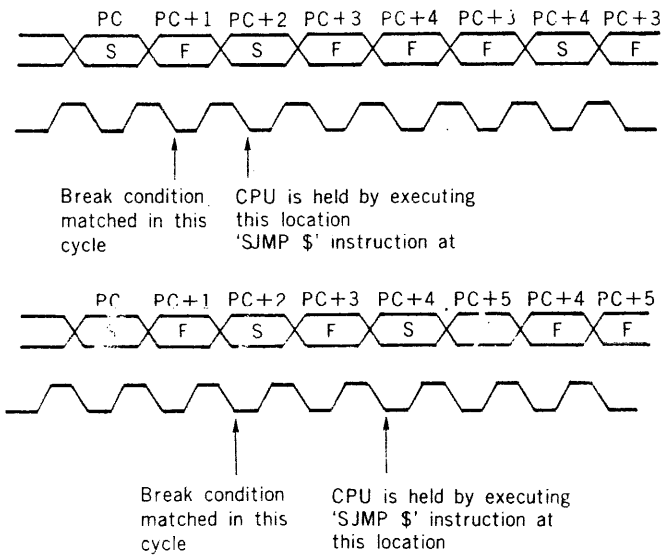


Figure C-3 Run State Timing Diagram for H3/H4

S is an opcode fetch cycle (i.e. 1st byte of an instruction; no branch instruction is assumed in this example).

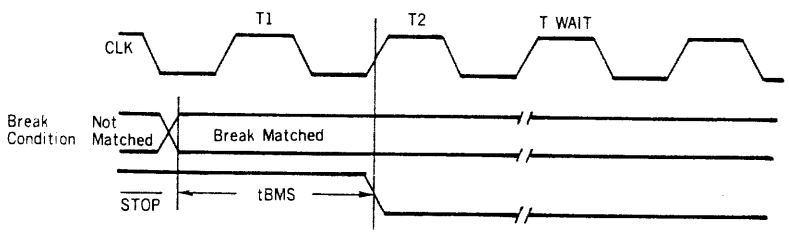
R is a memory read cycle for an instruction.

F	is a program memory read cycle for an instruction.
stable state	is a full bus cycle.
transition state	is end of a bus cycle and start of the next one.

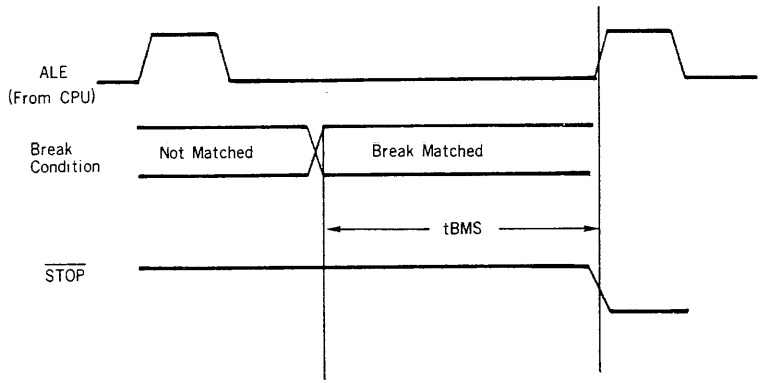
C.6 Breakpoint Timing (H5/H6) for Respective NEW MICE-II

NEW MICE-II	Break Match Timing Minimum Value (tBMS)	Stop CPU Setup Time Mirumin Value	Emulation Processor	MHz
8085	202ns + tRYS	tRYS = 110ns	8085A	6
		110ns	8085AH	6
		100ns	8085AH-2	10
8048	188.5ns		8048	11
8031/8344	215.4ns		80C31F	12
8052/80C152	217 ns		85C154VS/ 80C152JB-1	16
80515/80535	185ns + TWHLH		80515-E-A	12

1) 8085



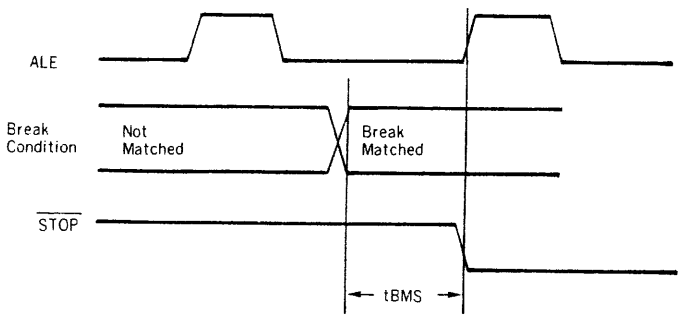
2) 8048



1-cycle instruction: If timing is within the specified parameters, emulation will stop at the next instruction cycle; otherwise the next instuction will be completed.

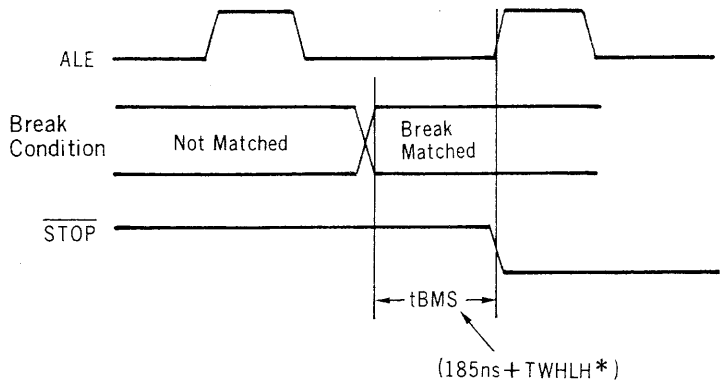
2-cycle instruction: If timing is within the specified parameters for either cycle, emulation will stop at the next instruction cycle; otherwise the next instruction will be completed.

3) 8031/8344



Note: Emulation will stop at the end of the current instruction, where $\overline{\text{STOP}}$ means that the CPU will be executing "SJMP $" instructions.

4) 80515/80535



* TWHLH is $\overline{WR}$ or $\overline{RD}$ to ALE high. (Refer to AC Characteristics, page 202, Seimens Microcomputer Components, SAB 80515/80535 Single-Chip Microcontroller User's Manual, 7/85).

Note: Emulation will stop at the end of the current instruction, where $\overline{STOP}$ means that the CPU will be executing "SJMP $" instructions.

SUEM SECTION

First Edition
March 1988

Microtek International, Inc.
6 Industry Road East 3
Science-Based Industrial Park
Hsinchu, Taiwan, 30077, R.O.C.
TEL: (035)772155
TLX: 32169 MICROTEK
FAX: (035)772598

C O N T E N T S

1.0 INTRODUCTION

This document provides reference information on the operation of the SUEM.

The user of this document is assumed be reasonably familiar with the NEW MICE-II emulator family.

2.0 SPECIFICATION

2.1 DESCRIPTION

The SUEM is a high performance, soft mapped emulation memory board, that replaces an HUEM in all NEW MICE-II/S units.

2.2 MEMORY SIZE

256 Kbytes

2.3 MAPPING RSOLUTION

Three modes:
1) 128 byte resolution (address range 0 to 128K)
   (64K code, 64K data for microcontrollers)
2) 1 Kbyte resolution (address range 0 to 1M)
3) 8 Kbyte resolution (address range 0 to 16M)

2.4 NON-VOLATILE STORAGE

The memory map settings can be stored in EEPROM. This setup is automatically recalled on power up.

2.5 INDICATIONS

The SUEM provides two LED indicators

2.5.1 Run Light

This shows whether the emulation processor is in a run state.

2.5.2 Power Light

This shows that 5 Volt supply is applied.

2.6 ACCESS VIOLATION

Outputs provided for address violation are:

* Internal (emulation memory) write access
* External (target memory) write access
* Guard memory (non-existent) access

2.7 VIOLATION ACCESS SETTINGS

The Violation Access Signals generated from SUEM are located at J3 (3-pin connector).

J3 Pin Definition

Pin 1 (J3-1)	Guard access violation
Pin 2 (J3-2)	Write protect violation
Pin 3 (J3-3)	Either guard access or write protect violation

The signals are active at high level.

To enable the violation to break the program execution, J3-1 and J3-2 should be wired (using jumper wires or others) to CEP spare bits X0 and X2 respectively. Then use BPP command to set the break conditions H5, H6.

Application notice for violation access break see appendix B.

3.0 SWITCH SETTINGS

3.1 DESCRIPTION:

SW1, an 8-way DIP switch, is located at the front of the SUEM and is accessible when the cover of the NEW MICE-II is removed. This switch configures the host interface (between host and MICE) and the inter board (between CEP and SUEM).

host baud rate	S1	S2	S3
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
19200	off	off	off

host mode	S4	S5	S6
8 data bits	off		
7 data bits	on		
parity enable		off	
parity disable		on	
even parity			off
odd parity			on

	SUEM DIP Switch		CEP DIP Switch					
inter comms*	S7	S8	S1	S2	S3	S4	S5	S6
9600	n/u**	on	on	on	on	off	on	n/u**
19200	n/u	off	off	off	off	off	on	n/u

* Notethat inter communication mode is 8 databits, no parity. On CEP, the DIP switch should be set to 19200 (or 9600) bauds, no parity and 8 data bits.

** not used

3.2 INSTALLATION

NOTE

This section is applicable only when replacing an HUEM with SUEM in MICE-II

a) Remove the existing HUEM board from NEW MICE-II and replace with the SUEM board.

b) Reconnect the two 50-pin connectors to SUEM.

c) Remove the Interface (DTE/DCE converter) header from the CEP board. Plug-in the CSCB (CEP SUEM Communication Board) cable header into the just vacated header socket.

4.0 COMMAND REFERENCE

4.1 HELP COMMAND

***?**

---

*? is the command for SUEM help.

This command provides the user with a command summary together with a syntax guide.

EXAMPLE:

```
>*?
              SUEM   V#.#
              -----------

DIAGNOSTIC TESTS        *D
HIGH SPEED DOWNLOAD     *L[OAD]
MEMORY MAP CONTROL      *M[AP]    [ADDR1 ADDR2 I|IR|E|ER|G]|[E|D]
MAPPING RESOLUTION      *R[ANGE]  [0|1|2]
RECALL MAP              *RE[CALL]
SAVE MAP                *S[AVE]
HELP                    *?
>
```

4.2 DIAGNOSTIC TESTS COMMAND

***D**

***D** is the command for SUEM diagnosis. Issuing this command will test emulation memory as well as SUEM hardware ciruits. Results are shown when diagnostic testing is completed.

After diagnostic testing, the origienal map settings, map resolution and contents in emulation memory are destroyed.

EXAMPLE:

```
>*D
Diagnostics: testing U12 & U26

Diagnostics: testing U13 & U27

Diagnostics: testing U14 & U28

Diagnostics: testing U15 & U29

Diagnostics: 256K ram installed
>
```

4.3 HIGH SPEED DOWNLOAD COMMAND

***LOAD**

***LOAD** is the command for high speed download.

After entering this command, the SUEM will accept an hex file sent via the serial port. An end of file record or an escape will terminate this mode and return the MICE prompt.

EXAMPLE:

>*LOAD <CR>

NOTE

The *LOAD command does not use the ACK/NAK protocol, but it does support hardware of software handshaking for flow control. The MICE CEP must have the ETX handshaking code installed in the firmware for this command to function.

4.4 MEMORY MAP CONTROL

***M[AP]**

***M[AP] start-address end-address memory-attrib**

***M[AP] global-controls**

This command has several options, allowing the memory map to be set, displayed or modified.

***M[AP]** is the command for memory map control. If no other parameters specified, inputting "MAP<CR>" displays the current memory map setting.

start-address is a a hexadecimal address indicating the start of a boundary as defined by the current mapping range setting. (also refer to the *RANGE command).

end-address is a hexadecimal address indicating the end of a boundary as defined by the current mapping range setting. (also refer to the *RANGE command).

memory-attrib can be one of the following:

I - internal (emulation) memory with read/write access

IR - internal (emulation) memory with read only access

E - external (target) memory with read/write access

ER - external (target) memory with read only access

G - guarded access

global-controls can be one of the following:

E - enable map settings

The current memory map settings will be used by the emulator.

D - disable map settings

The memory map settings are ignored and the emulator will use target system memory. The previously set memory map can be reenabled by typing *MAP E.

EXAMPLE:

```
>*MAP
Range: 0 to 16Mb with 8k resolution

000000 to FFFFFF     guarded              ( G)

>*MAP 0 1FFF I
>*MAP
Range: 0 to 16Mb with 8k resolution

000000 to 001FFF     internal read/write ( I)
002000 to FFFFFF     guarded              ( G)
```

```
>*MAP 100000 165FFF ER
>*MAP
Range: 0 to 16Mb with 8k resolution

000000 to 001FFF     internal read/write ( I)
002000 to 0FFFFF     guarded             ( G)
100000 to 165FFF     external read only  (ER)
166000 to FFFFFF     guarded             ( G)

>*MAP D
Map: emulation memory globally disabled
>*MAP 0 1FFF E
Map: emulation memory globally disabled
>*MAP E
Map: emulation memory is enabled
>

>*MAP
Range: 0 to 16Mb with 8k resolution

000000 to 001FFF     internal read/write  ( I)
002000 to 0FFFFF     guarded              ( G)
100000 to 165FFF     external read only   (ER)
166000 to FFFFFF     guarded              ( G)

>
```

NOTE

The SUEM halts the emulation processor when modifying the memory map, but it does not halt the emulation processor when displaying the memory map. It is up to the user to take the appropriate action when using this command.

4.5 MAPPING RESOLUTION CONTROL

*R[ANGE]

*R[ANGE] type

This command allows the user to display or select three different types of mapping resolution.

*R[ANGE] is the command for memory mapping resolution control. If no other parameters specified, inputting "*RANGE<CR>" displays the current mapping resolution.

type may be one of the following:

0 - 128 byte resolution, in the range 0 to 128K (17-bit addressing). For processors with separate address space for code and data (8031, 8052, 80515, Z8, ZS8), the SUEM will use the 17th address line of the memory map to signify data memory. This gives us 0 to 64K for code and 0 to 64K for data (mapped from 64K to 128K). Processors having 16 address lines will only be able to access the first 64K of emulation memory.

1 - 1K resolution in the range 0 to 1M bytes (20-bit addressing).

2 - 8K resolution in the range 0 to 16M bytes (24-bit addressing).

The emulation processor will be halted after this command is executed.

EXAMPLE:

```
>*RANGE
Rang: 0 to 16Mb with 8k resolution
>*RANGE 1
Range: 0 to 1Mb with 1k resolution
>*RANGE 0
Range: 0 to 128K with 128 byte resolution
```

NOTE

When the mapping resolution is changed, all the memory is mapped GUARDED.

If the emulation processor's memory range is 0 to 16M byte (24-bit addressing) and the user want to get a smaller mapping resolution, user can set a lower type of mapping resolution (0 or 1) with *RANGE command, but the memory will overlap.

EXAMPLE:

```
>*R 1
Range: 0 to 1Mb with 1k resolution
>*M 0 FFF I
>M FF0 FFF AAAA
>M FF0 FFF
       0000 0002 0004 0006 0008 000A 000C 000E     ASCII-CODE
000FF0 AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA  ................
>M 100FF0 100FFF
       0000 0002 0004 0006 0008 000A 000C 000E     ASCII-CODE
100FF0 AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA  ................
>
```

4.6 RECALL MEMORY MAP

*RECALL

This command allows the user to recall a previously saved memory map from the EEPROM. The current memory map will be destroyed.

EXAMPLE:

```
>*RECALL
>
```

NOTE

The emulation processor will be halted after executing this command.

4.7 SAVE MEMORY MAP

***S[AVE]**

SAVE is the command for saving memory map settings to EEPROM.

EXAMPLE:

```
>*SAVE
.............
>
```

NOTE

The *SAVE command will take 20 seconds (worst case) to execute.

APPENDIX A

LIMITED WARRANTY; Service

Microtek International Inc. (Microtek) warrants **Super Universal Emulation Memory or SUEM** (the Product) and the user's manual for the Product to be free from physical defects for a period of twelve (12) months from the date of the original retail purchase. This warranty applies only to the original retail purchaser who bought these Products from an authorized Microtek representative. This warranty is void if the Product is damaged by improper or abnormal use or by accident, if the Product is altered or modified in any way, or if any attempt is made to repair the Product without authorization from Microtek.

If you find a Product to be defective contact your authorized Microtek representative or Microtek. When you receive authorization to do so, return the Product as directed; include proof of purchase and purchase date. Microtek will correct physical defects in Products under warranty at no charge to you by repairing or, at its option, replacing such Products. THIS IS THE SOLE AND EXCLUSIVE REMEDY AVAILABLE FOR BREACH OF WARRANTY OR UNDER ANY OTHER LEGAL THEORY WITH RESPECT TO MICROTEK PRODUCTS. Product repairs not covered by warranty, and Product updates, are provided at a set rate.

ALL OTHER WARRANTIES AND REPRESENTATIONS, ORAL OR WRITTEN, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, ARE EXCLUDED AND DO NOT AP-

PLY. Except as set forth in this limited warranty, the purchaser assumes the risk as to these Products' quality, accuracy, design, and performance.

It is solely the purchaser's responsibility to determine the suitability of these Products for each particular application. Microtek Products are in all events not suitable, and are not authorized, for use in connection with life support devices or systems, or in other devices or systems potentially injurious to life or health.

IN NO EVENT WILL MICROTEK BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY BREACH OF WARRANTY OR UNDER ANY OTHER LEGAL THEORY, even if advised of the possibility of such damages. Microtek is thus not liable for lost profits or goodwill; downtime; damage or destruction of any program, data, equipment, or other property; costs of recovering, reprogramming, or reproducing any program, data, or equipment; personal injury or loss; or any other damages.

Except as required by law, no representative, agent, or employee of Microtek is authorized to commit Microtek to warranties, representations, or obligations inconsistent with or in addition to those set forth in this limited warranty.

Please complete and return the registration section of the warranty card (provided in the packing box) to Microtek.

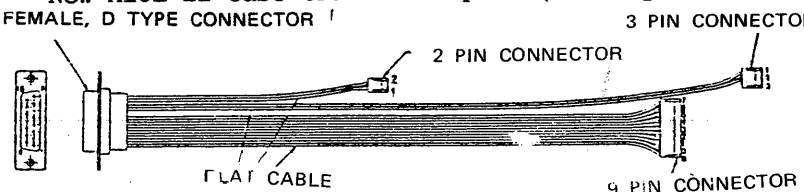
APPENDIX B

PROVIDING THE VIOLATION ACCESS BREAK

---

1. In order to provide a violation access break for New MICE-II/S, the trace cables have to be modified as follows:

   1) Connect the violation access signals (SUEM J3) to New MICE-II case trace bits port (see figure 1).



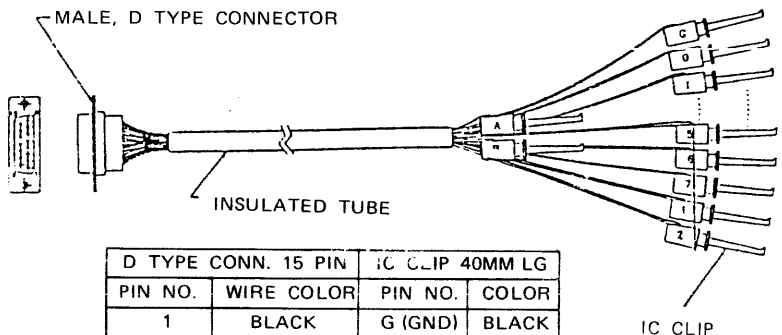
D TYPE CONN.	.098' CONN	WIRE COLOR
FEMALE 15 PIN	8 PIN & 2 PIN & 3 PIN	
PIN NO.	PIN NO.	
1	G (GND)	BLACK
9	0 (8 PIN)	BROWN
2	1 (8 PIN)	RED
10	2 (8 PIN)	ORANGE
3	3 (8 PIN)	YELLOW
11	4 (8 PIN)	GREEN
4	5 (8 PIN)	BLUE
12	6 (8 PIN)	PURPLE
5	7 (8 PIN)	GRAY
15	1 (2 PIN)	BLACK
8	2 (2 PIN)	BROWN
13	1 (3 PIN)	BLACK
6	2 (3 PIN)	BROWN

Figure 1

* 3-pin connector pin 1 (connecting to SUEM J3-1) connects to D type (female) connector pin 13.
* 3-pin connector pin 2 (connecting to SUEM J3-2) connects to D type (female) connector pin 6.

2) Connect two IC clips (marked A and B with shorter wire) to D type (male) connector pin 13 and 6 on external trace cable (see figure 2).

* IC clip A connect to D type (male) connector pin 13.
* IC clip B connect to D type (male) connector pin 6.



D TYPE CONN. 15 PIN		IC CLIP 40MM LG	
PIN NO.	WIRE COLOR	PIN NO.	COLOR
1	BLACK	G (GND)	BLACK
9	BROWN	O (8 PIN)	RED
2	RED	1 (8 PIN)	RED
10	ORANGE	2 (8 PIN)	RED
3	YELLOW	3 (8 PIN)	RED
11	CREEN	4 (8 PIN)	RED
4	BULE	5 (8 PIN)	RED
12	PURPLE	6 (8 PIN)	RED
5	GRAY	7 (8 PIN)	RED
15	BLACK	1 (2 PIN)	BLACK
8	BROWN	2 (2 PIN)	BLACK
13	BLACK	A (2 PIN)	RED
6	BROWN	B (2 PIN)	RED

Figure 2

2. If user wants to use the violation access break feature, follow the procedure listed below:

   1) Connect the red IC clip 0 (trace bit X0) to IC clip A.
   2) Connect the red IC clip 2 (trace bit X2) to IC clip B.
   3) Set the BPP external hardware breakpoint H5 and H6 as follows:

```
> H5 X1              ;for guard access violation
> H6 X1              ;for write protect violation
```

   If an attempt is made to write into Read-only area (defined as "IR" or "ER" in the MAP command) while emulation processor running, this message will display -

"PROGRAM BROKE AT BREAKPOINT-6"

   Likewise, if a non-existent area (defined as "G" in MAP command) is accessed, this message will display -

"PROGRAM BROKE AT BREAKPOINT-5"

Notes: 1. Only the delivery of New MICE-II/S or/H includes these two trace cables (refer to fig. 1 and 2), SUEM single package does not.

2. Only New MICE-II/S provides the violation access break.

3. The PCB revision (Rev-B) of SUEM will move the position of J3 connector near the U63. This modification will shorten the wire length between the J3 and D type (female) connector.