MILLENNIUM

MicroSystem Emulator Users Manual

Leadership in Microprocessor Instrumentation

WARNING:

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. As temporarily permitted by regulation, it has not been tested for devices pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

The following procedures may help to alleviate the Radio or Television Interference Problems.

1. Reorient the antenna of the receiver receiving the interference.

2. Relocate the equipment causing the interference with respect to the receiver (move or change relative position).

3. Reconnect the equipment causing the interference into a different outlet so the receiver and the equipment are connected to different branch circuits.

4. Remove the equipment from the power source.

NOTE:

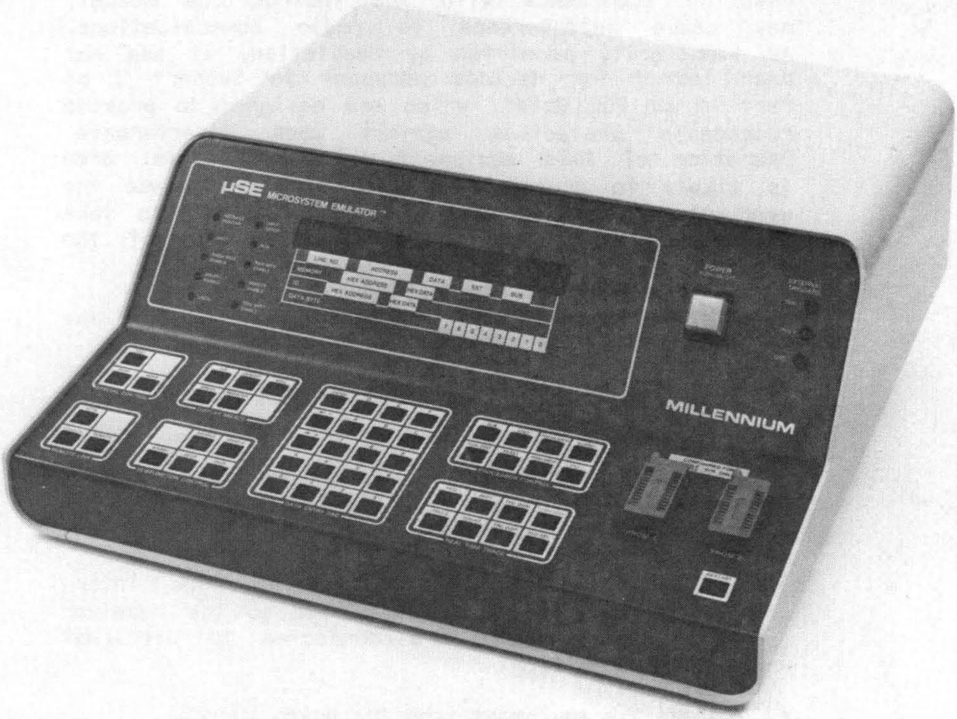
The user may find the following booklet prepared by the FCC helpful: "How to Identify and Resolve Radio-TV Interference Problems". This booklet is available from the U.S. Printing Office, Washington, D.C. 20402. Stock # 004-000-00345-4.

Publication Number 87000040
Release 3.0
March 1981
$25.00

MICROSYSTEM EMULATOR

USERS MANUAL

Millennium Systems, Inc.
19050 Pruneridge Avenue
Cupertino, CA 95014
Telephone: (408) 996-9109
TWX/TELEX # 910-338-0256



MicroSystem Emulator

This users manual provides detailed information on the operation of Millennium's MicroSystem Emulator. The MicroSystem Emulator (uSE) is designed to support a variety of different microprocessors including types that will become available in the future.

Millennium assumes no responsibility for inaccuracies in this publication. The material in this publication is subject to change without notice.

Copies of this publication and other Millennium publications may be obtained from the Millennium sales office or distributor serving in your locality.

MANUAL OVERVIEW

Chapter 1, INTRODUCTION, is an overview of the uSE.

Chapter 2, INSTALLATION, gives the installation and hardware interfaces of the uSE.

Chapter 3, KEYBOARD OPERATION, describes the functions of the keyboard in detail.

Chapter 4, COMMUNICATIONS, explains the RS-232 interfaces of the uSE.

Chapter 5, REAL-TIME TRACE, gives a detailed description of the real-time trace feature of the uSE.

Chapter 6, PROGRAMING, explains how to program diagnostic PROMs for the uSE.

Chapter 7, OPERATION, contains cookbook examples of how to use the uSE.

ASSISTANCE

If you require any assistance on this product, please call Customer Service on the toll-free hot-line numbers listed below:

National	(800) 538-9320/9321
California	(800) 662-9231

RELATED PUBLICATIONS

Manuals

MicroSystem Emulator 6800/2 Emulator Option Manual

MicroSystem Emulator 6801/3 Emulator Option Manual

MicroSystem Emulator 8080 Emulator Option Manual

MicroSystem Emulator Z80A Emulator Option Manual

MicroSystem Emulator 8085 Emulator Option Manual

MicroSystem Emulator 8048 Emulator Option Manual

Application Notes

Programming with uSA Microsystem Analyzer

Diagnostic Programming for Microprocessor-based Systems

Guide to Testing Microprocessor-based Systems and Boards

Miscellaneous Publications

RS-232 Specifications

INTRODUCTION

PRODUCT DESCRIPTION

The MicroSystem Emulator (uSE) shown in Figure 1-1 is a stand-alone, in-circuit emulator used for developing microprocessor-based products. The uSE provides low-cost, universal hardware support for many of the most popular microprocessors and microcomputers.

Operating either in a stand-alone mode or with a host computer system, the uSE offers the most comprehensive in-circuit emulation technology available today for 8-bit microprocessors. The uSE can even be used in systems that feature full speed operation of the target microprocesor and systems that use DMA techniques for transferring data.

The user can enter small programs from the keyboard, download larger programs from a host computer, and debug the programs; either in the memory of the uSE or in the memory of the Unit Under Test (UUT). The functional keyboard speeds the debugging process by reducing many of the common operations used in debugging to a single keystroke.

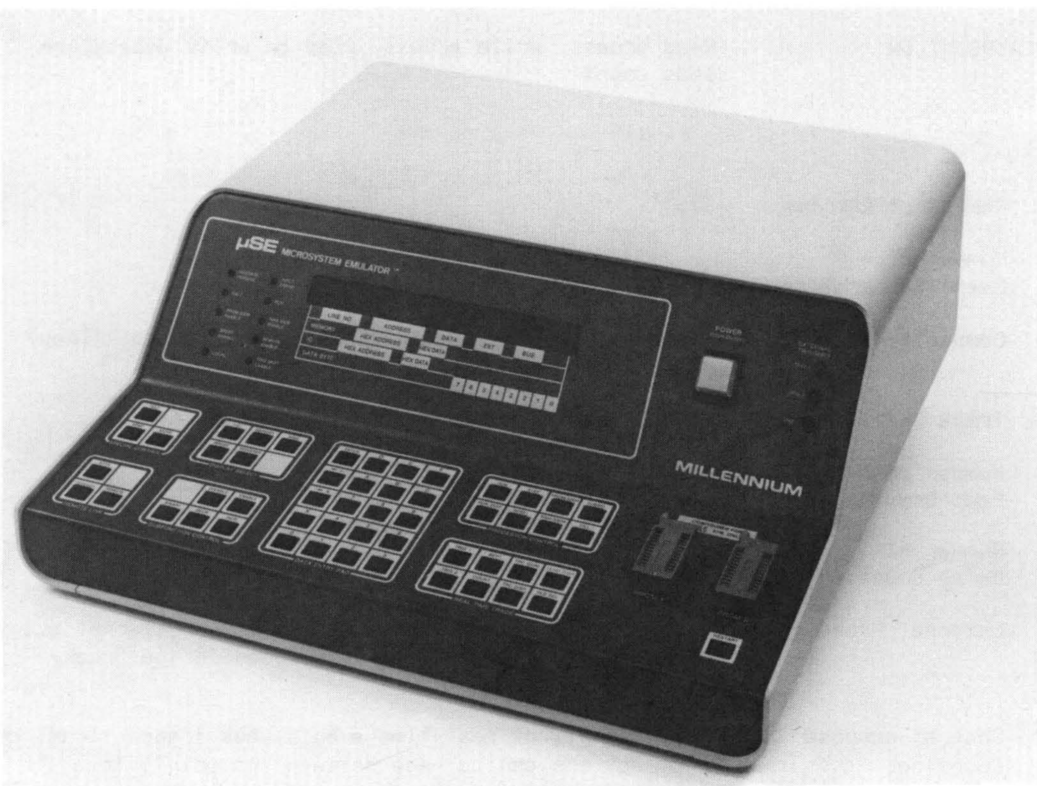


Figure 1-1. MicroSystem Emulator (uSE)

SPECIFICATIONS

PROGRAMMERS PANEL

20-character, 16-segment, alphanumeric display
Display Interpreter
Ten LED Indicators
Two External Triggers for Real-Time Trace
52-key data entry and command keyboard
Two zero-insertion-force sockets for 2708 or 2716 EPROMs

INTERNAL MEMORY

Comm/RAM board:	8K, 150ns system access time, static RAM, mappable in 1, 2, 4, or 8K blocks through the memory space
Expansion RAM board:	Same as the Comm/RAM board

BREAKPOINT ON:	Read access, write access, read or write access, or pass count

REAL-TIME TRACE

Number of Entries:	128
Number of Event Comparators:	2
Comparisons made on:	16-Address lines, 8-data lines, 3-bus control lines, and 8-external data bits
Trace Qualifiers:	Fetch, memory read/write, and I/O read/write
Number of Pass Counts:	Up to 65,534
Number of Delay Counts:	Up to 65,534 bus clocks, must be at least 3.
External Probes:	Eight input probes can be used to store external event data and can be selectively included in the breakpoint equation.
General-purpose Counting:	Measurements of real-time events, bus transactions, or instructions can be made between two points in a program or between events.

SPECIFICATIONS (continued)

COMMUNICATIONS INTERFACE (COMM LINK PORT)

Connector:	RS232C D type, 25-pin female
Inputs:	Rx data, CTS, DCD
Outputs:	Tx data, RTS
Data Transfer:	Block mode - hexadecimal format (compatible with Tektronix 8002 Development System)
Levels:	RS-232C
Data Rates:	110, 300, 600, 1200, 2400, 4800, 9600 baud.
Control:	Integral command interpreter handles interaction with host computer.

DISPLAY TERMINAL INTERFACE (TERMINAL PORT)

Connector:	RS232C D type, 25-pin female
Inputs:	Rx data
Outputs:	Tx data
Data Transfer:	Serial Transfer - ASCII Format
Levels:	RS-232C
Data Rates:	110, 300, 600, 1200, 2400, 4800, 9600 baud (must be less than or equal to the baud rate of the COMM LINK Port).
Control:	Integral command interpreter handles interaction with terminal.

POWER

100, 120, 220, and 240 VAC, +5%, -10%
50/60 Hz, 120 watts max.

ENVIRONMENTAL

Operating Temperature:	0°C to 40°C (32°F to 104°F)
Storage Temperature:	-40°C to 75°C (-40°F to 167°F)
Humidity:	95% RH 15°C to 40°C (59°F to 104°F)

PHYSICAL

Width:	17.5"
Height:	6.5"
Depth:	18.0"
Weight:	24 lbs

PHYSICAL DESCRIPTION

Front Panel

Figure 1-2 shows a front view of the uSE.

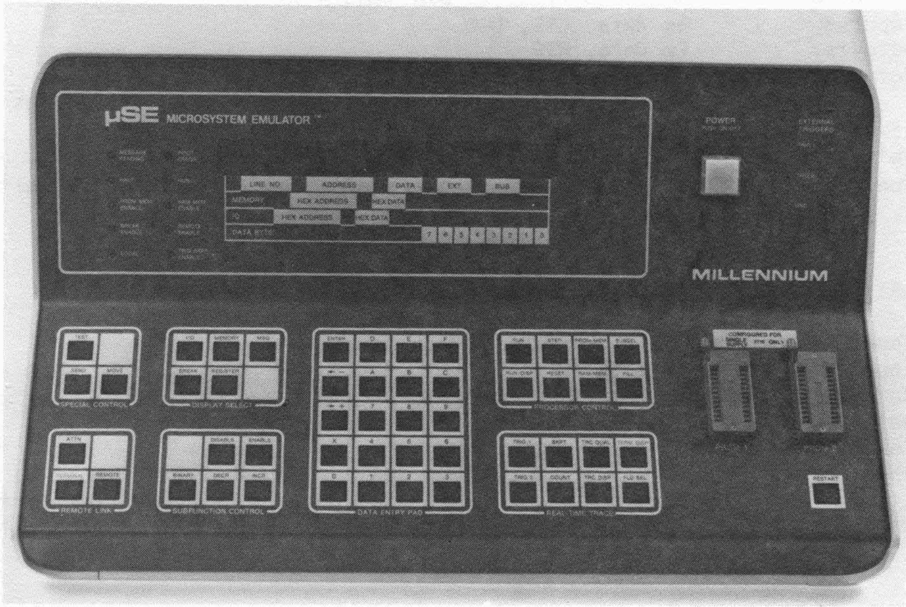


Figure 1-2. uSE Front View

The upper front panel contains:

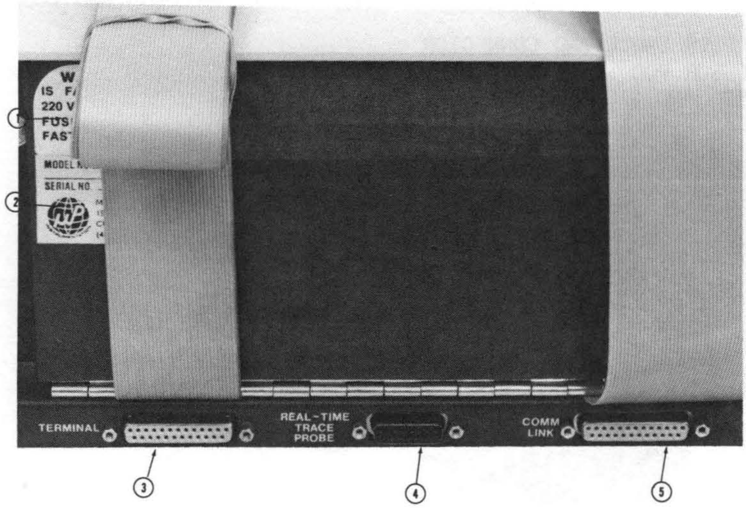
1 LED STATUS INDICATORS
2 20-CHARACTER ALPHANUMERIC DISPLAY
3 DISPLAY FIELD INTERPRETER
4 EXTERNAL TRIGGERS FOR REAL-TIME TRACE
5 POWER ON/OFF SWITCH

The lower front panel (keyboard) of the uSE contains:

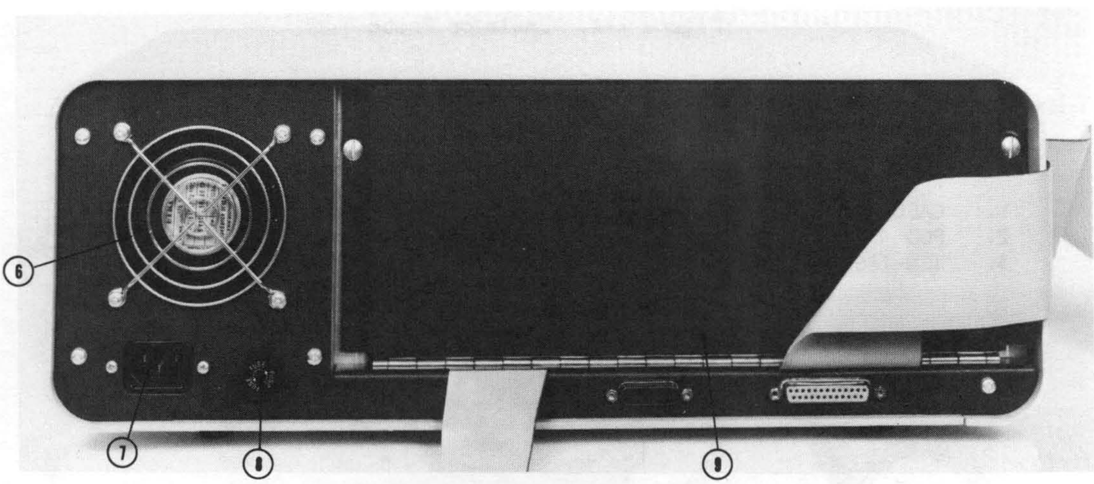
6 SPECIAL CONTROL KEYS
7 DISPLAY SELECT KEYS
8 DATA ENTRY KEYPAD
9 PROCESSOR CONTROL KEYS
10 REMOTE LINK KEYS
11 SUBFUNCTION CONTROL KEYS
12 TRACE CONTROL KEYS
13 RESTART KEY
14 TWO PROM SOCKETS

Rear Panel

Figure 1-3 shows a rear view of the uSE.



View A. Connectors and Labels



View B. Chassis Components

The rear panel of the uSE contains:

1 WARNING LABEL
2 MODEL NUMBER/SERIAL NUMBER LABEL
3 DISPLAY TERMINAL RS-232 CABLE CONNECTOR
4 REAL-TIME TRACE PROBE CONNECTOR
5 COMMUNICATIONS LINK RS-232 CABLE CONNECTOR
6 FAN
7 POWER CORD PLUG
8 AC POWER FUSE
9 CARD CAGE ACCESS DOOR

Figure 1-3. uSE Rear View

Emulator Probe

The In-Circuit Emulator probe, shown in Figure 1-4, consists of:

1. PROCESSOR CABLE AND CONNECTOR
2. PROCESSOR POD
3. CONTROL AND SIGNAL CABLES

Figure 1-4. Emulator Probe

Real-Time Trace Probe

The Real-Time Trace Probe, shown in figure 1-5, consists of:

1. CABLE
2. POD
3. TEN-TEST CLIPS

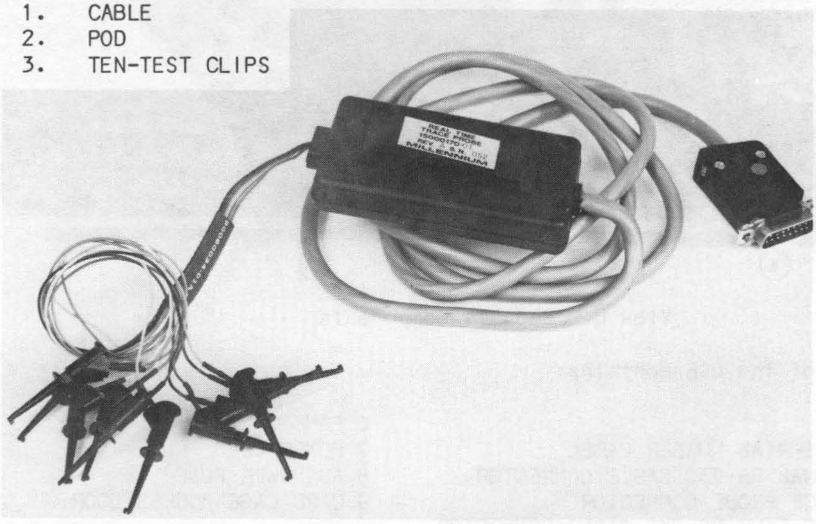


Figure 1-5. Real-Time Trace Probe

OPERATIONAL FEATURES

In-Circuit Emulation

In-Circuit Emulation is a technique that uses an external device to imitate the functions of the microprocessor in the Unit Under Test (UUT). This technique is implemented in the uSE by using a probe, connected into the microprocessor socket of the UUT, to connect the uSE to the UUT. The microprocessor in the uSE is an exact copy of the microprocessor it replaces in the UUT (see figure 1-6).

By using in-circuit emulation, the uSE can operate the UUT, using the programs and memory of either the UUT or the uSE. In addition, the uSE can set hardware breakpoints, step through program instructions, examine, display and alter CPU registers, memory and I/O values. In this way, in-circuit emulation allows the user to monitor and control a microprocessor-based product and detect faulty hardware and software logic.

In-circuit emulation can operate without built-in test points, and requires only that a clock in the UUT be functional.

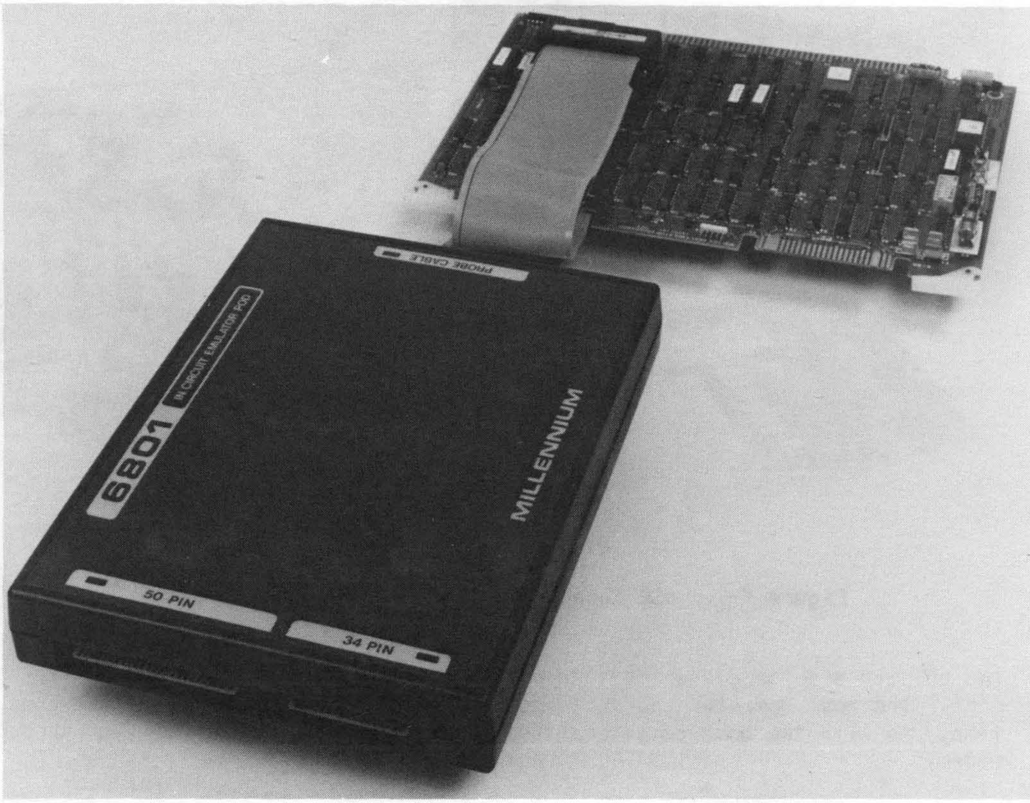


Figure 1-6. Emulator Probe connected to the UUT Microprocessor Socket

Communications

Comm Link Port

The uSE connects to a host computer system via an RS-232C EIA interface (figure 1-7). The connection is made through the COMM LINK port at the rear of the uSE (see figure 1-3) with a 25-pin subminiature D type female connector. The serial I/O port of the host computer should supply a cable and a male 25-pin subminiature D type connector.

The uSE may be configured for direct connection to a host computer or it may be configured for connection through a modem.

Figure 1-7. uSE Connected Directly to a Host Computer

The uSE may also be connected to another uSE through the COMM LINK serial I/O port. The host computer and the second uSE may not be connected at the same time. As with the host computer, the uSE may be connected directly or through a modem.

See chapter 2, INSTALLATION, for detailed information on interfacing through the COMM LINK serial I/O port.

TERMINAL Port

The uSE connects to a display terminal via an RS-232C EIA interface (figure 1-8). This connection is made through the TERMINAL port at the rear of the uSE (see figure 1-3) with a 25-pin subminiature D type female connector. The serial I/O port of the terminal should supply a cable and a male, 25-pin subminiature D type connector.

Figure 1-8. uSE Connected to a Display Terminal

The uSE may only be configured for direct connection to the display terminal.

Both a host computer (or a second uSE) and a display terminal may be connected to the uSE at the same time. With a display terminal connected, and the host computer's input/output assigned to the remote input and output ports, the display terminal keyboard can be used to communicate with the host computer via the uSE. The uSE is transpanent in this mode.

See chapter 2, INSTALLATION, for detailed information on interfacing through the TERMINAL serial I/O port.

Real-Time Trace

The Real-Time Trace feature of the uSE allows the user to take a snapshot dump of 128 selected cycles of his system while it is operating at full speed. The Trace Control keys are explained briefly below:

Key	Description
TRC DISP	Displays the contents of the trace buffer.
TRC QUAL	Selects the type of bus transactions stored in the trace buffer.
TRIG 1	Specifies address, data, control, external data, pass counts, and delay counts for trigger 1.
TRIG 2	Specifies address, data, control, external data, pass counts, and delay counts for trigger 2.
BKPT	Specifies a breakpoint on trigger 1 or trigger 2.
COUNT	Counts events such as fetches, microseconds, and milliseconds.
FLD SEL	Selects options and modes for each trace command key.

The bus history of address and data information is stored in the memory of the uSE for the operator to examine. A pass counter and delay counter can be used to start tracing after a number of passes through an iterative program loop.

Trace has two event detectors in addition to the standard breakpoint (BREAK key). These can be used to send trigger signals to oscilloscopes or logic analyzers through the two front panel probe points (see figure 1-2). The event trigger can be defined to occur in several different locations to locate difficult problems. In addition, up to eight external logic signals can be captured in the trace compare by using the external data probes. For details on the Real-Time Trace, see chapter 5.

INSTALLATION

UNPACKING, INSPECTION, AND SERVICE

Upon receiving the uSE, inspect the instrument and accessories for physical damage. If damage is evident, do not operate the instrument. Instead, notify the carrier and Millennium Systems at once. Millennium will arrange for repair or replacement without waiting for settlement of the claim against the carrier.

If the instrument is to be returned to Millennium, attach a tag showing owner, address, part number, and a description of the failure. If available, the original shipping carton and packing material should be re-used with a Returned Material Authorization (RMA) number prominently displayed on the carton. An RMA number can be obtained by calling Customer Service at Millennium.

Unless notified to the contrary, any claims for operations, assistance and/or service will be provided by Millennium from its plant in Cupertino, California. Should assistance be required, call Customer Service at:

National	(800) 538-9320/9321
California	(800) 662-9231

ACCESSORIES

The following standard accessories are included with each uSE:

ONE EMULATOR BOARD
ONE EMULATOR POD WITH EMULATOR PROBE AND 40-PIN AUGAT HEADER
ONE REAL-TIME TRACE PROBE
ONE DOCUMENTATION PACKAGE

OPTIONS

The following option is available for the uSE:

EXTENDED 8K RAM BOARD

POWER REQUIREMENTS

The standard uSE operates from an AC line supplying of 120v (+5%, -10%) at 50/60Hz with 120 watts of power, maximum. The 3-wire power cable outlet, grounds the instrument (see figure 1-2). To preserve this safety feature when operating from an outlet without a ground connection, use an appropriate adapter and connect the ground lead to an external ground.

CIRCUIT PROTECTION

The fuse is located on the left side of the rear panel beside the power cord connector (see figure 1-3). Table 2-1 gives the value and types of fuses required for various input voltages.

Table 2-1. Fuse Requirements

INPUT VOLTAGE	FUSE
100/120 VAC	3A, 3AG
220/240 VAC	11/2 A, 3AG

USER PRECAUTIONS

Applying Power

Before applying power to the instrument, check the warning label located out the rear of the instrument (see figure 2-1). Ensure that the ac line output coincides with the power requirements.

WARNING - THIS UNIT IS FACTORY WIRED FOR 240 VAC 50/60 HZ. REPLACE FUSE WITH 3AG 2 AMP FAST BLOW ONLY.

Figure 2-1. Warning Label

Inserting PROMs

Before inserting a PROM into one of the sockets on the front panel, ensure that the instrument will accept that PROM. The instrument will accept either 2708 PROMs or 2716 PROMs, but not both.

Connecting/Disconnecting the Emulator

When connecting or disconnecting the emulator probe, power must be turned off in both the UUT and the uSE. Otherwise, both instruments may be permanently damaged.

Ensure that pin 1 of the emulator probe is aligned with pin 1 of the target microprocessor on the UUT.

CARD CAGE CONFIGURATION

The card cage of the uSE is universal, so any uSE board can be placed in any slot of the card cage. The uSE is shipped configured as shown in figure 2-2. Table 2-2 identifies the boards and cable interface connection points.

```
                     uSE Rear View

                                   DISPLAY
      [=========================== 26 ===]    <--- Control Processor Board

      [=== 26 ===========================]    <--- Debug Board
                 PROBE
      [================= 5  26 ========]      <--- Real-Time Trace Board

      [                                  ]    <--- Emulator Board *
           N/A         CRT        COMM
      [=== 26 ========== 26 ====== 26 ===]    <--- Comm/RAM Board
           N/A         N/A        N/A
      [=== 26 ========== 26 ====== 26 ===]    <--- Extension RAM Board
                                                   (no cables used)

                                                 * Shipped separately.
```

NOTE: The colored traces should all be on the right as you face the back of the uSE.

Figure 2-2. Board Configuration as Shipped

Table 2-2. Cable Identification

Board	Number of Cables/ board	Connector Size	Identification Label
Control Processor	1	26	CTRL PROC P3
Debug	0	-	
Emulator	1 or 2	34, 50	none, SLAVE P2
Real-Time Trace	2	5, 26	none, none
Comm/RAM	2	26,26	COMM/RAM P2,COMM/RAM P3
Extension RAM	0	-	

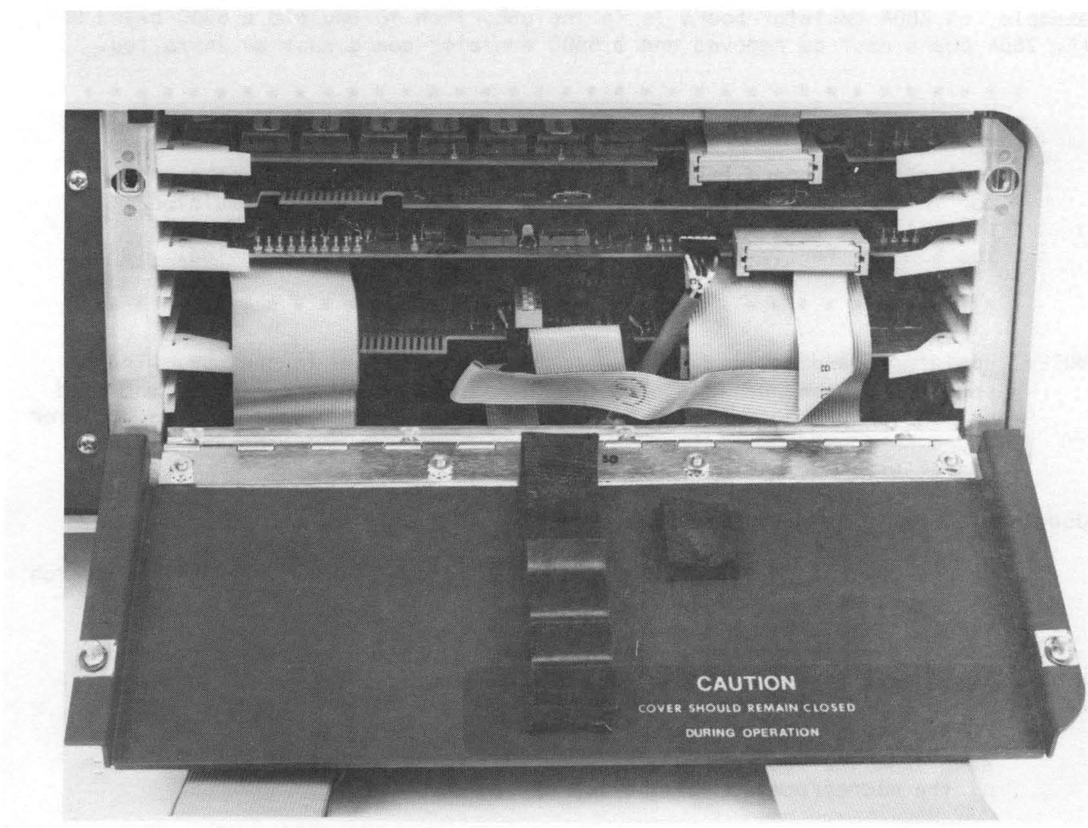


Figure 2-3. uSE Internal View

CHANGING THE EMULATOR BOARD

Before any kind of system testing is performed, an emulator board designed for the specific microprocessor (6800, Z80, etc.) your UUT uses must be installed in the card cage of the uSE. Only one emulator board may be installed in the card cage at a time. Therefore, if the emulator board in the uSE is not the one required to emulate your UUT, removed it and install the correct board. If, for example, an Z80A emulator board is in the uSE, then to emulate a 6800-based UUT, the Z80A board must be removed and a 6800 emulator board must be installed.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                            W A R N I N G                                *
*                                                                         *
* Each emulator board has an associated emulator pod that works only     *
* for the specific type of emulator.  For example, a 6800 emulator       *
* pod will not work with an 8080 emulator board and it may damage        *
* the board.  Verify that the correct emulator pod is installed.         *
*                                                                         *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

NOTE: Emulator board options are never shipped installed in the uSE. The boards are packaged in a separate plastic shipping wrapper and enclosed within the uSE shipping container. Procedures for installing an emulator board in a uSE are similar to the procedures for changing the emulator board, with the exclusion of steps 3, 4, 5, and 6.

Use the following step-by-step procedure when changing emulator boards.

1. Remove power from the uSE by pressing the POWER ON/OFF switch (the switch is illuminated when power is on).
2. Loosen the two retaining screws on both sides of the card cage access panel (see figure 1-3) and open the panel.
3. Locate the emulator board in the card cage. It is in the fourth slot down from the top of the card cage. It can be identified by the name of the microprocessor at the top of the board.
4. Disconnect the ribbon cable connector(s) connected to the emulator board.
5. Using both hands, grasp the release tabs on each side of the emulator board and simultaneously pull up on the tabs so that the board is pulled out of its connector.
6. Remove the board and store it.
7. Insert the new emulator board into the same slot of the card cage and simultaneously push it down with both hands until the board is fully seated.
8. Connect the ribbon cable connector(s) onto the mating portion at the top of the emulator board. (Refer to the individual emulator manuals for installation illustrations relative to specific emulator boards.)
9. Close the back panel and tighten the two retaining screws. Heat build-up due to improper air flow may result if the back panel is not replaced.

CONNECTING THE uSE TO THE UUT

The uSE is connected to the UUT by first removing the microprocessor in the UUT from it's socket and connecting the emulator probe into the now empty socket. Before applying power ensure that pin 1 of the emulator probe is aligned with pin 1 of the microprocessor socket.

When power is applied to both systems, the uSE will be in control of the UUT.

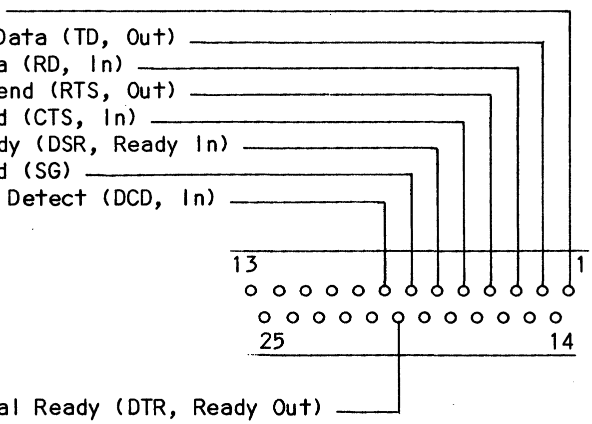
SERIAL PORTS

The uSE can be connected to a host computer or another uSE, through the COMM LINK serial port; and to a display terminal through the TERMINAL serial port (refer to figure 1-3 for the locations of the connectors). The COMM LINK and TERMINAL ports on the uSE both use standard RS-232, "D" type, 25-pin female connector. All voltage levels are standard RS-232 levels (refer to RS-232 specifications for signals definitions). The REMOTE LINK keys (see chapter 3) are used to set the parameters and control the operation of these two ports.

Comm Link Port

Figure 2-4 shows the pin functions of the COMM LINK Serial Port.

Pin 1 Frame Ground
Pin 2 Transmitted Data (TD, Out)
Pin 3 Received Data (RD, In)
Pin 4 Request to Send (RTS, Out)
Pin 5 Clear to Send (CTS, In)
Pin 6 Data Set Ready (DSR, Ready In)
Pin 7 Signal Ground (SG)
Pin 8 Data Carrier Detect (DCD, In)



Pin 20 Data Terminal Ready (DTR, Ready Out)

NOTE: The remaining pins are not used.

Figure 2-4. Pin Functions of the Comm Link Port

Terminal Port

Figure 2-5 shows the pin functions of the TERMINAL Port.

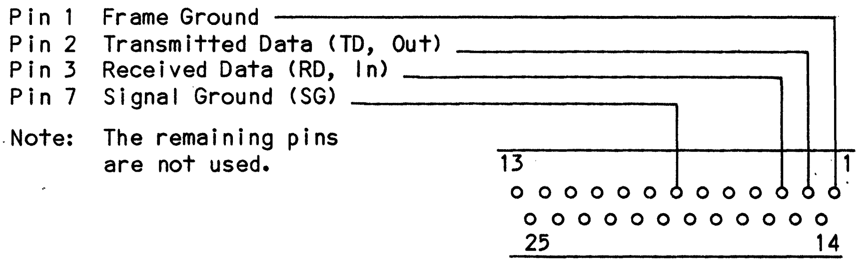


Figure 2-5. Pin Functions of the Terminal Port

Note: In figures 2-6 through 2-9, each unit connected to the RS-232 Interface will be designated as DCE (data communication equipment) or DTE (data terminal equipment. Refer to EIA STANDARD for the RS-232-C Interface.

CONNECTING THE uSE TO A HOST COMPUTER

The uSE is connected to the host computer by connecting the RS-232 cable from the host computer into the serial port on the uSE labeled COMM LINK (see figures 1-3 and 1-7). Connecting the uSE to a host computer provides communications capabilities for software transfer (upload and download) and display terminal interface with the host computer via the uSE.

Figure 2-6 shows the wiring diagram for connecting the uSE to a host computer.

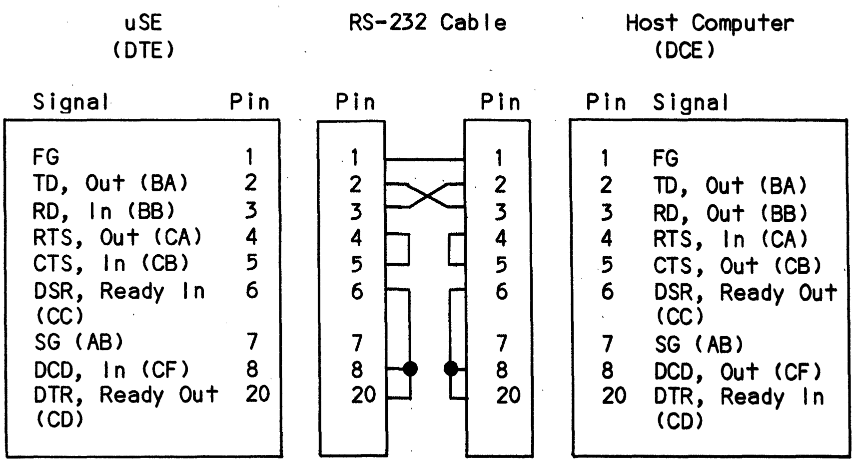


Figure 2-6. uSE to Host Computer Wiring Diagram

CONNECTING THE uSE TO A MODEM

The uSE is connected to a modem by connecting the RS-232 cable from the modem into the serial port on the uSE labeled "COMM LINK" (see figure 1-3). Connecting the uSE to a modem allows the uSE to communicate via telephone lines to interface with another uSE, or any appropriate computer system.

Figure 2-7 shows the wiring diagram for connecting the uSE to a modem. Millennium recommends the Anderson Jacobson modem, model 342A.

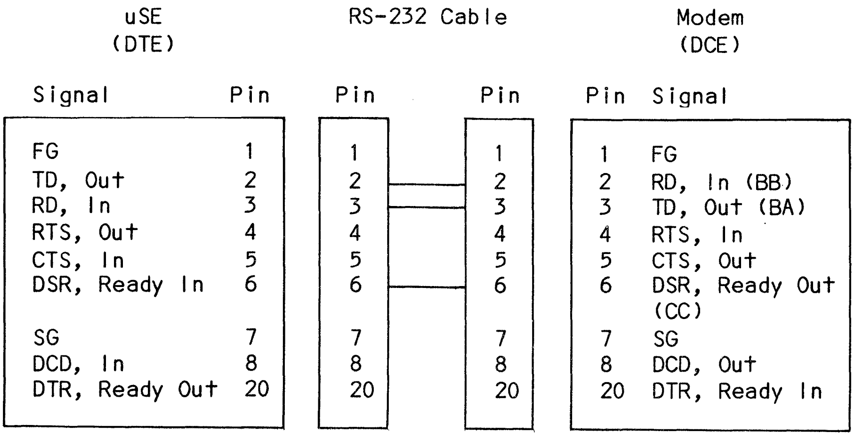


Figure 2-7. uSE to Modem Wiring Diagram

CONNECTING THE uSE TO ANOTHER uSE

A uSE is connected to another uSE by connecting an RS-232 cable from the COMM LINK serial port on one uSE into the COMM LINK serial port on the second uSE (see figure 1-3).

Figure 2-8, shows the wiring diagrams for connection one uSE to another uSE.

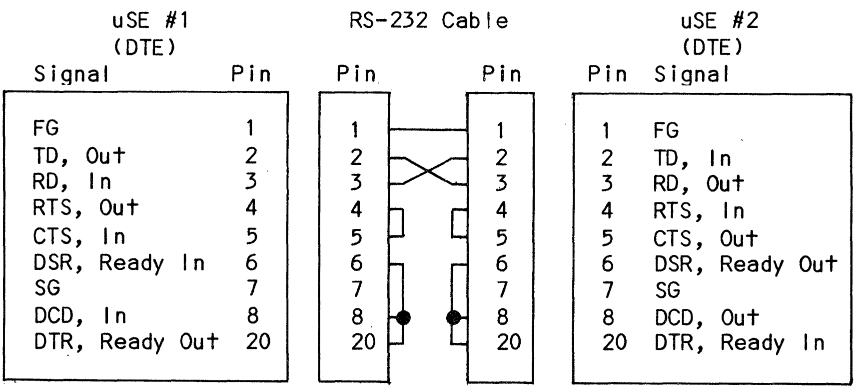


Figure 2-8. uSE to uSE Wiring Diagram

CONNECTING THE uSE TO A DISPLAY TERMINAL

The uSE is connected to a display terminal by connecting the display terminal RS-232 cable into the serial port on the uSE labeled TERMINAL (see figures 1-3 and 1-8).

Figure 2-9 shows the wiring diagram for connecting a uSE to a terminal.

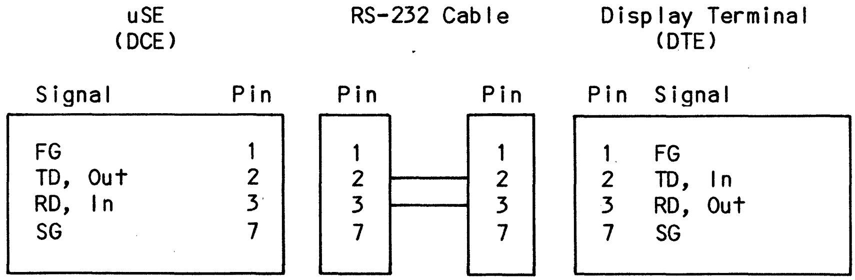


Figure 2-9. uSE to Display Terminal Connection

SETTING THE BAUD RATE FOR THE SERIAL PORTS

Comm Link Port

The REMOTE key sets the baud rate, parity, and selects the mode of operation between the uSE and the host computer or another uSE. Defaults are taken from the DIP switches located at D7 on the Comm/RAM board. Figure 2-10 shows the DIP switch settings.

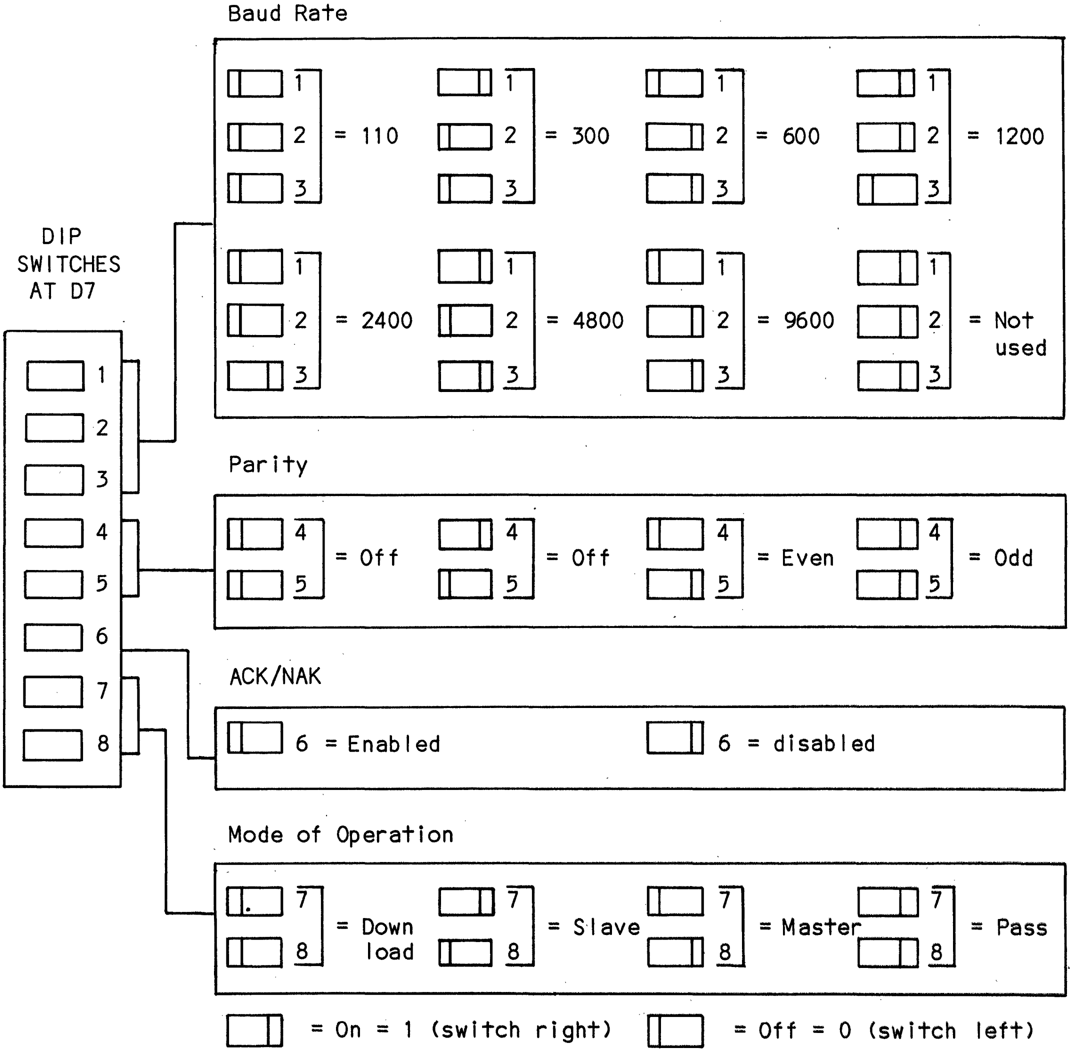


Figure 2-10. DIP Switch Settings for Remote Link

Figure 2-11 shows the position of the Remote Link DIP switches at D7 for the Comm Link default conditions on the COMM/RAM board.

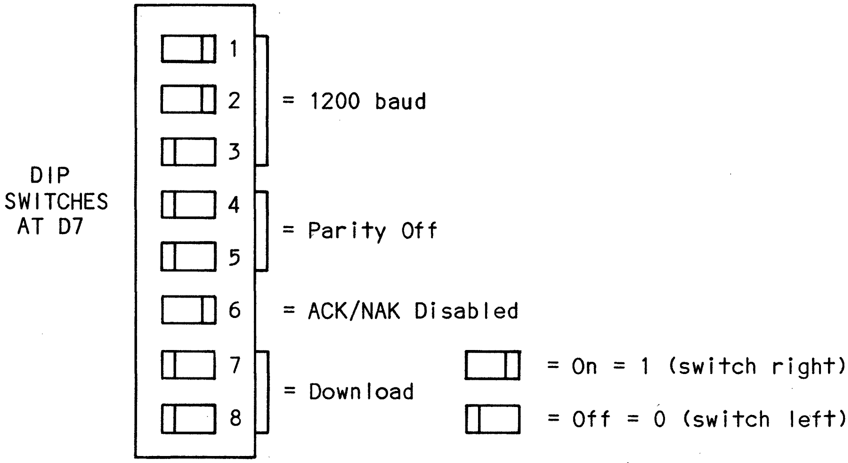


Figure 2-11. Remote Link DIP Switch-Configuration for Default

Figure 2-12 shows the configuration of the remote option byte in memory.

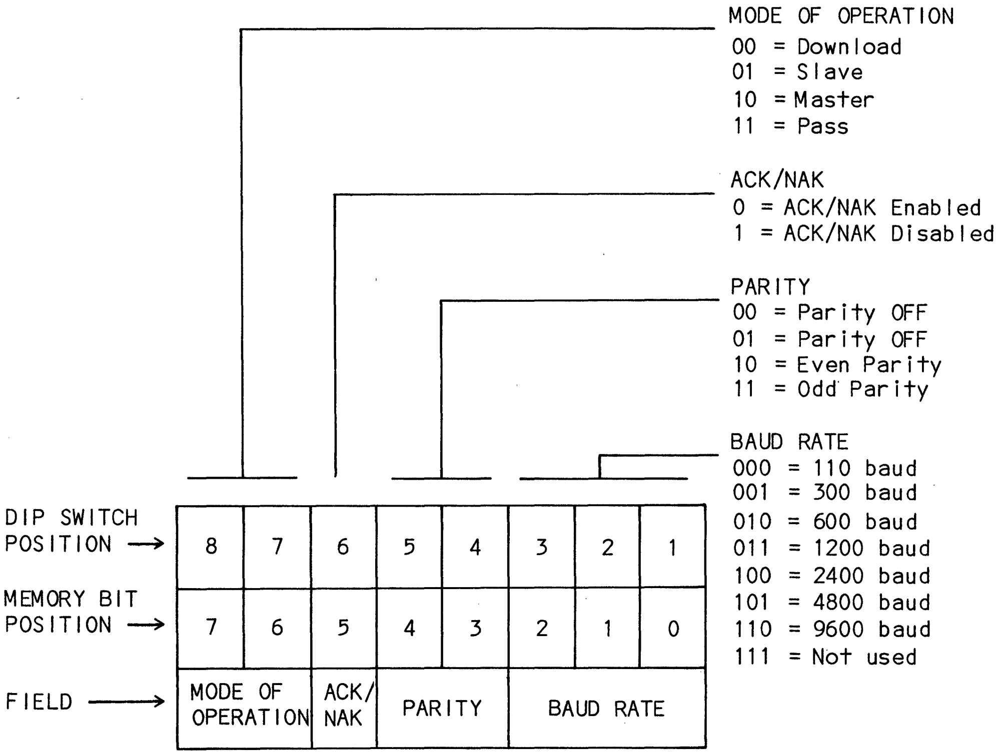


Figure 2-12. Remote Option Byte in Memory

As you can see in the Figures 2-10 and 2-12, the position of the DIP switches are different than the position of the bits in memory that the switches set.

An example of the DIP switches at D7 set for download, ACK/NAK disabled, parity off and a baud rate of 1200 is shown in figure 2-11.

Terminal Port

The baud rate selected for the display terminal must be greater than, or equal to the baud rate set for remote communications; otherwise, the output of data (received via the communications link) to the display will be indeterminate. The user cannot enter the baud rate, parity, etc. for the display terminal from the uSE keyboard. However, these parameters can be set through the DIP switches at location A-10 on the Comm/RAM board. Figure 2-11 shows the position of the DIP switches on the COMM/RAM board. Figure 2-13 shows the DIP switch settings for the display terminal option.

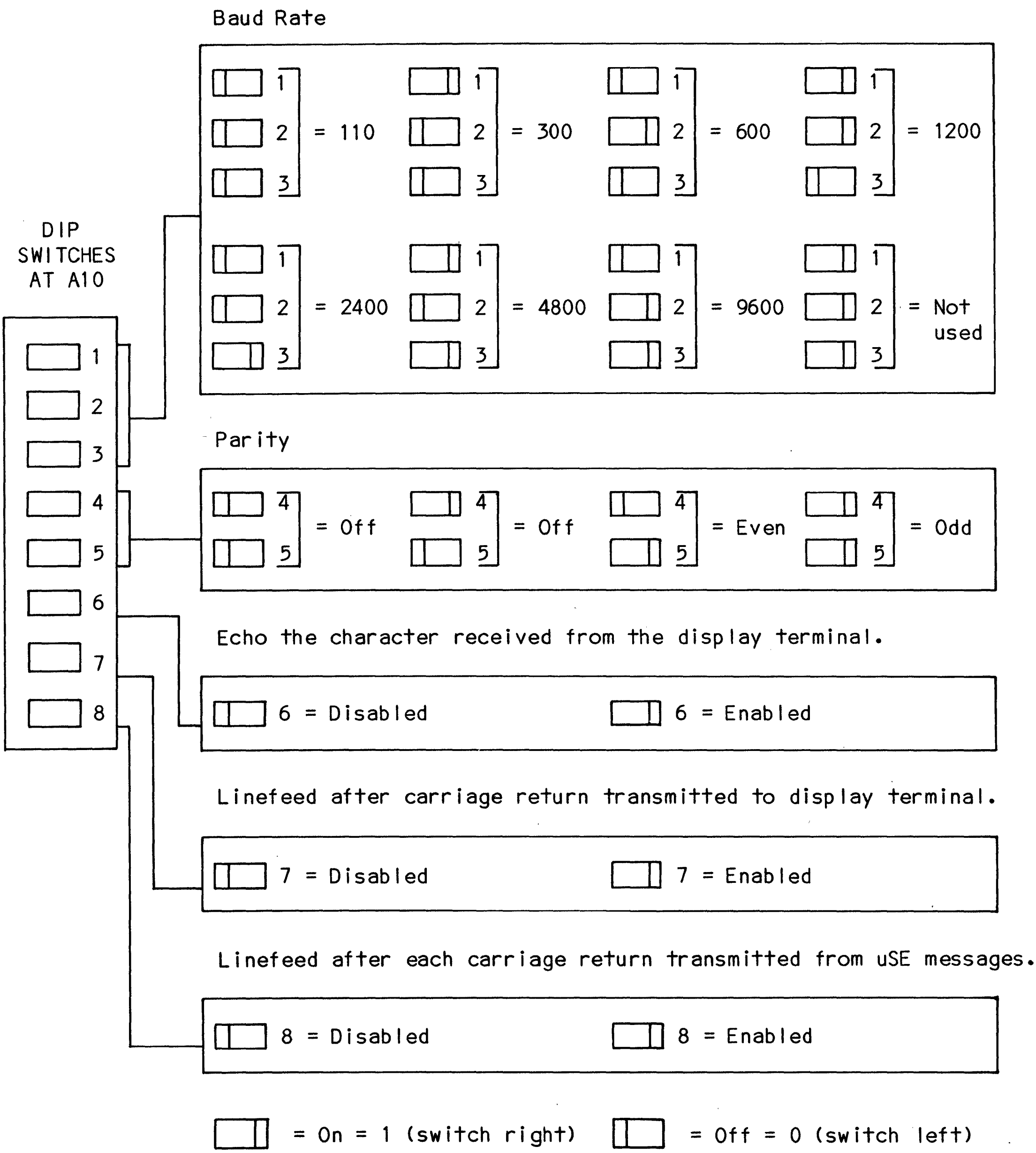


Figure 2-13. Display Terminal DIP Switch Settings

Figure 2-14 shows the configuration of the display terminal option byte in memory.

1 = Append linefeed after each carriage return transmitted.

1 = Append linefeed after each carriage return received.

1 = Echo the character received from the display terminal.

PARITY
00 = Parity OFF
01 = Parity OFF
10 = Even Parity
11 = Odd Parity

BAUD RATE
000 = 110
001 = 300
010 = 600
011 = 1200
100 = 2400
101 = 4800
110 = 9600
111 = Not allowed

DIP SWITCH POSITION →	8	7	6	5	4	3	2	1
MEMORY BIT POSITION →	7	6	5	4	3	2	1	0
FIELD →	TX LF TOO	LF W/ RX	ECH	PARITY		BAUD RATE		

Figure 2-14. Display Terminal Option Byte in Memory

As you can see in figures 2-13 and 2-14, the position of the DIP switches are different than the position of the bits in memory that the switches set.

Figure 2-15 is an example of the switches set for append linefeed after a transmitted carriage return, even parity, and 9600 baud.

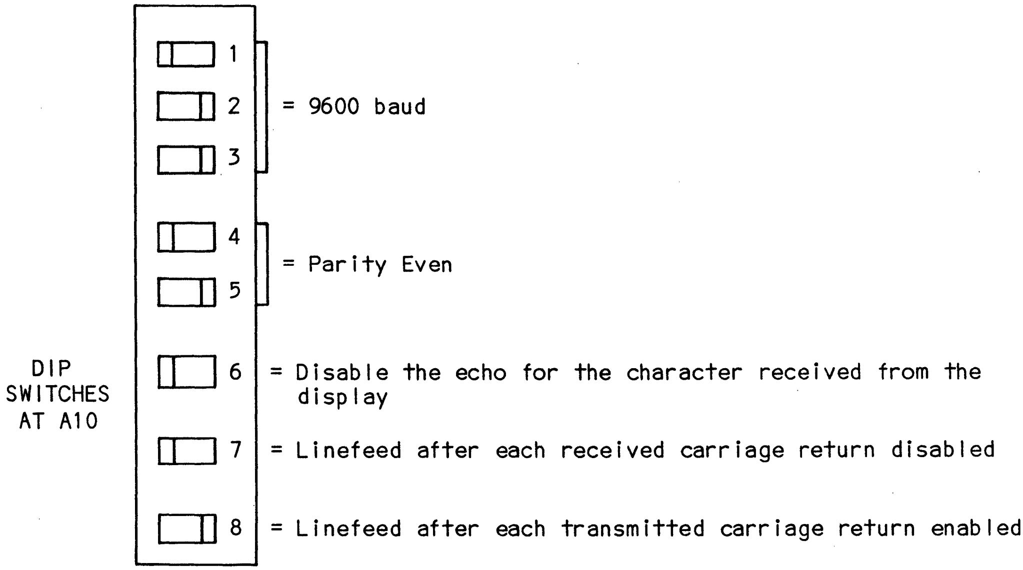


Figure 2-15. Example of Display Terminal DIP Switch Configuration

INTRODUCTION

The keyboard of the uSE is shown in figure 3-1. The keys are divided into two catergories: major function keys and subfunction keys. Major function keys select a particular operation, and subfunction keys modify or select options for that operation. Each major function has its own set of valid subfunctions. Non-valid subfunctions are ignored. The current function can be aborted by pressing any other major function key.

The major function keys are:

1 RESTART
2 SPECIAL CONTROL
3 PROCESSOR CONTROL
4 DISPLAY SELECT
5 REMOTE LINK
6 TRACE CONTROL

The subfunction keys are:

7 SUBFUNCTION CONTROL
8 DATA ENTRY PAD

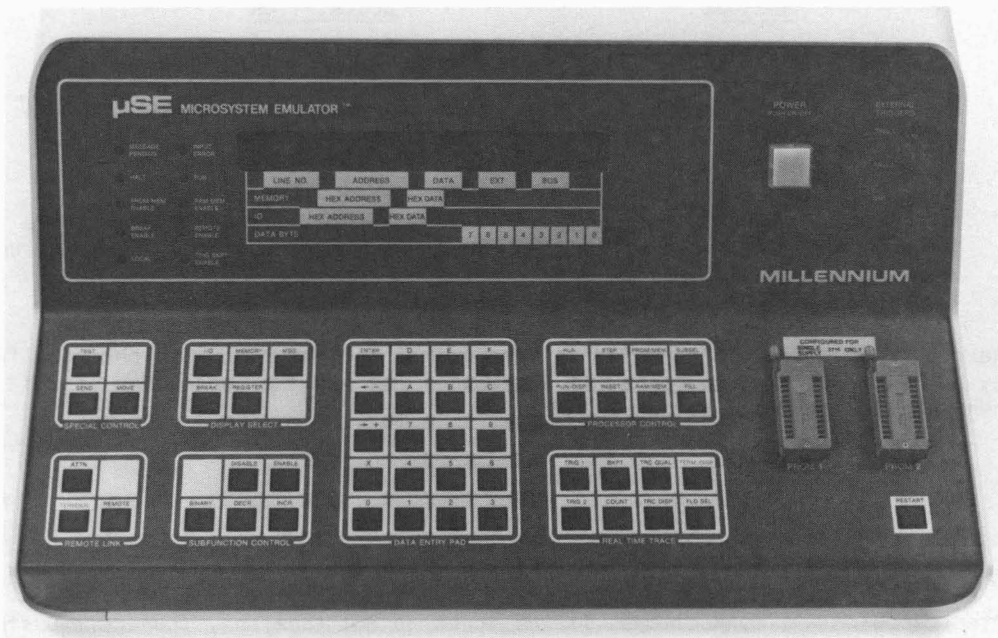


Figure 3-1. MicroSystem Emulator Keyboard

RESTART KEY

This key resets the uSE to an initialized power-on state and tests the Master RAM, Emulator ROM, and System ROM. All front panel indicators are extinguished, all active functions are reset. While the uSE is initializing, the following message is displayed:

```
uSE INITIALIZING
```

After initialization is completed, the following message is displayed to indicate the uSE is ready to accept commands:

```
xxxxxx READY yyyy
```

xxxxxx is the emulator type.
yyyy denotes configuration of front panel PROM sockets. 2708 configuration or 2716 configuration.

If there is an error on startup, the following message is displayed:

MESSAGE	DESCRIPTION
uSE CONDITION xxxx	xxxx = 0001 = Master RAM Failure
	0002 = Emulator ROM Failure
	0004 = Master ROM Failure
	0008 = Emulator not Responding
	0010 = RTT PROM Failure
	0020 = Shadow RAM Failure
	0040 = Comm/RAM Board, PROM Failure
	0080 = Comm/RAM Board, #1 RAM, Failure
	0100 = Extension RAM, #2 RAM, Failure
	0200 = RTT Board is not installed

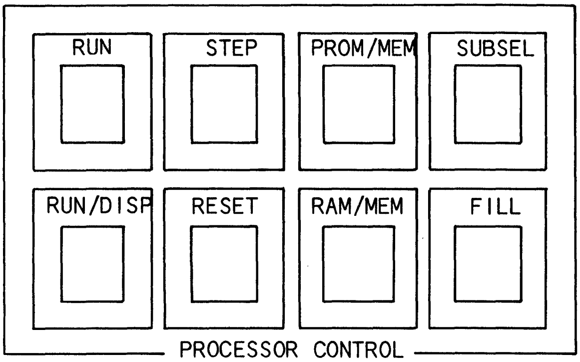
The condition codes are reported in hexadecimal so that more than one condition may be reported. For instance, if the Master RAM and the Shadow RAM both failed then the message would be:

```
uSE CONDITION 0021
```

After checking all connections, if the error is not apparent, then call Customer Service at the toll-free hot-line numbers listed in the preface.

PROCESSOR CONTROL

The Processor Control keys, illustrated and described below, control the operation of the UUT:



RUN Key

This key starts program execution at the address contained in the program counter status register the program next (PN). PN can be changed by using the REGISTER key. When the RUN key is pressed, the RUN LED status indicator is illuminated.

RUN/DISP Key

The RUN/DISP key performs the same functions as the RUN key; and, in addition, displays the contents of either: Memory, Registers, or I/O, ten times per second.

STEP Key

The STEP key starts emulation in either single-step mode or autostep mode. In single-step mode, a single instruction is executed. After the instruction is executed, emulation stops and all displays are updated. To continue instruction execution, the STEP key must be activated for each instruction.

NOTE: The description of the autostep mode requires the mention of several other keys on the uSE keyboard. A full description of these keys is provided in subsequent paragraphs in this chapter.

In the autostep mode, a single instruction is executed and at the completion of the executed instruction, all displays are updated and the next instruction is automatically undertaken. The automatic sequencing of instructions will continue until paused by pressing the STEP key or until the autostep mode is aborted by pressing the MEMORY or I/O keys.

The single-step/autostep modes are controlled by the TERMDISP key. When no functional display is selected and the TERMDISP key is pressed, the uSE front panel is illuminated to display

DISPLAY OFF

indicating that the single-step mode is selected. To enable the autostep mode, the INCR key must be pressed until the uSE display is advanced to the desired operation (e.g., DISPLAY REG, DISPLAY MEM, DISPLAY REG + MEM, etc.) until the uSE displays the required operation. To start the autostep mode, the STEP key is pressed and the uSE goes into the autostep mode. To enable the display terminal, the TERMINAL key on the uSE is pressed and the uSE displays

CRT ENABLED

With the display terminal enabled, the single-step mode or the autostep mode operation is displayed on the display terminal. To pause the display terminal, pressing the space bar on the display terminal will stop the display. To restart the operation, press the space bar again.

RESET Key

The RESET key forces a hardware RESET signal to the emulator microprocessor in the uSE each time the key is pressed. The UUT is not affected and the internal registers of the uSE are not affected.

PROM/MEM Key

This key maps diagnostic PROMs from the PROM sockets on the keyboard into the UUT memory space. The PROM MEMORY ENABLE LED status indicator is illuminated and the initialization message is displayed. The subtest number is initialized to zero. When the PROM/MEM key is pressed the operator sees

Initialize Message

(Dependent on message contained in diagnostic PROM header)

and the subfunction INCR is enabled. The message displayed is the diagnostic initialization message as defined by the PROM header. If the MESSAGE PENDING LED status indicator is on, the INCR key can be used to display multi-line messages. After the PROM/MEM key is pressed, the REGISTER key can be used to find the PROM origin, since the program-counter last (PL) contains the starting address of the PROM overlay. See chapter 6 for more information.

RAM/MEM Key

The RAM/MEM key enables the RAM on the Comm/RAM board and maps it into the UUT address space. Programs are initially placed in this memory by a download via the REMOTE LINK, or by a MOVE function. When the RAM/MEM key is pressed, the uSE displays:

```
DIAG RAM-1 ADR=0000
```

The memory may be used in diagnostic, or applications mode.

Diagnostic Mode

In the diagnostic mode, the data in the RAM must conform to the standard uSE diagnostic PROM data. This means that the first 256 bytes contain control information which is overlayed by the shadow RAM. This mode is exactly the same as that used by the MicroSystem Analyzer. If you are going to use this feature, you should be familiar with the application notes listed in the preface of this manual.

Applications Mode

In the applications mode, there is no requirement to conform to the diagnostic format. The shadow RAM is not used, and no communications to the operator or automatic setup of measurement parameters is provided. This allows direct use of the uSE as a debug tool for user system software. The operator may select 1, 2, 4, or 8K block of RAM memory.

The address entered must be on a boundary corresponding to the amount of RAM enabled, as shown below:

1K boundaries = x000, x400, x800, xC00
2K boundaries = x000, x800,
4K boundaries = x000
8K boundaries = y000

where: x = 0, 1, 2, ... F
y = 0, 2, 4, 6, ... E

Any other address causes a boundary error.

RAM/MEM Subfunctions:

INCR - Selects either the diagnostic or the applications mode. The possible displays are:

```
DIAG RAM-1 ADR=0000
```

```
DIAG RAM-2 ADR=0000
```

```
AP RAM-1=0000  S=1K
```

```
AP RAM-2=0000  S=1K
```

HEXPAD - Allows the operator to modify the address where the diagnostic/application RAM is enabled. In the application mode the size of the RAM can be modified. The only valid entries in the size display are 1, 2, 4 and 8.

ENTER - Freezes the enable address and the size parameter (if present) and enables the memory as specified by the operator (providing an error condition does not exist). If any RAM is already enabled, it will be reenabled to the new specifications.

If the user has specified diagnostic memory and the expansion RAM board is present, then the diagnostic RAM space to be enabled could be up to 16K contiguous bytes. If so, 8K must be enabled as applications RAM otherwise the user could load RAM 1 with diagnostic x, and RAM 2 with diagnostic y and enable either as he wishes.

If the user has specified the 1 applications RAM, then that RAM is enabled. A check is made to verify that the address space to be enabled does not overlap an enabled address space of the extension RAM board. If it does, then a boundary error message will be displayed and the RAM will be disabled.

If the user has specified 2 applications RAM, then two checks are performed: 1) verification that the expansion RAM is present, and 2) check for overlap as described above. If the user attempts to enable expansion RAM and it is not present, then the following message will be displayed:

```
RAM-2 NOT RESPONDING
```

ENABLE - Enables the Comm/RAM board RAM or the expansion RAM according to the current display. An asterisk (*) will be displayed in the left-most position of the display to indicate that the RAM is enabled.

DISABLE - Disables the Comm/RAM board RAM and the expansion RAM.

RAM/MEM Examples

Enabling the RAM as Diagnostic RAM.

Operation	Display	Explanation
RESTART	xxxxxx READY 2708	xxxxxx is the microprocessor type. 2708 or 2716 = PROM Configuration.
RAM/MEM	DIAG RAM-1 ADR=0000	
2, ENTER	DIAG RAM-1 ADR=2000	Select the starting address to be at 2000H.
ENABLE	Initial Message	RAM has been enabled as diagnostic RAM. The initial message from the PROM HEADER is displayed and the OPT/MEM ENABLE LED is illuminated.

Enabling 2K bytes of applications RAM starting at address 0.

Operation	Display	Explanation
RESTART	xxxxxx READY 2708	xxxxxx is the microprocessor type. 2708 or 2716 = PROM Configuration.
RAM/MEM	DIAG RAM-1 ADR=0000	
INCR	DIAG RAM-2 ADR=0600	
INCR	AP RAM-1=0000 S=1K	Choose the application mode.
ENTER	AP RAM-1=0000 S=1K	Keep starting address of 0.

2, ENTER	AP RAM-1=0000 S=2K	Select a block size of 2K bytes.
ENABLE	* AP RAM-1=0000 S=2K	Enable the RAM. The RAM/MEM ENABLE LED is illuminated and * appears in the display.

SUBSEL Key

The SUBSEL key selects one of 99 possible subtests as defined in the Diagnostic Header. Selecting Subsel=00 causes all the diagnostic subtests to be executed. Selecting subroutine N causes only the designated subtest to execute. If the subtest number selected does not exist in the program, an input error occurs. This causes the INPUT ERROR LED status indicator to light and the audio error alert to be sounded.

When the SUBSEL key is pressed the uSE displays

SUBSELECT=dd	dd = the subtest number in decimal.

and the subfunction DECIMAL keypad is enabled. After a decimal number is entered, pressing the ENTER key terminates the input mode.

FILL Key

The FILL key stores a specified byte of data in RAM (local or user) according to the current mapping. All data is verified as the fill process occurs. Any byte that will not verify produces the following error display and terminates the fill process:

FILL ERR xxxx=yy	xxxx = the address where the error occurred. yy = the data read after the write.

After pressing the FILL key, the uSE displays

FILL xxxx yyyy zz	xxxx = starting address. yyyy = ending address. zz = data byte to be stored.

and prompts for the beginning address. After the starting address has been specified the ENTER key is pressed and the uSE prompts for the ending address:

FILL xxxx yyyy zz

After the ending address has been specified, the ENTER key is pressed and the systems prompts for the data used to fill memory:

```
FILL xxxx yyyy zz
```

When the operator terminates this field (using the ENTER key), the uSE fills the memory with the specified byte from the starting address to the ending address and displays

```
FILL COMPLETE
```

if there are no errors. If the ending address is smaller than the starting address, the fill process will wrap-around memory.

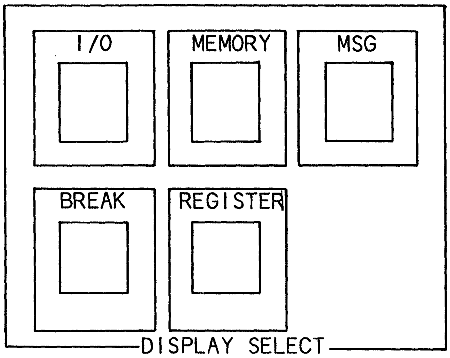
FILL Subfunctions

HEXPAD - Allows the operator to modify the starting and ending addresses.

ENTER - Freezes the starting address, ending address and the data byte. When the ENTER key is pressed to freeze the data byte, the memory fill is initiated.

DISPLAY SELECT KEYS

The DISPLAY SELECT keys, illustrated and described below, select the functions to be displayed:



I/O Key

The I/O key reads from or writes to I/O devices in the UUT. Data can be read or written in hexadecimal or binary. The displays shown vary according to the microprocessor being emulated. (See emulator supplements for details.)

BREAK Key

The BREAK key selects the address, options, and parameters for the hardware breakpoint. The uSE can break on:

READ ACCESS
WRITE ACCESS
READ OR WRITE ACCESS
PASS COUNT

Pass count is the number of times a location is referenced before a break occurs.

The operator selects the action is to be taken when a breakpoint is found. The options are to pause the processor or to jump to a specific address. When the operator presses the BREAK key, the uSE displays the current break address and options:

```
BRKA=xxxx OPT=y
```

xxxx = break address.
y = the options in hexadecimal format.

Pressing the BINARY key displays the options in binary format as shown on the following page:

BITS	FUNCTION
3	1 = Break On Write Access at the specified address
2	1 = Break On Read Access at the specified address
1	1 = Pass Count Enable
0	0 = Pause uP, 1 = Jump to Address

The default value, OPT = C, breaks on write or read access and pauses the processor. Either break on write access or break on read access must be set.

After all the fields have been edited, the existence of other displays depends upon which options were selected. If the PASS COUNT option was selected, pressing the INCR key displays:

PASS COUNT = dddd

dddd = number of times the specified memory location is accessed before a break occurs. Range is 1 through 9999.

If the JUMP option was chosen, the INCR key is used to display:

JUMP ADDR = xxxx

xxxx = address the uSE jumps to when breakpoint occurs.

After the breakpoint options and addresses are entered, the ENABLE key activates the breakpoint. ENABLE must be pressed after the BREAK setup and before another major function key is pressed. When a breakpoint is enabled, the BREAK ENABLE LED status indicator is illuminated. When the UUT program is executing and a breakpoint occurs, the uSE displays:

NOTE: If the display terminal is enabled, a dump of the data currently selected via the TERMINAL display key will be output to the display.

BREAK POINT = xxxx

xxxx = address of the instruction where the breakpoint occurred.

A breakpoint is cleared by pressing the BREAK key and then pressing the DISABLE key.

BREAK Subfunctions

HEXPAD - Allows the operator to change the contents of RAM memory.

BINARY - Displays the contents of RAM memory in binary.

INCR - Increments the memory location by one, and displays the contents of that location.

ENABLE - Enables the breakpoint.

DISABLE - Disables the breakpoint.

MEMORY Key

This key is used to display or modify the contents of UUT memory in hexadecimal or binary format unless:

1. The PROM/MEM function is enabled and the address selected falls within the boundaries of the diagnostic PROM; in that case, the diagnostic PROM memory is displayed.

2. The RAM/MEM function is enabled and the address selected falls within the boundaries of the Comm/RAM memory enabled; in this case, Comm/RAM memory is displayed.

3. The display terminal is enabled, in which case, the display will be

DUMP MEM xxxx yyyy	xxxx = starting address. yyyy = ending address.

When the MEMORY key is pressed, the auto-edit mode is enabled. If the emulator is running, the processor is paused,

```
MEM xxxx
```

is displayed, and the hex keypad subfunction is enabled to allow a new address to be entered. After the ENTER key is pressed, the contents of the specified address are displayed:

MEM xxxx=yy	xxxx = the memory location address. yy = the contents of that address.

If the user attempts to modify ROM or there is an error in modifying memory an error message is displayed:

MEM xxxx=yy ERR	yy = the data read from the memory address after attempting to modify it.

MEMORY Subfunctions

HEXPAD - Allows the operator to change the contents of RAM memory.

BINARY - Displays the contents of RAM memory in binary.

INCR - Increments the memory location by one, and displays the contents of that location.

DECR - Decrements the memory location by one and displays the contents of that location.

ENTER - Without the display terminal enabled, freezes address and data and advances to the next address. With the display terminal enabled, freezes start and stop addresses and initiates the dump to the display terminal.

REGISTER Key

The REGISTER key displays or modifies the UUT microprocessor internal registers. Each specific microprocessor has its own register notations and therefore, its own displays (see the supplement for the individual microprocessor emulator board). One or more registers are displayed when the REG key is pressed. The next set of registers (in the register list) is displayed when the INCR key is pressed. By continuing to press the INCR key, all of the registers can be displayed. Each uSE display contains fields that can be modified. After editing the first field and pressing the ENTER key, the cursor will automatically move to the second field.

REGISTER Subfunctions

HEXPAD - Using the hexpad, the operator can modify the contents of the registers.

INCR - Increments to the next entry in the linked register list.

DECR - Decrements to the previous entry in the linked register list.

ENTER - The ENTER key freezes the data in registers.

MSG Key

This key displays messages on the front panel display as requested by the diagnostic programs testing the UUT. When the display is being used, the MESSAGE PENDING LED indicator is illuminated to notify the operator of any messages waiting to be displayed. Messages for display can be of a single or multi-line. The MESSAGE PENDING LED indicator is extinguished when all messages have been displayed.

The MSG function is automatically initiated when the diagnostic memory is enabled and the RUN key is pressed.

If the MSG key is pressed but the MESSAGE PENDING LED is not illuminated then

NO MESSAGE

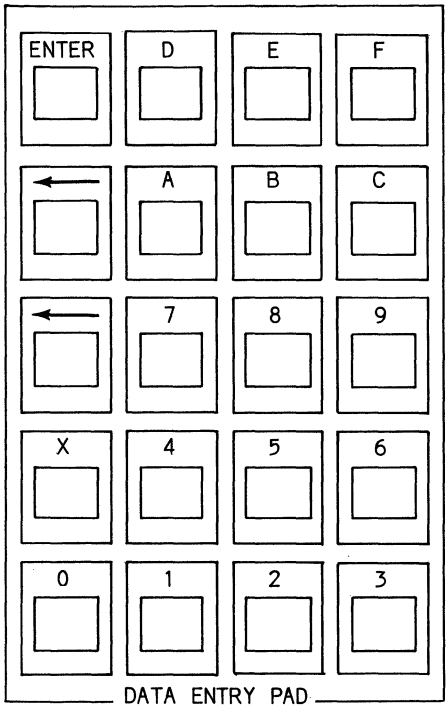
would be displayed.

MSG Subfunctions

INCR - Displays the next line of a multi-line message.

DATA ENTRY PAD

The DATA ENTRY PAD, illustrated and described below, is used to input and edit data, select options, and specify parameters for major functions:



Hex Keypad

The hex keypad contains the hexadecimal character set 0-9 and A-F and is used to enter data and addresses. Data can be modified only in the input mode. In some cases, this mode is entered automatically as soon as a function key is pressed (auto-edit). In other cases, it is entered as soon as a valid key on the data entry pad is pressed (implicit-edit). When a message has been displayed and auto-edit has not been entered, edit mode is entered by pressing a key of the appropriate key pad. The uSE scans the display for the first input field, substitutes the typed character for the first character, and sets the cursor position at the next character, if any. The cursor position is identified by a blinking character followed by a period. Only the character pointed to by the cursor can be changed.

Arrow Keys

The cursor position is controlled by the right arrow (--> +) and left arrow (<-- -) keys. The arrow keys move the cursor position to the right or left. If the cursor is positioned at the right boundary of a field, the --> + key is ignored, and the cursor position remains unchanged. Likewise, if the cursor is at the left boundary of a field, the <-- - key is ignored.

ENTER Key

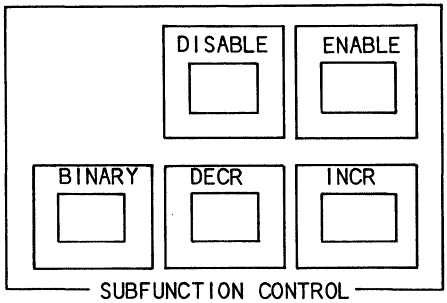
The ENTER key terminates data input and enters the displayed value of the current field. If there are more input fields in the display, the cursor is positioned at the first character of the next field.

X Key

The X key is used to enter don't cares into the current cursor position, if appropriate. The X key may also be used for the Real-Time Trace, Data Acquisition mode. (See Data Acquisition, chapter 5.)

SUBFUNCTION CONTROL KEYS

The SUBFUNCTION CONTROL keys, illustrated and described below, operate in conjunction with the major function keys:



BINARY Key

Displays data byte of the current display value in an 8-bit binary mode.

DISABLE Key

A breakpoint is disabled (cleared) by pressing the BREAK key and then pressing the DISABLE key. The DISABLE key also disables RAM memory.

ENABLE Key

This key enables the breakpoint and the RAM memory.

DECR Key

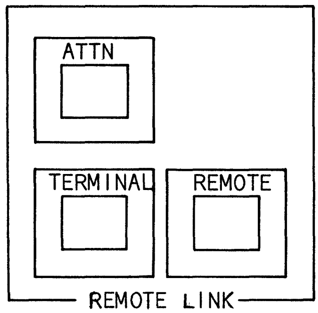
Decrements the memory and I/O addresses.

INCR Key

Increments the memory and I/O addresses, linked displays, and register content display. The INCR key is also used with the remote option, the RAM/MEM key, the MOVE and SEND keys, and the TERMINAL display key to increment through RAM memory, TRIG 1 and TRIG 2 options, and the TRC DISP displays.

REMOTE LINK KEYS

The REMOTE LINK keys, listed and described below, allow the uSE to operate as a remote slave controlled by a master uSE. Diagnostic programs can be downloaded via telephone link from a master uSE or a host computer:



REMOTE Key

The REMOTE key sets the baud rate, parity, and selects the mode of operation for the remote communications link. The format of the remote link byte is shown in figure 3-2. Defaults are taken from the DIP switch settings at location D7 on the Comm/RAM board (see figure 2-11).

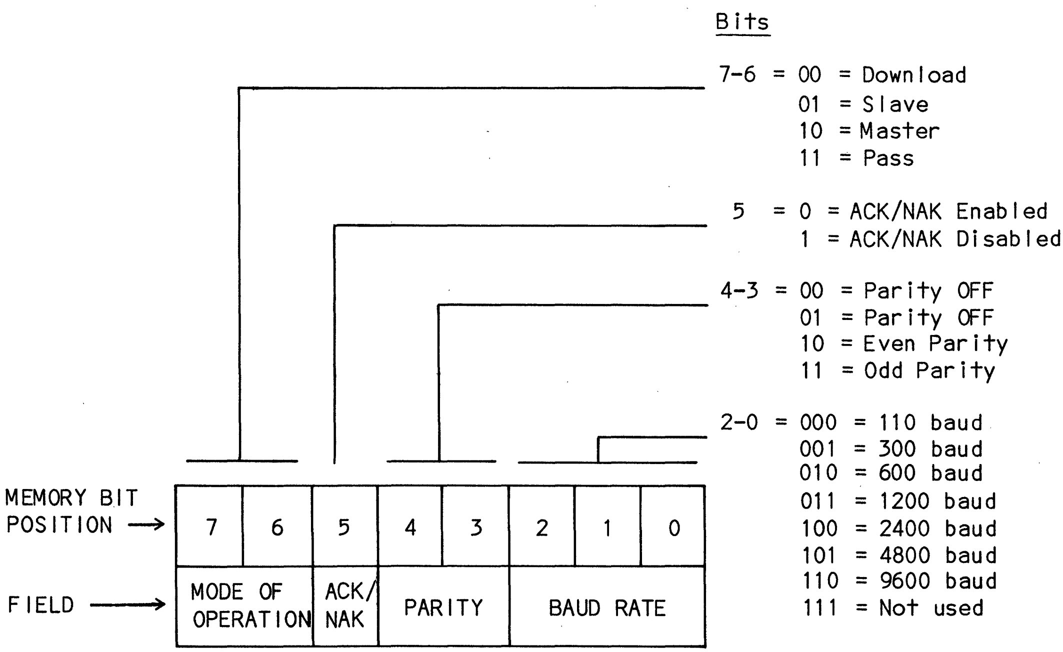


Figure 3-2. Remote Option Byte in Memory

During initialization of the uSE, the defaults for the remote function are read from the DIP switches at location D-7 on the Comm/RAM board and stored. The most-significant bit of the DIP switches, at location A-10, is sampled to determine if the transmitted messages are to be followed by a line feed (see figure 2-11), and the affirmative acknowledge sequence 0<CR> (30H, 0DH) and the negative acknowledge sequence 7<CR> (37H, 0DH) are stored in the data base to be used while ACK/NAK is being enabled.

The ACK/NAK sequence may be altered prior to enabling the remote options. There are eight bytes reserved in the data base for each of the ACK/NAK sequences and the operator may specify up to seven characters in either or both sequences, but the ACK/NAK sequences must be terminated with FFH in the eigth byte.

The remote option is enabled by pressing the ENABLE key. If the options for the function are not correct, the operator can change then via the hexpad and the ENTER key. The new data will be displayed when the REMOTE key is pressed.

If the remote option has been enabled, and local mode has selected; then, to re-enable the remote option (master, slave, download pass), the REMOTE key is pressed.

REMOTE Subfunctions

MODIFY - Allows the operator to modify the remote options.

BINARY - Allows the operator to modify the options in binary.

INCR - Increments the display.

REMOTE Displays

Display	Explanation
REM OPTS = xx	
REM OPTS = bbbbbbbb	The operator may select desired options in either hex or binary.
ACK = 300DFFFFFFFFFF	Default ACK sequence by initialization.
NAK = 370DFFFFFFFFFF	Default NAK sequence by initialization.

When changing the ACK/NAK sequence, any sequence less than seven characters long must be followed by a character of all ones (FFH).

Remote Example

The following example includes a sequence that enables the remote option:

Operation	Display	Explanation
RESTART	xxxxxx READY 2708	xxxxxx is the microprocessor type. 2708/16 = PROM Configuration.
REMOTE	REM OPTS = 96	The default options are switch selectable.
BINARY	REM OPTS = 10010110	Display the options in binary.
10000110, ENTER	REM OPTS = 10000110	Selection: Master, ACK/NAK, No parity, 9600 Baud.
INCR	ACK=300DFFFFFFFFFF	ASCII 0 followed by a CR is default for Data Message Received without an error.
INCR	NAK=3700FFFFFFFFFF	ASCII 7 followed by a CR is default for Data Message Received with an error.
INCR	REM OPTS = 10000110	Back to remote options.
ENABLE	MASTER MODE	Enable the chosen options. The Remote ENABLE LED is illuminated.

When in the remote mode, the operator can select local mode by pressing the ATTN key. To return to the remote mode (slave, master, download: 1.Press the REMOTE key, this will display the remote options, if no changes are necessary, press the ENABLE key to enable the remote function; 2.Press the ATTN key again, which will Transmit an escape (1BH) on the COMM LINK and redisplay the remote mode previously enabled.

TERMINAL Key

The TERMINAL key enables or disables input/output to a display terminal connected to the uSE via the TERMINAL RS-232 port. Only limited input from the keyboard to the uSE is allowed.

TERMINAL is an alternate action key. The display terminal is not enabled by system initialization. The first time it is pressed the uSE displays:

CRT ENABLED

The next time the TERMINAL key is pressed the uSE disables the display terminal and displays:

CRT DISABLED

NOTE: All received data will be displayed on the display terminal subject to the display terminal transmit data register being empty. When the uSE is in the CRT ENABLED mode, all data received from the uSE communications link (including download data, key-stroke message data) is output to the display terminal. In addition, all data input from the display terminal keyboard is transmitted out the uSE communications link, if a uSE dump to the display terminal is not in progress. If a dump is in progress, the characters that would cause the dump to pause, continue, or abort, will not be transmitted.

The following input from the display terminal keyboard is allowed during a uSE dump to the display terminal:

1. PAUSE - the operator can pause the next line of data to the display terminal indefinitely by pressing the SPACE bar on the display terminal keyboard. If transmission of a line of display data has begun, then it will continue until the next carriage return.

2. CONTINUE - the operator can continue the paused line and any following lines of data to the display terminal by pressing the SPACE bar on the terminal keyboard again.

3. ABORT - the operator can abort the next line and any following lines of data to the display terminal, excepting download data, by pressing any character generating key on the display terminal keyboard (i.e., virtually any key excepting the SPACE bar). Also, pressing any key on the uSE keyboard aborts output to the display terminal.

The baud rate selected by the REMOTE key for remote communications must be equal to or less than the baud rate of the display terminal; otherwise, the output of data (received via the communications link) to the display terminal will be indeterminate.

ATTN Key

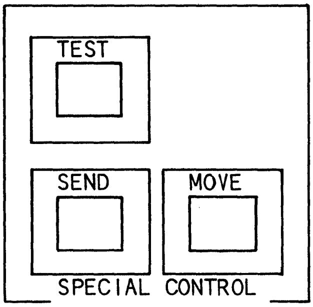
The ATTN (attention) key regains control of the uSE when it is interfacing with a host computer or a master uSE. When the remote is enabled, as a master or slave, keystrokes from the front panel are no longer treated as functions. In the slave mode, keystrokes are ignored, in the master mode they are transmitted as keystroke messages for the slave to act upon. The ATTN key is used as a mechanism for temporarily breaking the remote link to allow functions to be performed locally.

Pressing the ATTN key once allows all keys to be acted upon locally. Pressing the REMOTE key returns the uSE to the Remote Mode. When the uSE is in the Remote Mode, the REMOTE ENABLE LED on the front panel is illuminated. When the uSE is in the Local Mode, the LOCAL LED on the front panel is illuminated.

Pressing the ATTN key twice in succession will cause an Operator Attention Message to be transmitted. The ATTN message is 1BH (ASCII escape key).

SPECIAL CONTROL KEYS

The SPECIAL CONTROL keys are illustrated and described below:



TEST Key

The TEST key performs a self-test of the uSE, checking RAM, ROM, the front panel indicators, and various internal functions, then resets it to the initialized power-on state. While the uSE is initializing, the following message is displayed:

```
uSE INITIALIZING
```

After initialization is completed, the following message is displayed to indicate the uSE is ready to accept commands:

```
xxxxxx READY   2708
```

xxxxxx is the emulator type.
2708 or 2716 is the PROM Configuration.

If there is an error on startup, the following message is displayed:

```
uSE CONDITION xxxx
```

xxxx = error code. See appendix C for details.

MOVE Key

This function allows the operator to move blocks of data from one memory space to another. Five types of moves are allowed by this function:

1. PROM to RAM (PR) - moves data from front panel PROM(s) to option RAM.
2. PROM to USER (PU) - moves datas from the front panel PROM(s) to user RAM.
3. RAM to USER (RU) - moves data from option RAM to user RAM.
4. USER to RAM (UR) - moves data from user memory to optional RAM.
5. RAM to RAM (RR) - move data from one area of optional expansion RAM to another.

When the MOVE key is pressed the uSE displays:

```
MV ww xxxx yyyy zzzz
```

ww = Source and destination of data (PR, PU, RU, UR or RR).
xxxx = Starting Address (source).
yyyy = Ending Address (source).
zzzz = Starting Address (destination).

Optional RAM must be enabled prior to a move that reads/writes system RAM. If the RAM is not enabled, the following message is displayed:

```
ENABLE RAM @ xxxx
```

MOVE Subfunctions:

INCR - This key increments through the source and destination options (the ww Field), allowing the operator to choose one of the five options.

HEXPAD - Using the hexpad, the operator is able to modify the start and end addresses.

ENTER - The ENTER key freezes the starting source address, ending source address and starting destination address. When the ENTER key is pressed to freeze the starting destination address this initiates the move. If the starting source address has not been modified and the ENTER key is pressed, the move is initiated with the displayed addresses.

MOVE Displays:

Display	Explanation
MOVE IN PROGRESS	The message is displayed as the operation is executed.
BOUNDARY ERROR	The operator has tried to cross an illegal boundary. For example, moving from PROM to emulation RAM. If the user has 2708 PROM, only less than 2K may be moved.
WRITE ERROR xxxx	This message is displayed when a memory write error has occurred. xxxx = the address where the error occurred.
MOVE COMPLETE	This message is displayed when the operation is completed.
ENABLE RAM @ xxxx	This message indicates that emulation RAM was not enabled before the move was attempted.

SEND Key

The SEND key transmits blocks of data from memory over the remote communications link. (The remote link must have been set up first.) Data is sent from either the emulation RAM or the front panel PROM.

When the SEND key is pressed, the uSE displays:

Display	Explanation
SEND x yyyy zzzz	x = Data source (P=PROM, R=RAM) yyyy = Starting address zzzz = Ending address

The Emulation RAM in the uSE must be enabled prior to the send. If the RAM is not enabled, the following message is displayed:

```
ENABLE RAM @ xxxx
```

SEND Subfunctions:

INCR - Changes the data source field, x (P or R).

MODIFY - Using the hexpad, the operator can modify the starting and ending addresses.

ENTER - The ENTER key freezes the starting address and the ending address. When the ending address has been frozen, the ENTER key sends the data (i.e., information contained in the memory between the start and end addresses inclusive), over the remote link in download message format. If the addresses have not been modified, the ENTER key sends the data.

SEND Displays:

Display	Explanation
SEND IN PROGRESS	The transmitting uSE displays this message to indicate a transmission is in progress.
LOAD IN PROGRESS	The receiving uSE displays this message to indicate a load is in progress.
SEND COMPLETE	When the transmitting uSE has sent the data, this message is displayed.
LOAD COMPLETE	When the receiving uSE has received the data, this message is displayed.
ENABLE RAM @ xxxx	This message indicates that emulation RAM was not enabled before the send was attempted.
or	
REMOTE NOT ENABLED	This message indicates the remote unit has not been enabled and is unable to perform the request

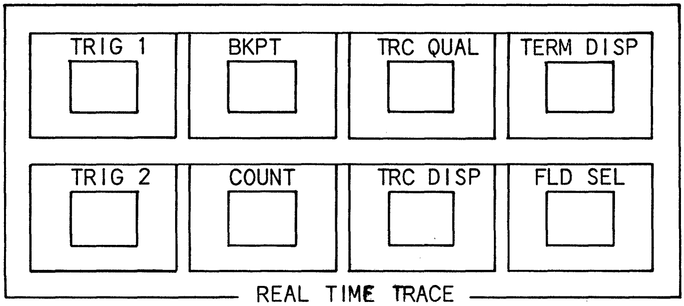
SEND Example:

Send the data from the master uSE front panel PROMs to slave uSE. The PROM data is at address 2000H thru 27FFH. The remote link for the master and slave uSE's have already been established.

Operation	MASTER uSE Display	Explanation	SLAVE uSE Display
ATTN	LOCAL MODE	Return master to local mode. LOCAL LED is illuminated to indicate local mode.	
SEND	SEND P 0000 0000	Choose the send function.	
2, ENTER	SEND P 2000 0000	Select starting address 2000H.	
27FF	SEND P 2000 27FF	Select ending address 27FFH.	
ENTER	SEND IN PROGRESS	Start transmission.	LOAD IN PROGRESS
	SEND COMPLETE	All the data has been sent.	LOAD COMPLETE

REAL-TIME TRACE KEYS

The REAL-TIME TRACE keys are illustrated and described briefly below. See chapter 5 for a detailed description.



TRIG 1 Key

The TRIG 1 key defines the trigger equations. When all elements are satisfied, the appropriate trigger pulse is provided on the uSE front panel trigger jack. The event equations can also be used in conjunction with the BKPT function to terminate the run condition.

External data can be applied to the equation by using the 8-bit external probe input. Data is sampled on the external probes synchronously to bus activity or asynchronously using an external clock.

TRIG 2 KEY

The TRIG 2 key operates the same as the TRIG 1 key except for the Mode Parameter.

BKPT Key

The BKPT Key enables the user to set break points determined by the TRIGGER 1 and TRIGGER 2 equations. Options and Mode are selected by using the FLD SEL key.

NOTE: If the display terminal is enabled when such a break occurs, the data selected via the TERM DISP key will be output to the display terminal.

TRC DISP Key

The trace display key provides for the display of information accumulated in the real-time trace buffer.

TRC QUAL Key

The TRC QUAL key allows the user to select the type of transactions that are stored in the Real-Time Trace Buffer.

COUNT Key

The COUNT key allows the counting of the various parameters up to a maximum value of 65,534 units. Maximum counts are an indication of counter overflow and therefore, cannot be viewed as valid measurements.

FLD SEL Key

The FLD SEL key is used to select options and parameters for TRIG 1, TRIG 2, and TRC QUAL keys.

TERM DISP Key

This key displays the type of data to be output to the display terminal. If the display function is OFF, the autostep function is disabled; if the display function is not OFF, the autostep function is enabled. If the display terminal is enabled, data corresponding to type (S) selected will be output to the display terminal.

TERM DISP Subfunction

INCR - Selects and displays the next type (S) of a data to be output to the display terminal.

TERM DISP Displays

Key Strokes	Display	Explanation
TERM/DISP	DISPLAY OFF	
INCR	DISPLAY REG	Dump all register data.
INCR	DISPLAY MEM	Dump memory by parameters entered through the DUMP MEM function.
INCR	DISPLAY REG+MEM	
INCR	DISPLAY I/O	Display the last examined I/O port and its current data.
INCR	DISPLAY REG+I/O	
INCR	DISPLAY MEM+I/O	
INCR	DISPLAY REG+MEM+I/O	
INCR	DISPLAY OFF	Wraps around.

INTRODUCTION

The Remote Link RS-232 interface of the uSE permits two uSE's to be connected through a modem or data set. When the Remote Link is enabled, the onsite slave uSE emulates the Unit Under Test (UUT) and performs all normal functions, but the master uSE controls the operation of the slave uSE from the master uSE keyboard. The results are displayed on the master uSE and the slave uSE.

REMOTE CONTROL LINK

The remote link can operate in Slave, Master, Download, and Pass modes.

Slave Mode

When a uSE is operating in the slave mode, it is totally under the control of the master uSE. The keyboard of the slave uSE is locked out and all operations are initiated by keystroke messages received from the master uSE. Results are displayed on the slave uSE and transmitted back to the master uSE as display messages and LED messages.

Master Mode

When a uSE is operating in the master mode, it totally controls the operation of the slave uSE. The master uSE does not perform any functions locally; instead, all keystrokes are transmitted. All display messages and LED messages are received from the slave uSE and displayed on the master uSE.

Download Mode

When a uSE is operating in the download mode, it is connected to a host computer system. The operator can download a program from the host computer into the uSE RAM memory (up to 16K). The program can then be executed by the uSE.

In the slave, master or download mode, with the display terminal enabled, one of the following will occur:

1. All transmitted data will be displayed on the display terminal.

2. All received data will be displayed on the display terminal subject to the display terminal transmit data register being empty.

Pass Mode

When a uSE is operating in the pass mode, any data received from the host computer is transmitted directly to the display terminal without being operated on. Likewise, any data from the display terminal is passed directly to the host computer. However, the display terminal must be enabled by the uSE TERMINAL key.

DATA MESSAGES

Four types of data messages can be transmitted and/or received by the uSE. A fifth type of data can be transmitted and/or received by the uSE but the uSE does not manipulate the data.

LED (front panel indicators)
KEYSTROKE
DISPLAY (front panel alphanumeric display)
DOWNLOAD (memory load)
HOST (to and from host computer and display terminal)

All data messages consist of printable ASCII characters, with the exception of the carriage return (ODH) and the optional line feed (OAH). Those characters indicate the end of a message. Table 4-1 shows what data can be received and transmitted by each of the remote modes.

Table 4-1. Data Messages

MESSAGE	MODE							
TYPE	DOWNLOAD		SLAVE		MASTER		PASS	
	RECEIVE	TRANSMIT	RECEIVE	TRANSMIT	RECEIVE	TRANSMIT	RECEIVE	TRAN
Keystroke			X			X		
LED				X	X			
Display				X	X			
Download	X	X	X	X	X	X		
HOST							X	

All data except HOST is affirmatively acknowledged (ACK) by the receiver. If an error is found in the message, a negative acknowledge (NAK) is sent. The message is retransmitted if an ACK <u>is not</u> received after a transmission, or a NAK <u>is</u> received after a transmission. If an ACK is not received after the fifth transmission, the task is terminated and the following message is displayed:

TRANSMISSION ERROR

LED Messages

The LED message defines a 16-bit integer, bit coded, to specify the ON/OFF state for each LED. These messages are generated by a uSE when it is in the slave mode and received and acted upon by a master uSE. When an LED message is received by a master uSE, it illuminates the appropriate LED on its front panel.

Bit values of the LED display lights are shown in Table 4-2.

Table 4-2. LED Bit Values

Bit	LED	Bit	LED
	(Most significant bit)		
15	BEEP (Audible Alarm)	7	RAM MEM ENABLE
14	Unused (must = 0)	6	RUN
13	Unused (must = 0)	5	HALT
12	Unused (must = 0)	4	PROM MEM ENABLE
11	Unused (must = 0)	3	BREAK ENABLE
10	Unused (must = 0)	2	LOCAL
9	TRIG BKPT ENABLE	1	MESSAGE PENDING
8	REMOTE ENABLE	0	Unused (must = 0)
			(least significant bit)

NOTE: 1 = On, 0 = Off

The message format consists of four parts, the TYPE. VALUE. TEKSUM, and EOL (end-of-line). An example of an LED message causing the HALT, RUN, TRIG BKPT ENABLE and REMOTE ENABLE LED's to be illuminated is shown in Table 4-3.

Table 4-3. LED Message

ASCII	HEX	FORMAT	DESCRIPTION
!	21	TYPE	Defines LED message.
0 3 6 0	30 33 36 30	VALUE	Four ASCII hex digits representing the 16 bit value. The LEDs illuminated are: TRIG BKPT ENABLE, REMOTE ENABLE, HALT, and RUN.
0 9	30 39	TEKSUM	Two ASCII hex digits. An 8-bit arithmetic sum of the four hex digits of VALUE converted to two ASCII hex digits. Example: If VALUE = 01234H, then TEKSUM=AHEX (0+1+2+3) modulo 256 = 06H = 30 36 in ASCII. For this example TEKSUM = 0+3+6+0 = 09.
CR	OD	EOL	CR = Message terminator.

Keystroke

The keystrokes are encoded to an unsigned 8-bit integer. When the uSE is in the master mode, the key message is sent by the uSE each time the operator presses a function key. When in the slave mode and a key message is received, the key is acted upon. Table 4-4 gives actual values of keystrokes.

Table 4-4. Keystroke Functions

Key Type	Hex Value	Key Type	Hex Value	Key Type	Hex Value
PROCESSOR CONTROL		TRACE CONTROL		HEX KEYPAD	
RUN	23	TRIG 1	0B	<--	1F
STEP	22	BKPT	0A	-->	17
PROM/MEM	21	TRC QUAL	09	ENTER	27
SUBSEL	20	TERM DISP	08	"0"	07
RUN/DISP	1B	TRIG 2	03	"1"	06
RESET	1A	COUNT	02	"2"	05
RAM/MEM	19	TRC DISP	01	"3"	04
FILL	18	FLD SEL	00	"4"	0E
				"5"	0D
SUBFUNCTION CONTROL		DISPLAY SELECT		"6"	0C
				"7"	16
				"8"	15
DISABLE	32	I/O	47	"9"	14
ENABLE	31	MEMORY	4A	"A"	1E
BINARY	2B	MSG	49	"B"	1D
DECR	2A	BREAK	43	"C"	1C
INCR	29	REGISTER	42	"D"	26
				"E"	25
SPECIAL CONTROL		REMOTE LINK		"F"	24
				"X"	0F
TEST	2F	ATTN	36		
SEND	46	TERMINAL	2E		
MOVE	45	REMOTE	2D		

The structure of the keystroke message consists of four parts: TYPE, KEYVALUE, TEKSUM and EOL. An example of the keystroke message sending the REGISTER key is shown in table 4-5.

Table 4-5. Keystroke Message

ASCII	HEX	FORMAT	DESCRIPTION
#	23	TYPE	
4 2	34 32	KEYVALUE	Two ASCII hex digits. Send the register key.
0 6	30 36	TEKSUM	Two ASCII hex digits as defined in the LED message using the sum of the two digit KEY VALUE field. TEKSUM = 4 + 2 = 06
CR	OD	EOL	Message terminator.

Display Messages

When the uSE generates messages in the slave mode, they are displayed on the slave uSE and transmitted. In the master mode they are only transmitted. The structure of the display message function is:

BYTE	DIAGNOSTIC HEADER	DESCRIPTION
0	TYPE	22H (double quote).
1,2	MESSAGE LENGTH	Number of characters in message. Range = 1-20 (1-14H).
3,4	CURSOR CONTROL	Legal values are: 00H = cursor off. 80H = cursor on.
5,6	CURSOR POSITION	Range = 0-20 (00-14H).
7,8	DECIMAL POSITION	Range = 0-20 (00-14H).
9,10	TEKSUM	Two ASCII hex digits as defined in the LED message, using the sum of bytes 1 through 4 of the DISPLAY MESSAGE.
11	MESSAGE	1 to 20 modified ASCII characters - see appendix A.
. . .		
11+n	MESSAGE	
12+n	ASUM	One ASCII hex digit. ASUM computed as follows: Let ASUM = 40H - Sum (hex equilvalent of ASCII character in message (modulo 64)). If ASUM < 20H then add 40H to ASUM. If ASUM >= 20H use ASUM.
13+n, 14+n	EOL	End-of-Line, an ASCII carriage return (ODH).

n = 1 to 20

Table 4-6 gives an example of a display message.

Table 4-6. Display Message

ASCII	HEX	FORMAT	DESCRIPTION
"	22	TYPE	Defines display message.
0 F	30 46	MESSAGE LENGTH	Number of characters = 15.
0 0	30 30	CURSOR CONTROL	Cursor off.
0 0	30 30	CURSOR POSITION	No cursor position.
0 0	30 30	DECIMAL POSITION	No decimal position.
0 F	30 46	TEKSUM	Sum of MESSAGE LENGTH, CURSOR CONTROL, CURSOR POSITION and DECIMAL POSITION. TEKSUM = 0 + F + 0 + 0 + 0 + 0 + 0 + 0 = 0F.
Z 8 0 A D E M O V 2 . 1 	5A 38 30 41 20 44 45 4D 4F 20 56 32 2E 31 20	MESSAGE	ASCII Message is "Z80A DEMO V2.1".

Table 4-6. Display Message (continued)

ASCII	HEX	FORMAT	DESCRIPTION
Q	51	ASUM	Sum of MESSAGE = 5A+38+30+41+20+44+45+4D+4F+20+56+32+2E +31+20 = 36F = 2F (modulo 64) ASUM = 40-(2F) = 11, ASUM is < 20, so 40 must be added ASUM = 11+40 = 51
CR	OD	EOL	Message terminator.

Download Messages

The download message moves memory data from one system to another, and can be sent or received in any mode. The structure of the download message function is:

1. TYPE - slash (2FH).
2. LOCATION - Starting address (four ASCII hex digits). Range = 0-FFFFH.
3. BYTE COUNT - Number of data bytes to transfer (two ASCII hex digits). Range = 1-20H.
4. TEKSUM 1 - Sum of LOCATION and BYTE COUNT (two ASCII hex digits).
5. DATA - Data transferred (ASCII hex digits). Range = 2-40H.
6. TEKSUM 2 - Sum of DATA (two ASCII hex digits).
7. EOL - End-of-Line, ASCII carriage return (ODH) and optional line feed (OAH).

Table 4-7 is an example of a seven-byte message sent starting at address 0800.

Table 4-7. Download Message

ASCII	HEX	FORMAT	DESCRIPTION
/	2F	TYPE	Defines download message.
0 8 0 0	30 38 30 30	LOCATION	Four ASCII hex digits start loading data at location 0800.
0 7	30 37	BYTE COUNT	Two ASCII hex digits. Transfer seven bytes of data.
0 F	30 46	TEKSUM 1	Checksum of LOCATION and BYTE COUNT fields. TEKSUM 1 = 0 + 8 + 0 + 0 + 0 + 7 = 0F
3 C 3 2 0 9 0 8 C 3 0 0 0 8	33 43 33 32 30 39 30 38 43 33 30 30 30 38	DATA	The ASCII hex digits of memory transfer. When this data is received, the uSE displays: "LOAD IN PROGRESS."
3 C	33 43	TEKSUM 2	Checksum of DATA. TEKSUM 2 = 3 + C + 3 + 2 + 0 + 9 + 0 + 8 + C + 3 + 0 + 0 + 0 + 8 = 3C (modulo 256)
CR	0D	EOL	CR = Message terminator.

A message with a byte count of zero is the file terminator. Memory data and TEKSUM 2 are not part of the termination message. Table 4-8 shows an example.

Table 4-8. File Terminator Message

ASCII	HEX	FORMAT	DESCRIPTION
/	2F	TYPE	Defines download message.
0 8 0 0	30 38 30 30	LOCATION	Starting address is 0800H.
0 0	30 30	BYTE COUNT	No bytes transferred - end record.
0 8	30 38	TEKSUM	TEKSUM = 0 + 8 + 0 + 0 + 0 + 0 = 08
CR	OD	EOL	Message terminator.

After the file terminator message is received, the uSE displays:

```
LOAD COMPLETE
```

The data that was received by the download would be in memory as follows:

ADDRESS	DATA
0800	3C
0801	32
0802	09
0803	08
0804	C3
0805	00
0806	08

The starting address is stored in the program counter, the move address, and the send address.

ESTABLISHING uSE TO uSE COMMUNICATIONS

To establish communications between the uSE master and uSE slave, both operators must press the REMOTE key and select the desired options.

For the slave uSE, bits 7 and 6 of the remote option byte must equal 0 and 1 respectively. For the master uSE, bits 7 and 6 of the remote option byte must equal 1 and 0 respectively.

Both master and slave operators must select the same ACK/NAK, parity and baud rate. The communications link can be established over wire lines (see RS-232D specifications for distance), or using acoustic couplers (see respective data sets) and the phone lines. Once the options have been selected, both operators should press the ENABLE key.

In the following example, the master uSE controls the slave uSE over phone lines. The master downloads the program to the slave and then executes the program.

MASTER uSE

Operation	Display	Explanation
RESTART	xxxxxx READY 2708	xxxxxx = Emulator type. 2708 = PROM configuration.
REMOTE	REM OPTS = 96	Default options.
BINARY	REM OPTS = 10010110	Put in binary.
1000001, ENTER	REM OPTS = 10000001	Modify options to MASTER, ACK/NAK on, NO PARITY, 300 BAUD.
ENABLE	MASTER MODE	REMOTE ENABLE LED is illuminated.

SLAVE uSE

Operation	Display	Explanation
RESTART	xxxxxx READY 2708	xxxxxx = Emulator type 2708 = PROM configuration
REMOTE	REM OPTS = 96	Default option.
BINARY	REM OPTS = 10010110	Put in binary.
0100001, ENTER	REM OPTS = 01000001	Modify option to SLAVE, ACK/NAK ON, NO PARITY, 300 BAUD.
ENABLE	SLAVE MODE	REMOTE ENABLE LED is illuminated.

Both units are enabled, and are ready to talk to each other.

	MASTER uSE		SLAVE uSE
Operation	Display	Explanation	Display
ATTN	LOCAL MODE	Put master in local mode, LOCAL LED is illuminated to indicate local mode.	
SEND	SEND R 0000 0000	Get ready to send data to the slave.	
INCR	SEND P 0000 0000	Choose SEND PROM data from master to slave.	
2, ENTER	SEND P 2000 0000	SEND PROM data from address 2000 thru 27FF to slave.	
27FF, ENTER	SEND IN PROGRESS		LOAD IN PROGRESS

Operation	MASTER uSE Display	Explanation	SLAVE uSE Display
	SEND COMPLETE		LOAD COMPLETE
REMOTE	REM OPT = 81	Return to master mode. REMOTE and LOCAL LED's extinguished, remote link broken.	
ENABLE	MASTER MODE	Re-establish remote link. REMOTE ENABLE LED illuminated.	
RAM/MEM	DIAG RAM -1 = 2000		DIAG RAM -1 = 2000
ENABLE	Z80 DEMO V3.1		Z80 DEMO V3.1
RUN	PROM TEST	The master tested the user's system that the slave uSE was emulating.	PROM TEST
	RAM TEST		RAM TEST
	END OF ALL TESTS		END OF ALL TESTS

DISPLAY TERMINAL FUNCTIONS

When the display terminal function is enabled (via the TERMINAL key) then the display terminal can be used to display memory, registers, and accumulated trace data.

With the auto-step function enabled, the display terminal, can be used to display all registers and/or the last memory dump specified and/or I/O locations examined

Display Terminal Display of Memory

The Dump Memory function displays memory data on a display terminal, according to the current memory map. If the display terminal is enabled, then, when the MEMORY key is pressed, the Dump Memory function is invoked instead of the Examine Memory function and the uSE displays:

```
DUMP MEM xxxx yyyy
```

xxxx = starting address
yyyy = ending address

Dump Memory Subfunctions

HEXPAD - Allows the operator to modify the start and end addresses.

ENTER - Freezes the start and end addresses. When the ENTER key is pressed to freeze the start address, the end address may be modified. When the ENTER key is pressed to freeze the end address, transmission of memory display data to the display terminal begins. However, if the start address has not been modified and the ENTER key is pressed the default addresses are used and the transmission will be initiated.

Display Terminal Display

```
xxxx = yy yy yy yy yy yy yy yy yy yy yy yy yy yy yy yy zzzzzzzzzzzzzzzz
```

where: xxxx = memory address.
yy = memory data.
z...z = ASCII equivalence of the memory data (non-printable bytes are replaced with a period).

If the addresses entered are not modulo 16, then there will be short lines of data displayed at the display terminal, e.g., if the following were entered at the uSE,

```
DUMP MEM 1234 1256
```

then, the following could be displayed at the display terminal:

```
1234 = 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F             456789:;<=
1240 = 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F @ABCDEFGHIJKLMNO
1250 = 50 51 52 53 54 55 56                             PQRSTUV
```

Display Terminal Display of Registers

The dump registers function displays the contents of a processor's registers on the display terminal. When the display terminal is enabled, pressing the REGISTER key invokes the Examine Registers function, the Dump Registers function, and displays the register data last displayed. All subfunctions are allowed with this function. Pressing the ENTER key, when the display terminal is enabled, reinitiates data transmission to the display terminal. The format of the display terminal data is processor dependent. The output to the display terminal is in the same format as the uSE's display, but there is one line of display terminal register data for each three unique lines of uSE register data.

Examples of display terminal register dumps:

1. For the Z-80:

```
PN=0000 PL=0000 I=00    SP=0000 CC=00       RA=00 BC=00 00
DE=00 00 HL=00 00       IX=0000 IY=0000     CC'=00
RA'=00 BC'=00 00        DE=0000 HL'=00 00   IR=00 R=00 IM=0
IFF2=0 IFF1=0
```

2. For the 6800:

```
RA=00 RB=00             RX=0000 SP=0000     PN=0000 PL=0000 I=00
CC=00
```

Display Terminal Display of Trace Data

The Dump Trace function displays accumulated trace data on the display terminal. When the display terminal function is enabled, pressing the TRC DISP key invokes Dump Trace instead of the Singular Display Trace function displayed on the uSE. The display for this function is as follows:

```
DUMP TRACE xxx
```

xxx = Starting line number of the trace dump, which, when the TRACE key is pressed, is the line number of the last line of the trace data collected; this line number can be modified, see below.

Dump Trace Subfunctions

HEXPAD - Allows the operator to modify the line number where to start dumping trace data.

ENTER - Freezes the trace dump starting line number. If the line number is not modified before the ENTER key is pressed, the default line number is used. Pressing the ENTER key begins transmission of trace data to the display terminal. The data transmitted is a "column-title" line followed by the 23 lines of accumulated trace data, beginning at the line number specified.

Display Terminal

```
LINE ADRS DA BU EX
   n wwww xx yy zzzzzzzz
 n+1 wwww xx yy zzzzzzzz
 n+2 wwww xx yy zzzzzzzz

   .
   .
   .

n+22 wwww xx yy zzzzzzzz
```

where: LINE = column header for the trace line number, range is 000 - 127 in decimal format.
ADRS = column header for the value of the address lines.
DA = column header for the value of the data lines.
BU = column header for the bus activity.
EX = column header for the value of the external data probes.

n = line number (1-127 decimal).
wwww = value of the address lines (0-FFF).
xx = value of the data lines (0-FF).
yy = F-Instruction fetch (implied memory read).
MR = memory read.
MW = memory write.
IR = I/O read.
IW = I/O write.

zzzzzzzz = the binary representation of the value of the external data probes.

If the line number specified is less than 23 lines from the end of the trace buffer, then the trace data dump at the display terminal will be less than 23 lines long and will be terminated with the word END.

Furthermore, if the line number specified is greater than 23 lines from the end of the trace buffer, then the "xxx" in the uSE display will contain the line number of the next consecutive line number to be displayed. Thus, if the trace buffer is full, one could enter line number 000, and then repetitively press the ENTER key to cause the entire contents of the trace buffer to be displayed on the display terminal.

REAL-TIME TRACE

INTRODUCTION

Real-Time Trace (RTT) provides the following debugging features for hardware and software development:

- o Real-time storage and display of the last 128-selected bus transactions.
- o Dual breakpoint and trigger capability using bus, address and probe data with event comparison further conditioned by a pass count of n events and or a delay of n clocks.
- o Eight external probe states that are stored with the bus transaction data and can be selectively included in the breakpoint equation.
- o General-purpose counter can measure time, events, bus transactions, trace stores, or instruction fetches made between two points in a program, or between events.

Real-Time Trace Buffer

The following data is stored in the Real-Time Trace (RTT) buffer during selected events:

- o Contents of the 16-bit address bus
- o Contents of the 8-bit data bus (software displays only pertinent data)
- o Value of the 8-bit external probe data
- o Type of operation: read, write, memory, I/O, or instruction fetch

Dual Breakpoint and Trigger Circuitry

RTT contains a set of event comparators that compare operator stated requirements to bus information. For an event to be recognized, the requirements must be met simultaneously. The requirements may be any or all of the following, because they may be specified in the AND function:

- o 16-Bit Address equal, less than or equal, greater than or equal
- o 8-Bit Data equal, less than or equal, greater than or equal
- o 8-Test Probe Bits compare (1 0, or X (don't care)
- o Bus transaction type

The command is cumulative, allowing changes or additions to be entered easily without respecifying the entire function.

EVENT DEFINITION

The ability to define two distinct system "events" (T1 and T2) is at the heart of the RTT. The following keystroke operations will demonstrate the various options available to the user in defining a particular event.

It describes the event definition equation in terms of:

- o Address Bus State
- o Bus Activity Status
- o External Events
- o Data Bus State

In addition, the modifications available by pass count and delay are listed.

Delay and Pass Conditioning

Once an event is recognized it may immediately set a trigger or initiate a breakpoint, or it may be conditioned by the pass count. The pass count register can be set to count up to 65,535 events before a trigger or break is generated. This feature facilitates debugging of programs that contain loops.

The capability to delay a trigger or a break from an event can also be helpful. For example, a delay of 64 trace stores allows centering of the trace display relative to the event. A delay of from 3 to 65,535 clocks may be specified. The clock is operator selected by the COUNT key and may consist of events, instruction fetches, bus cycles, trace stores, microseconds, or milliseconds.

The capability to specify pass and delay counts for trigger 1 and trigger 2 is included in the TRIG 1 and TRIG 2 keys. If no pass or delay count is specified, they default to OFF. A Pass Count of 0 or 1 is defined as equivalent to No Pass Count (OFF). If both pass and delay are selected, the pass count must first be satisfied, then the delay period starts.

Breakpoint and Trigger Modes

After a trigger condition is specified by the TRIG keys, a breakpoint on that trigger is specified by the BKPT key. The mode options allow further versatility in utilizing the two trigger circuits. For example, Limit mode is useful for displaying trace information between selected addresses. In limit mode EVT 1 is further required to meet the EVT 2 requirements. This allows breakpoint on events between certain selected addresses.

Real-Time Trace Probe

An active data acquisition probe with ten test clips provides the real-time trace with external data gathering capability (see figure 5-1). The ten probes consist of eight acquisition probes that attach to test points, a clock probe that may be connected to an external clock source and a ground.

Figure 5-1. External Test Clips Probe

The trace probe assembly consists of ten individual probe cables contained in a probe pod. It attaches to the uSE back panel with a discrete wire cable via a 15-pin D connector (see figure 5-2).

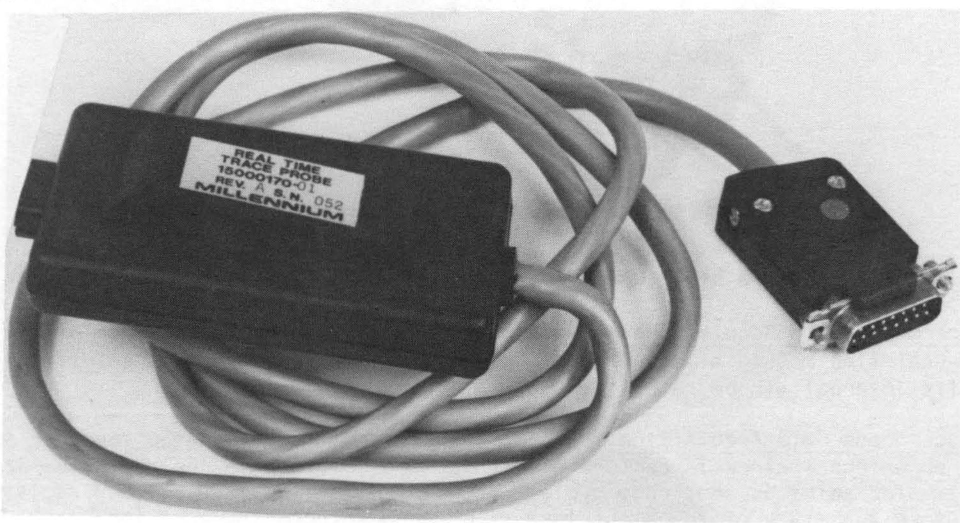


Figure 5-2. External Test Probe Connector

Data Acquisition

The data from the eight test clips is stored with the bus transaction data and may be displayed with the TRC DISP key. The test clip information can also be used in the definition of either or both events. Each bit can be specified as 1, 0, or don't care.

Three modes of operation are available with the trace probe:

- o Synchronous data acquisition
- o Asynchronous data acquisition
- o Latched mode

The three types of operation and external clock polarity are operator selectable by externally accessible switches located in the trace pod (see figure 5-3).

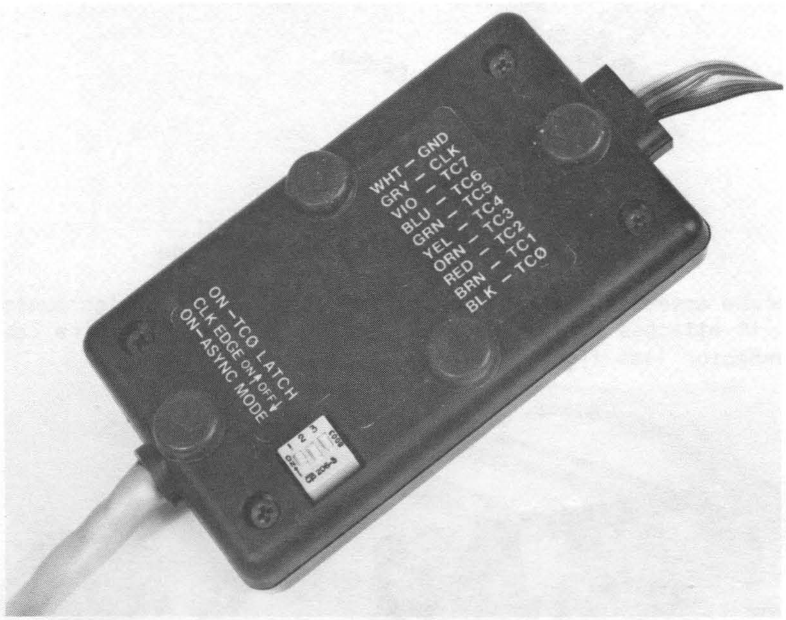


Figure 5-3. Operation and Clock Polarity Switches

Synchronous Data Acquisition. Synchronous operation presents test chip data to the real-time module at all times. Data is then captured by the real-time trace at its internal strobe rate.

Asynchronous Data Acquisition. Asynchronous operation allows the operator to use an external clock to capture events for presentation to the real-time trace at a later point in the trace cycle. An octal flip-flop is used as a register to store the state of the eight test clips on a selected edge of the clock. The data presented will be the data that was stored prior to sampling by the real-time trace.

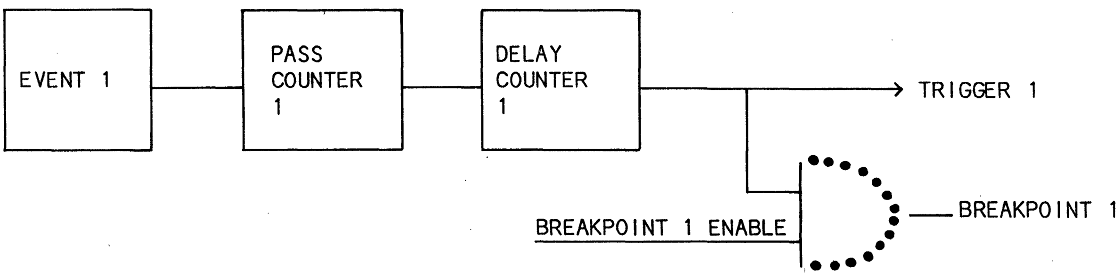
Latched Mode. The octal register is also used in the latched mode, but it is clocked by the internal strobe pulse of the real-time trace. Data bit 0 of the test clip is monitored constantly and compared with the data presented to the real-time trace on the previous cycle. Any state change will be presented at the next sampling period. This mode of operation allows the detection of a pulse within the trace window.

General-Purpose Counter

The general-purpose counter provides a method of measuring real-time or number of clocks between points in a program or between events. The operator specifies the clock from one of the count options. The counter is set to zero when the RUN key is pressed. When a break occurs, the count value may be displayed with the COUNT key.

EVENTS AND TRIGGERS

A distinction should be made between the concepts of events and triggers. An event occurs when a certain combination of address value, data value, external test clips value, and bus operation type occur simutaneously. A typical event will be defined as one or more of the values or bus operation types being in the off or don't care state, but the event will occur only when every item is simutaneously satisfied. An event can contribute to the production of its corresponding trigger by further qualification of pass count and delay count. The pass counter (which may be turned off) will determine when its corresponding event has occurred the required number of times. On satisfaction of the pass count, the delay counter (which may also be off) will count a given number of specified units, and then cause a trigger to occur. If the operator has further requested a break to occur on the occurence of the trigger, the slave CPU will be stopped. The sequence can be shown as follows for trigger 1:



Independent Mode

In the Independent mode of operation, the trace buffer contains the last 128-qualified cycles executed prior to encountering the breakpoint, whichever program path is taken. The event definitions do not interact. The two triggers and their corresponding events are shown in figure 5-4.

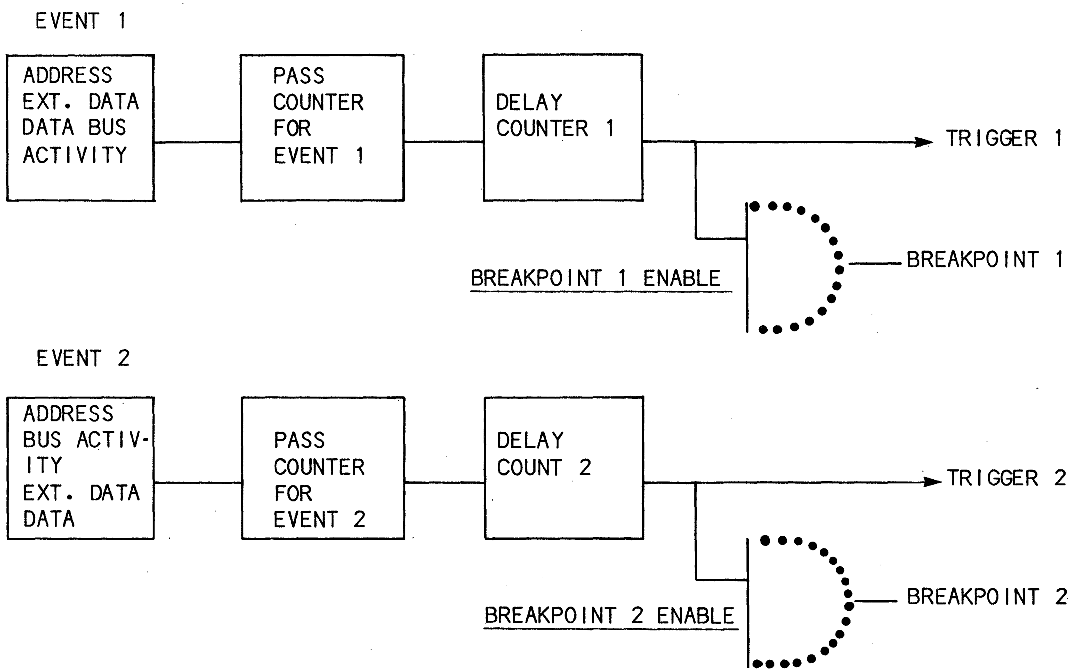


Figure 5-4. Independent Mode

Limit Mode

This mode allows the address bus conditions that define a breakpoint to be a logic AND of the address bus conditions for T1 and T2. This allows a closed range of addresses to be specified. (See figure 5-5.)

In the Limit mode of operation the events and triggers are configured as follows:

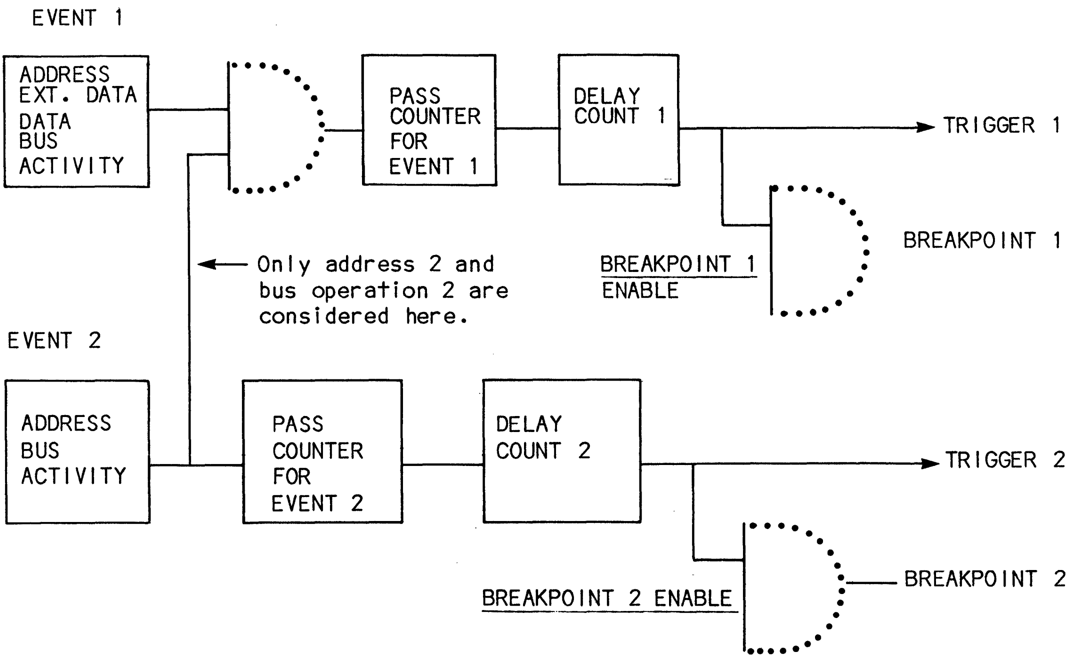


Figure 5-5. Limit Mode

In the Limit mode it should be noted that when event 1 and event 2 occur simultaneously to produce the input to pass counter 1, only the address and bus operation qualifiers for event 2 are considered. Trigger 1 should be used if the user desires to break on the simultaneous occurrence of event 1 and event 2. Trigger 2 will occur independently whenever its individual conditions are satisfied.

Arm Mode

In this mode, the breakpoint occurs at T2, but only after T1 has been encountered (see figure 5-6). That is, if T2 occurs prior to T1, no break is made. Also, if T2 is encountered a number of times, the break does not occur until the encounter after T1. The best example is having T2 in a subroutine called from a number of places, but not having a break occur until it is called from that part of the program that contains T1.

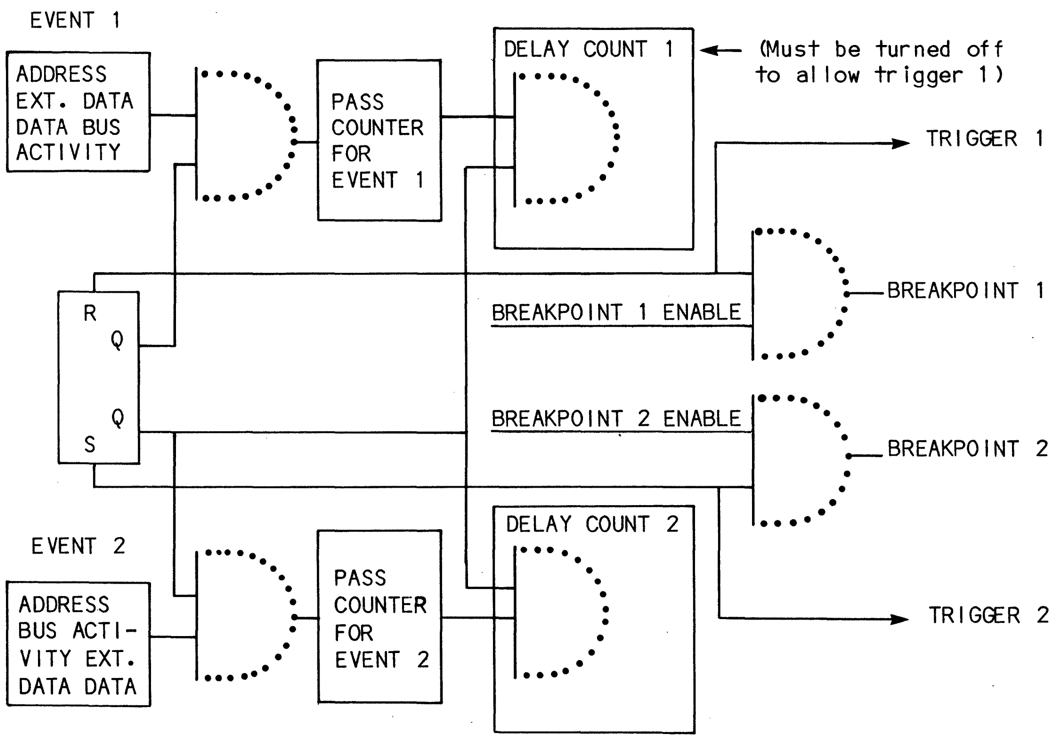


Figure 5-6. Arm Mode

In the Arm mode, trigger 1 is initially enabled. When trigger 1 occurs trigger 2 is enabled and trigger 1 is disabled. When trigger 2 occurs, trigger 1 is enabled and trigger 2 is disabled. This enabling/disabling sequence occurs indefinitely.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                           *
*                          NOTE                             *
*                                                           *
*  The general-purpose counter and the two delay counters   *
*  will not run when trigger 2 is disabled.  This implies   *
*  that delay counter 1 cannot be used in the Arm (or FRZ)  *
*  mode to define trigger 1 since the counter will not run  *
*  until trigger 2 is enabled by trigger 1.                 *
*                                                           *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

FRZ Mode

In this mode, the trace is frozen when T1 is encountered (see figure 5-7). This means that no new information is stored in the buffer, but the program is allowed to continue until T2 is encountered. The FRZ (Freeze) mode is identical to the Arm mode except that trace buffer stores are prevented between the occurrence of trigger 1 and trigger 2.

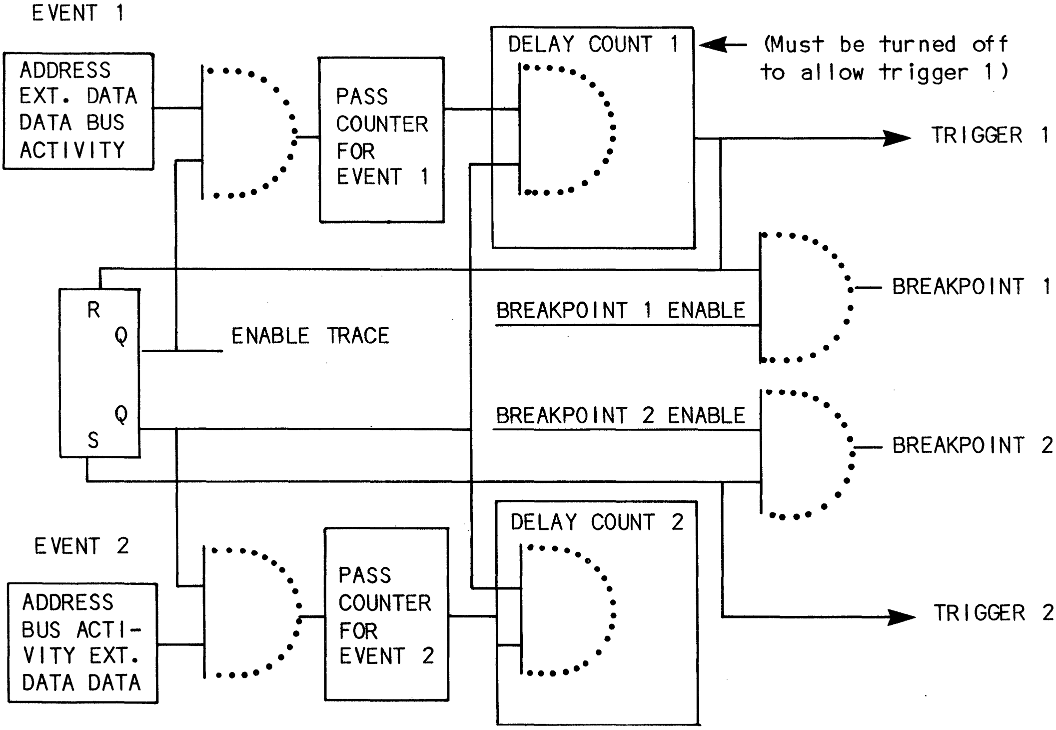


Figure 5-7. FRZ Mode

REAL-TIME TRACE LIMITATIONS

1. If the specifications are used to define event 2 for a high-speed processor (e.g. a 4 MHz Z-80A or a 5 MHz 8085A) a break specified on trigger 2 may not occur until after an additional instruction has been executed.

2. A delay counter value of 1 or 2 is illegal.

3. The maximum general-purpose counter value is 65534 (2^{16}-2).

REAL-TIME TRACE FUNCTION KEYS

TRIG 1 Key

The TRIG 1 key defines the trigger equations. Table 5-1 defines the parameters and options of the TRIG 1 key. When all elements are satisfied, the appropriate trigger pulse is provided on the uSE front panel trigger jack. The event equations can also be used in conjunction with the BKPT function to terminate the run condition.

Parameters for the TRIG 1 key are selected using the FLD SEL key. Parameter fields (i.e., address, data, etc.) may contain more than one option. Options within a parameter field are selected using the INCR key as shown in figure 5-8.

After the uSE is initialized, the first time a TRIG key is pressed it will display the address parameter. Each time it is pressed after that, the last parameter displayed will be displayed again. Each parameter field remains set to the last operator selection.

Table 5-1. TRIG 1 Parameters and Options

Parameter	Display	Explanation
Address	T1 EVT ADDR=OFF	Trigger 1 off (default).
	T1 EVT ADDR= xxxx	Hex bus address xxxx=0000-FFFF.
	T1 EVT ADDR=>xxxx	Hex bus address xxxx=0000-FFFF.
	T1 EVT ADDR=<xxxx	Hex bus address xxxx=0000-FFFF.

Table 5-1. TRIG 1 Parameters and Options (continued)

Parameter	Display	Explanation
Bus	T1 EVT BUS=ALL	All bus activity (default).
	T1 EVT BUS=FTCH	All instruction fetch.
	T1 EVT BUS=MEM RD/WR	All memory accesses.
	T1 EVT BUS=I/O RD/WR	All I/O accesses.
	T1 EVT BUS=ALL RD	All bus reads.
	T1 EVT BUS=ALL WRT	All bus writes.
	T1 EVT BUS=MEM RD	All memory reads.
	T1 EVT BUS=MEM WRT	All memory writes.
	T1 EVT BUS=I/O RD	All I/O reads.
	T1 EVT BUS=I/O WRT	All I/O writes.

Table 5-1. TRIG 1 Parameters and Options (continued)

Parameter	Display	Explanation
External *	T1 EVT EXT= bbbbbbbb	Binary External probe data. bbbbbbbb = 00000000 11111111 or xxxxxxxx (x = don't care)
Data	T1 EVT DATA=OFF	Trigger 1 off (default).
	T1 EVT DATA=xx	Hex bus data xx = 00-FF.
	T1 EVT DATA=xx	Hex bus data xx = 00-FF.
	T1 EVT DATA=xx	Hex bus data xx = 00-FF.
Pass	T1 PASS=OFF	Pass off (default).
	T1 PASS=ddddd EVT1	Decimal pass count ddddd = 00000-65534.

* External data can be applied to the equation by using the 8-bit external probe input. Data is sampled on the external probes synchronously to bus activity.

Due to timing constraints on the real-time trace board, it may not be possible, in certain cases, to stop the emulator processor exactly on the instruction causing a break. The break may occur on the next instruction in the following circumstances:

1. A high-speed processor is in use (such as a Z-80A at 4 MHz or an 8085A at 5 MHz).

2. Data and/or test clip values are specified for event 2.

Table 5-1. TRIG 1 Parameters and Options (continued)

Parameter	Display	Explanation
Delay	T1 DELAY=OFF	Delay off (default).
	T1 DELAY=xxxxx EVT1	Delay xxxx T1 events. xxxxx=00000-65534.
	T1 DELAY=xxxxx EVT2	Delay xxxxx T2 events. xxxxx=00000-65534.
	T1 DELAY=xxxxx TRC	Delay xxxxx traces. xxxxx=00000-65534.
	T1 DELAY=xxxxx uSEC	Delay xxxxx microseconds. xxxxx=00000-65534.
	T1 DELAY=xxxxx MSEC	Delay xxxxx milliseconds. xxxxx=00000-65534.
	T1 DELAY=xxxxx FTCH	Delay xxxxx fetches. xxxxx=00000-65534.
	T1 DELAY=xxxxx MCYC	Delay xxxxx machine cycles. xxxxx=00000-65534.

Table 5-1. TRIG 1 Parameters and Options (continued)

Parameter	Display	Explanation
Mode	T1 EVT MODE=IND	Independent triggers 1 & 2.
	T1 EVT MODE=LIMIT	Trigger 1 depends on trigger 2.
	T1 EVT MODE=ARM	Trigger 2 is armed by trigger 1.
	T1 EVT MODE=FRZ	Same as arm, but trace stops on trigger 1.

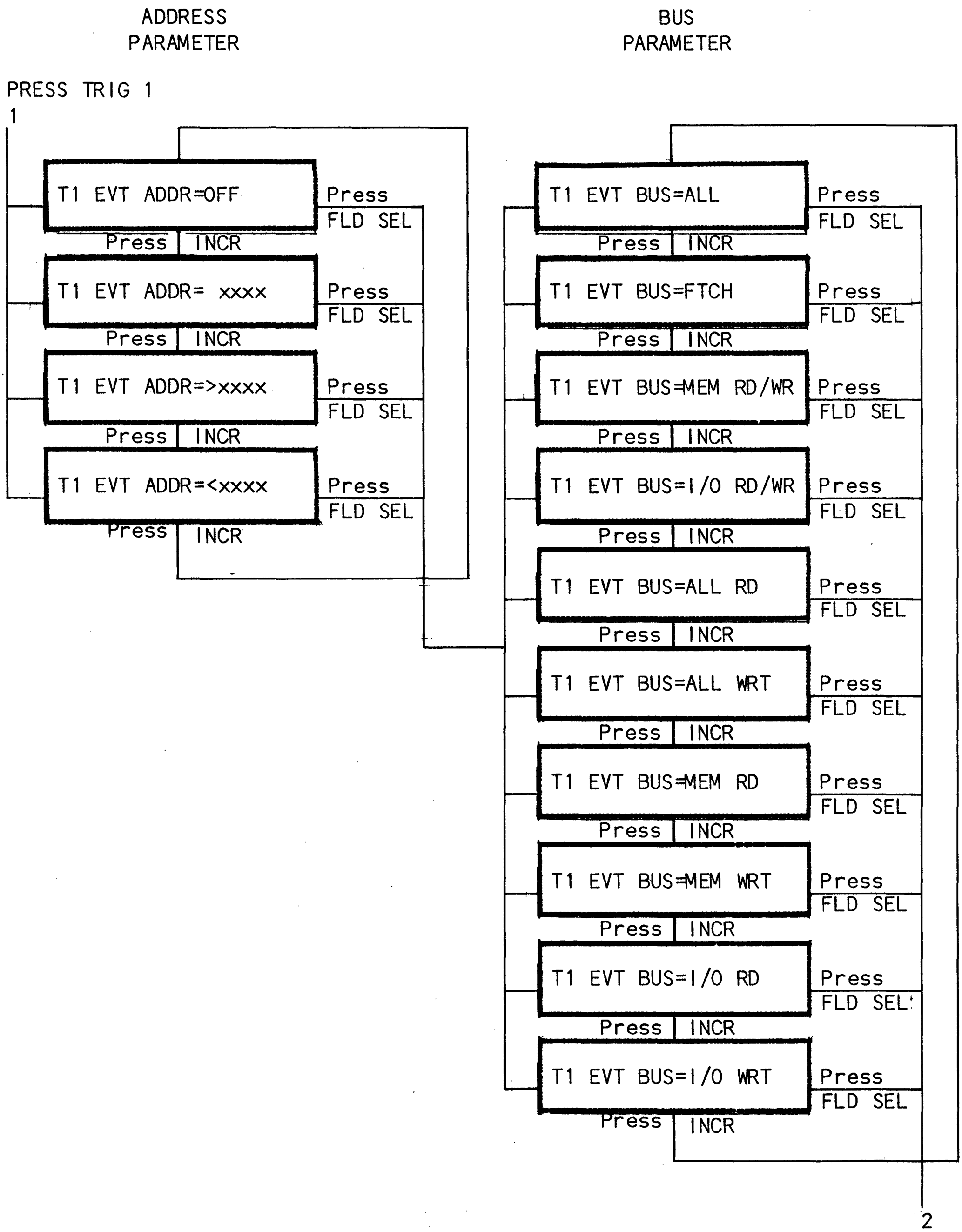


Figure 5-8. TRIG 1 Key Selection Sequence

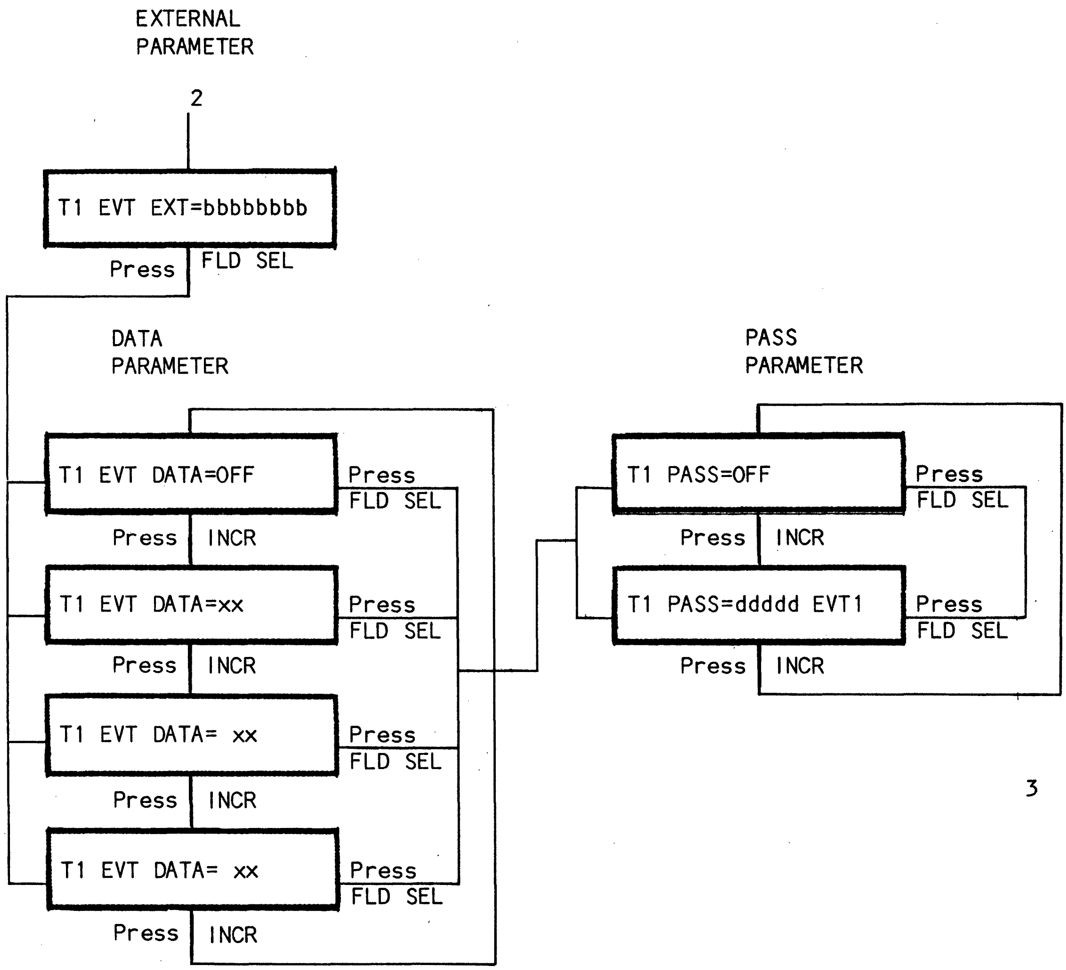


Figure 5-8. TRIG 1 Key Selection Sequence (continued)

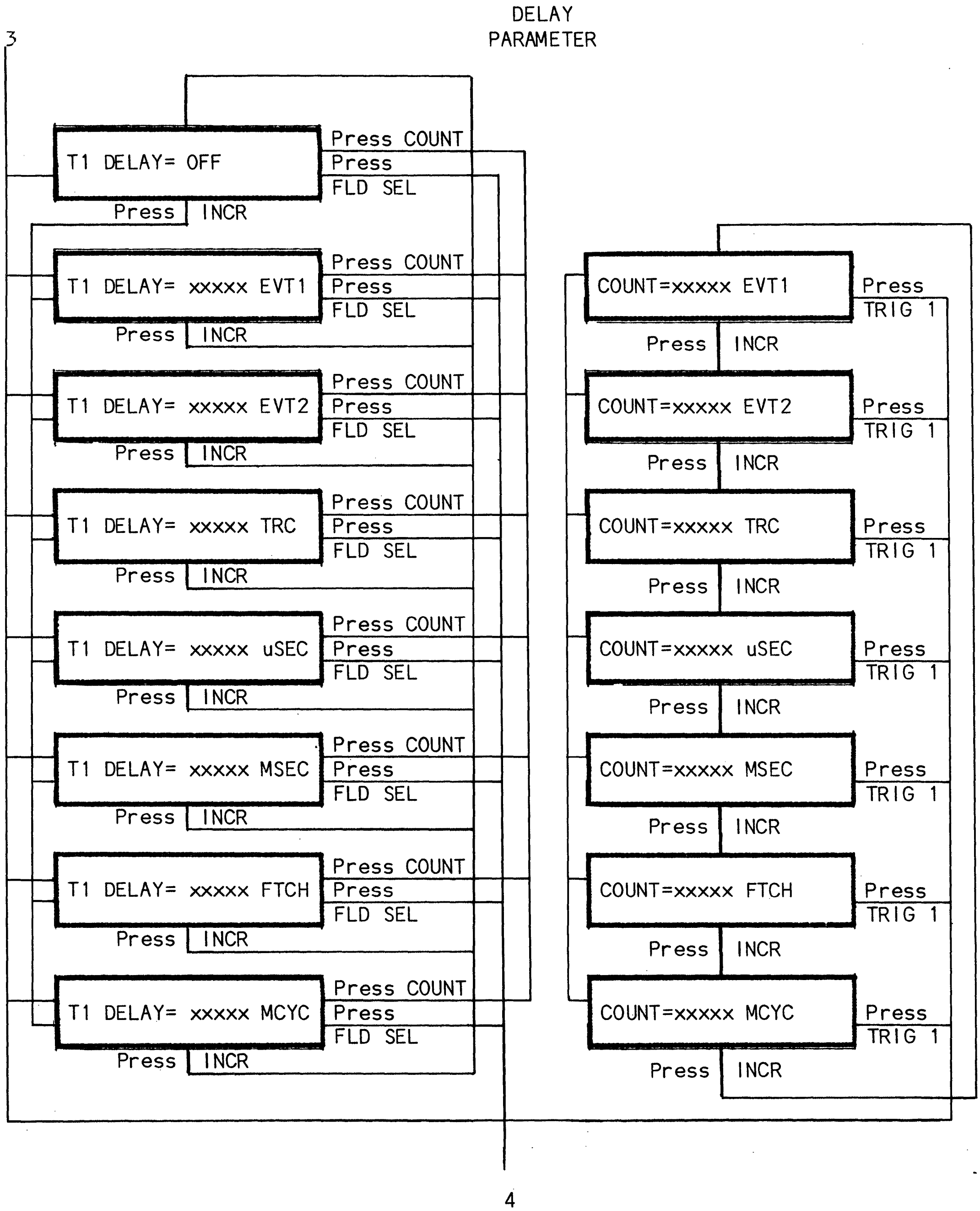


Figure 5-8. TRIG 1 Key Selection Sequence (continued)

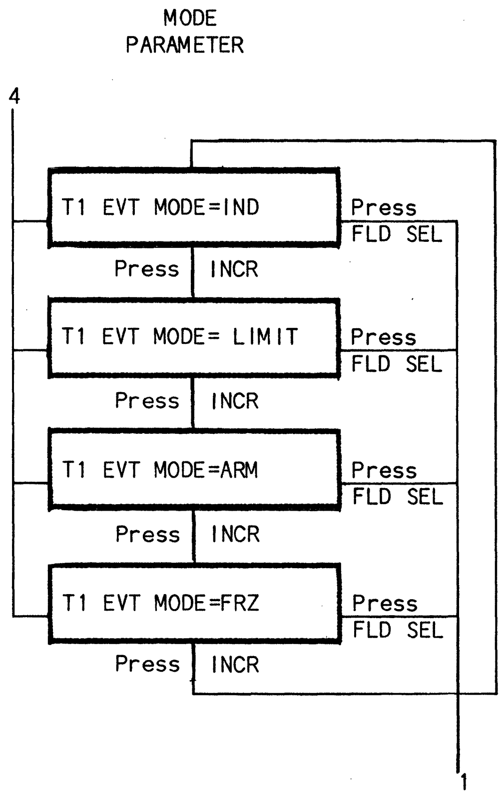


Figure 5-8. TRIG 1 Key Selection Sequence (continued)

RIG 2 KEY

he TRIG 2 key operates the same as the TRIG 1 key except for the Mode arameter. Table 5-2 shows the TRIG 2 parameters and options. The selection of arameters and options is shown in figure 5-9.

Table 5-2. TRIG 2 Parameters and Options

Parameter	Display	Explanation
Address	T2 EVT ADDR=OFF	Trigger 2 off (default).
	T2 EVT ADDR= xxxx	Hex bus address xxxx=0000-FFFF.
	T2 EVT ADDR=>xxxx	Hex bus address xxxx=0000-FFFF.
	T2 EVT ADDR=<xxxx	Hex bus address xxxx=0000-FFFF.
Bus	T2 EVT BUS=ALL	All bus activity (default).
	T2 EVT BUS=FTCH	All instruction fetch.
	T2 EVT BUS=MEM RD/WR	All memory accesses.
	T2 EVT BUS=I/O RD/WR	All I/O accesses.
	T2 EVT BUS=ALL RD	All bus reads.
	T2 EVT BUS=ALL WRT	All bus writes.

Table 5-2. TRIG 2 Parameters and Options (continued)

Parameter	Display	Explanation
	T2 EVT BUS=MEM RD	All memory reads.
	T2 EVT BUS=MEM WRT	All memory writes.
	T2 EVT BUS=I/O RD	All I/O reads.
	T2 EVT BUS=I/O WRT	All I/O writes.
External *	T2 EVT EXT= bbbbbbbb	Binary External probe data. bbbbbbbb = 00000000-11111111, or xxxxxxxx (x = don't care)
Data	T2 EVT DATA=OFF	Trigger 2 off (default).
	T2 EVT DATA=xx	Hex bus data xx = 00-FF.
	T2 EVT DATA=xx	Hex bus data xx = 00-FF.
	T2 EVT DATA=xx	Hex bus data xx = 00-FF.

* External data can be applied to the equation by using the 8-bit external probe input. Data is sampled on the external probes synchronously to bus activity.

Table 5-2. TRIG 2 Parameters and Options (continued)

Parameter	Display	Explanation
Pass	T2 PASS=OFF	Pass off (default).
	T2 PASS=ddddd EVT1	Decimal pass count ddddd = 00000-65534.
Delay	T2 DELAY=OFF	Delay off (default).
	T2 DELAY=xxxxx EVT1	Delay xxxx T1 events. xxxxx=00000-65534.
	T2 DELAY=xxxxx EVT2	Delay xxxxx T2 events. xxxxx=00000-65534.
	T2 DELAY=xxxxx TRC	Delay xxxxx traces. xxxxx=00000-65534.
	T2 DELAY=xxxxx uSEC	Delay xxxxx microseconds. xxxxx=00000-65534.
	T2 DELAY=xxxxx MSEC	Delay xxxxx milliseconds. xxxxx=00000-65534.
	T2 DELAY=xxxxx FTCH	Delay xxxxx fetches. xxxxx=00000-65534.
	T2 DELAY=xxxxx MCYC	Delay xxxxx machine cycles. xxxxx=00000-65534.

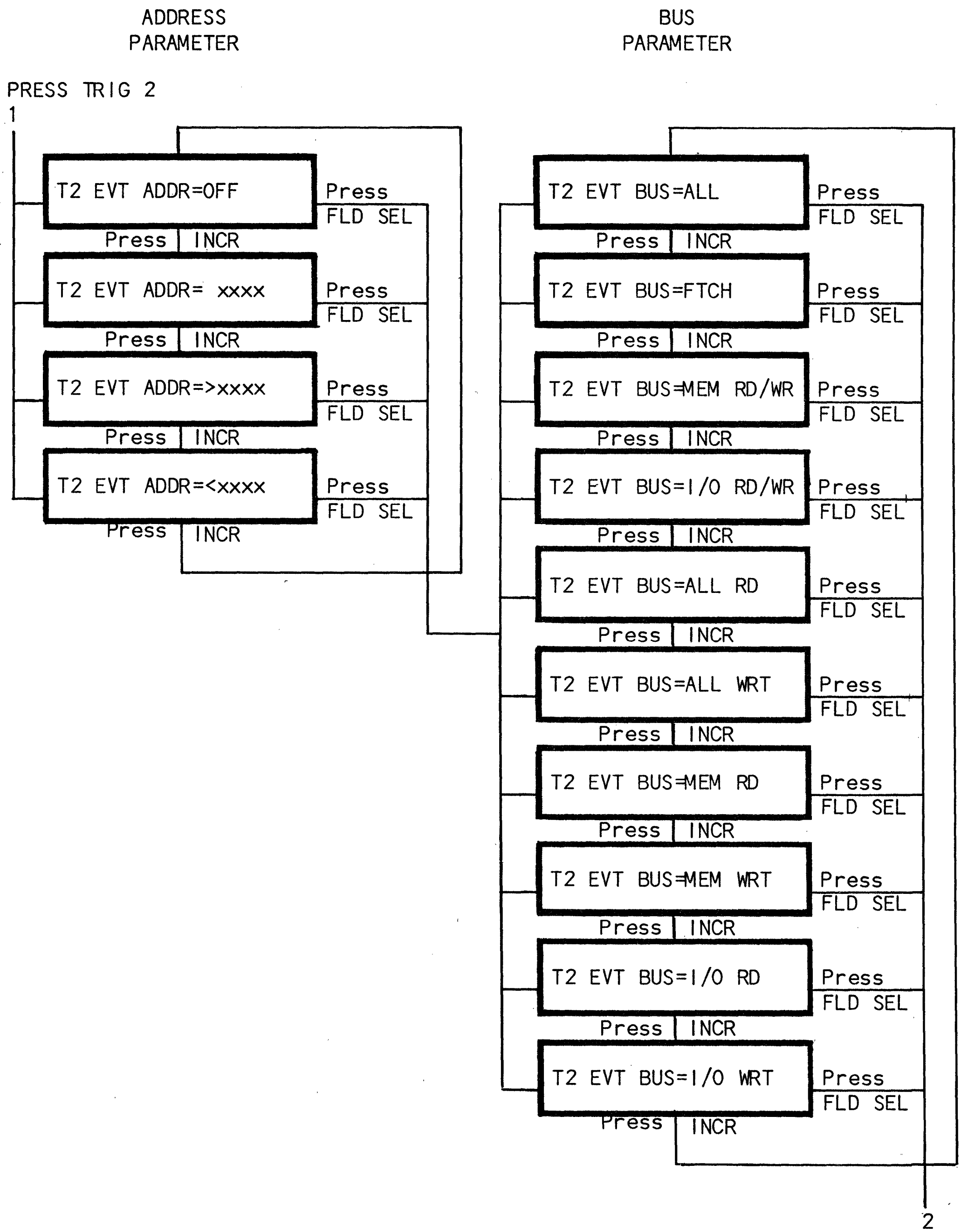


Figure 5-9. TRIG 2 Key Delay Parameter Selection Sequence

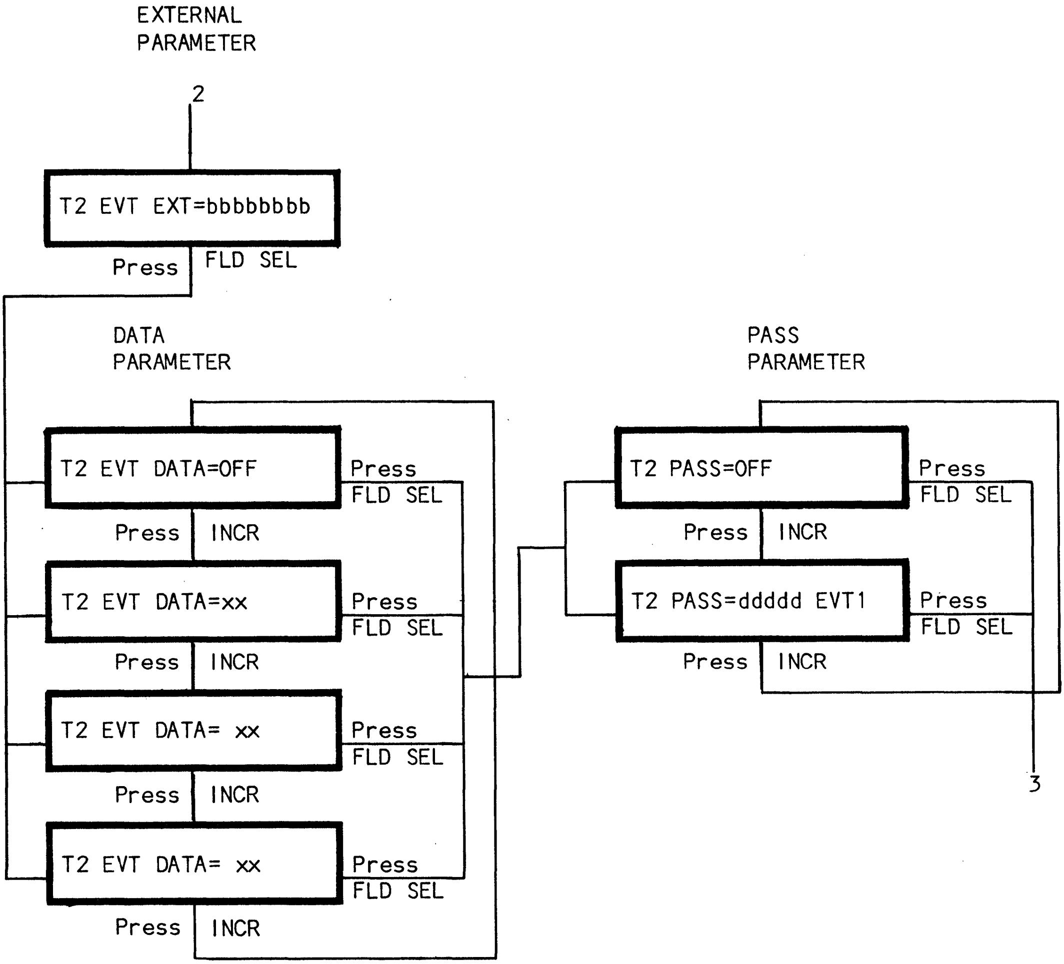


Figure 5-9. TRIG 2 Key Delay Parameter Selection Sequence (continued)

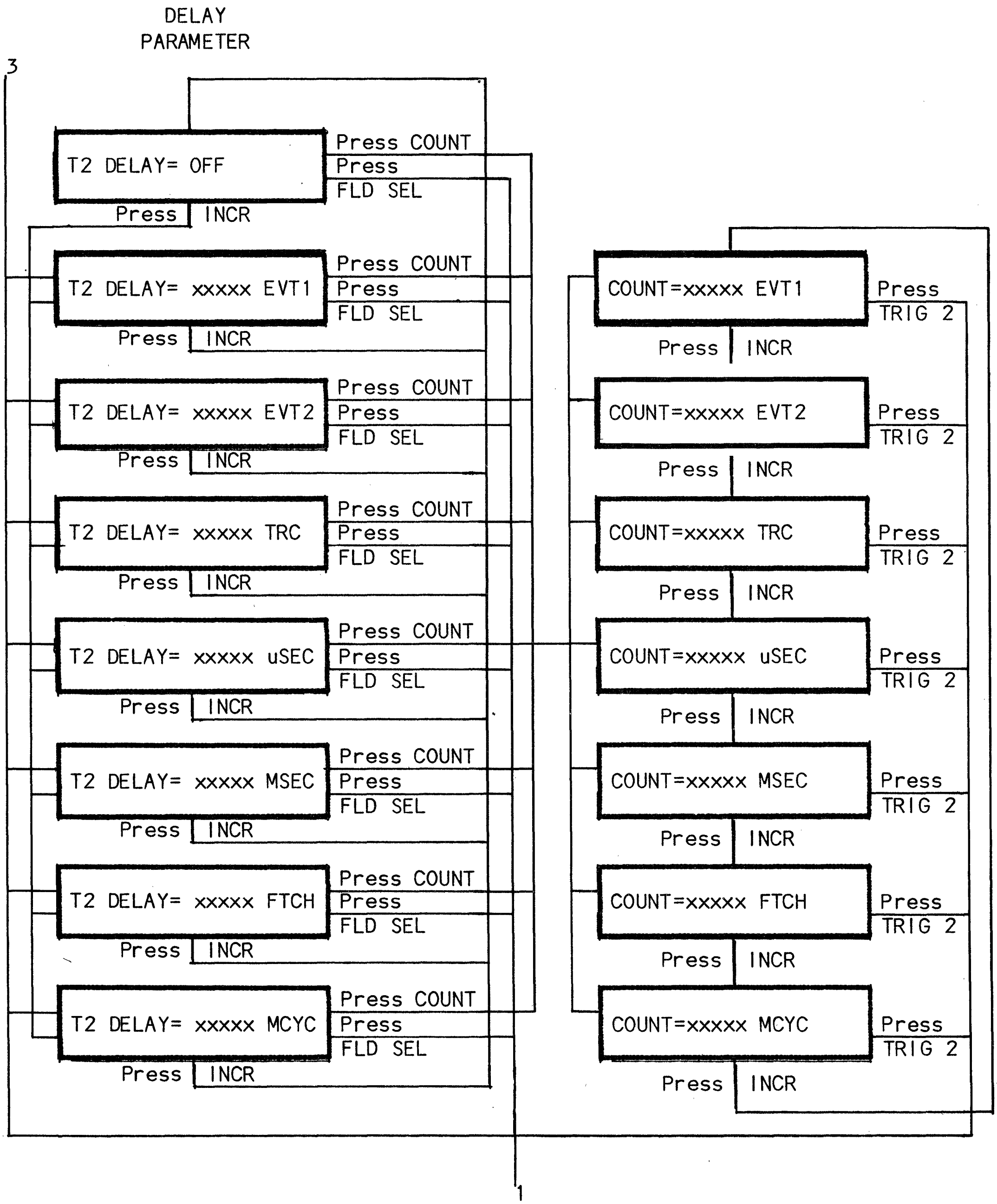


Figure 5-9. TRIG 2 Key Delay Parameter Selection Sequence (continued)

BKPT Key

The BKPT Key enables the user to set break points determined by the TRIGGER 1 and TRIGGER 2 equations. Options and Mode are selected by using the FLD SEL key. Parameters of the BKPT function are selected using the INCR key to step to the desired parameter. The selected parameter remains active until changed by operator selection or the RESTART key. Table 5-3 shows the parameters and options of the BKPT key.

Table 5-3. BKPT Parameters and Options

Parameter	Display	Explanation
Sel	BKPT SEL=OFF	No active breakpoint set (default).
	BKPT SEL=T1	Breakpoint when trigger 1 occurs.
	BKPT SEL=T2	Breakpoint when trigger 2 occurs.
	BKPT SEL=T1 or T2	Breakpoint when trigger 1 or 2 occurs.
Mode	BKPT MODE=STOP	Pause emulator after breakpoint (default).
	BKPT MODE=CONT	Continue emulator after breakpoint.

If a breakpoint occurs, the uSE displays:

BREAK AT xxxx	xxxx = hex address of the breakpoint, and, if the display terminal is enabled, the data selected via the TERM DISP key will be output to the display terminal.

TRC QUAL Key

The TRC QUAL key allows the user to select the type of transactions that are stored in the real-time trace buffer. The trace buffer always stores selected information when the Emulator is operating. The only exception is the freeze trace mode, commanded by the TRIG 1 key. In this case the trace storage is stopped after the trigger 1 requirements are met. The type of activity to be captured is selected by using the INCR key to position to the desired activity type. The real time trace buffer maintains up to 128 current TRACE entries. Operator selections remain active until modified by the operator or Restart key. The options of the TRC QUAL key are shown in table 5-4.

Table 5-4. TRC QUAL Options

Display	Explanation
TRC QUAL=ALL	All bus activity (default).
TRC QUAL=FTCH	All instruction fetch.
TRC QUAL=ALL RD	All bus reads.
TRC QUAL=ALL WT	All bus writes.
TRC QUAL=MEM RD/WRT	All memory accesses.
TRC QUAL=MEM RD	All memory reads.
TRC QUAL=MEM WRT	All memory writes.

Table 5-4. TRC QUAL Options (continued)

Display	Explanation
TRC QUAL=I/O RD/WRT	All I/O accesses.
TRC QUAL=I/O RD	All I/O reads.
TRC QUAL=I/O WRT	All I/O writes.

TRC DISP Key

The TRC DISP key allows the user to display the contents of the real-time trace buffer. Transactions are always displayed starting with the most recent. The TRACE display format is as follows:

ddd xxxx yy zz aa

dd = Trace line number (decimal).
xxxx = Value of the address lines (hex).
yy = Value of the data lines (hex).
zz = Value of the external data probes (hex). This value can be displayed in binary by pressing the FLD SEL key.
aa = Bus Operation
- F=Instruction fetch (implied memory read)
- MR=memory read
- MW=memory write
- IR=I/O read
- IW=I/O write

The initial trace displays the last trace buffer entry. The INCR and DECR keys are used to scroll through the trace lines. The trace line field (ddd) is alterable from the hexpad to position to any trace line. Partially filled trace buffers are indicated by a count (ddd) 127. Incrementing the count beyond the last trace entry displays a line number and a blank trace line. (Refer to chapter 4, COMMUNICATIONS, DISPLAY TERMINAL FUNCTIONS and DISPLAY TERMINAL DISPLAY OF TRACE DATA.)

COUNT Key

The COUNT key allows the counting of the various parameters up to a maximum value of 65,534 units. Maximum counts are an indication of counter overflow and, therefore, cannot be viewed as valid measurements. Depressing the COUNT key displays the current count value specified in the selected units. The general purpose counter is reset by depressing the DISABLE key while in the count function. The INCR key is used to step through the count parameters. A new count parameter selection will also reset the counter. The COUNT key options are shown in table 5-5.

NOTE: Pressing the COUNT key also allows the count betweenm T1 and T2 when in the Arm mode of the FRZ mode.

Table 5-5. COUNT Options

Display	Explanation
COUNT=xxxxx EVT1	Count event 1 occurrences.
COUNT=xxxxx EVT2	Count event 2 occurrences.
COUNT=xxxxx TRC	Counts real-time trace entries.
COUNT=xxxxx uSEC	Count microseconds.
COUNT=xxxxx MSEC	Count milliseconds.
COUNT=xxxxx FTCH	Count instruction cycles.
COUNT=xxxxx MCYC	Count machine cycles.

INTRODUCTION

This section describes in detail the formats of the memory used to execute diagnostic programs on a UUT. The formats are the same as those used on the Micro-System Analyzer. Functional diagnostics written in this format can be executed using either a uSE or uSA.

The uSE uses up to 4K bytes of PROM and 256 bytes of RAM (Shadow RAM) to execute the diagnostic programs that exercise the UUT. The uSE executes diagnostics on the UUT from the diagnostic program resident in the PROMs plugged into the front of the uSE or from the program in the UUT memory. Because the memory of the UUT can occupy any address locations, the diagnostic program could overlay the UUT memory or it could extend it. For example, the UUT memory could extend from 0000 to 2FFF and the diagnostic PROM could begin at 2000. In this case, the diagnostic program would overlay the UUT memory. If, however, the UUT memory was from 0000 to 1FFF and the diagnostic program began at 2000, the diagnostic program would extend the program memory space.

In addition to the mapping of the diagnostic program to UUT memory, the uSE contains a 256-byte block of RAM, called Shadow RAM, that dynamically overlays the control information of the diagnostic program in PROMs. The diagnostic program uses the Shadow RAM during execution. Figure 6-1 illustrates how the diagnostic program can overlay UUT memory, and how the Shadow RAM overlays the diagnostic program.

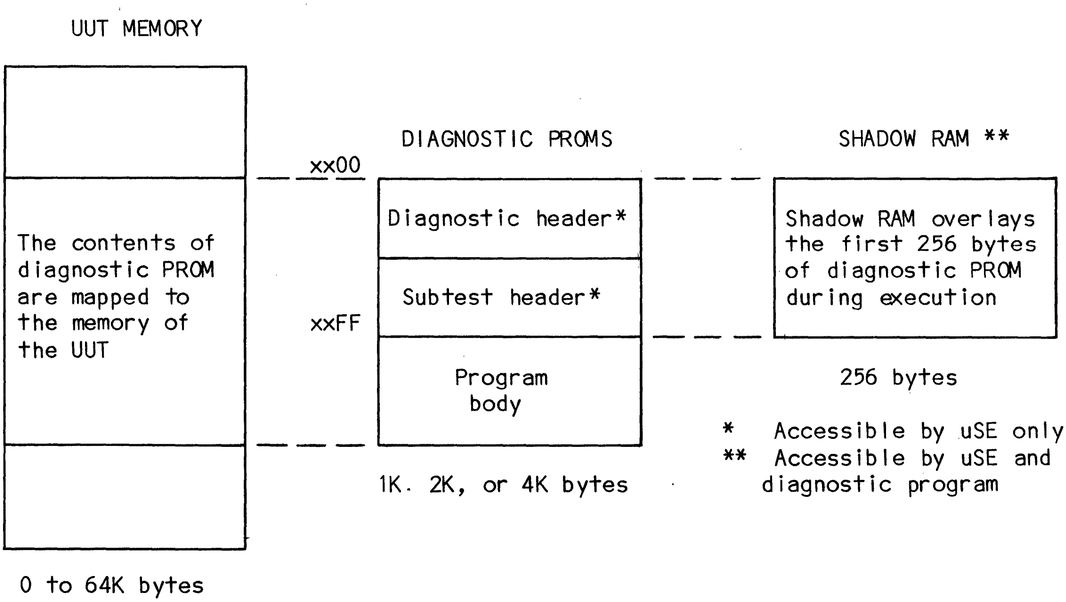


Figure 6-1. Program Memories

DIAGNOSTIC PROM

The diagnostic program in the PROM (or pair of PROMs) consists of control information and a program body as shown in figure 6-2.

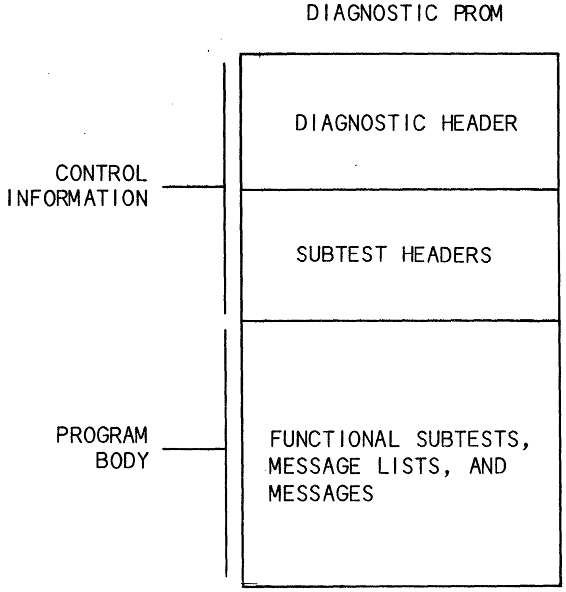


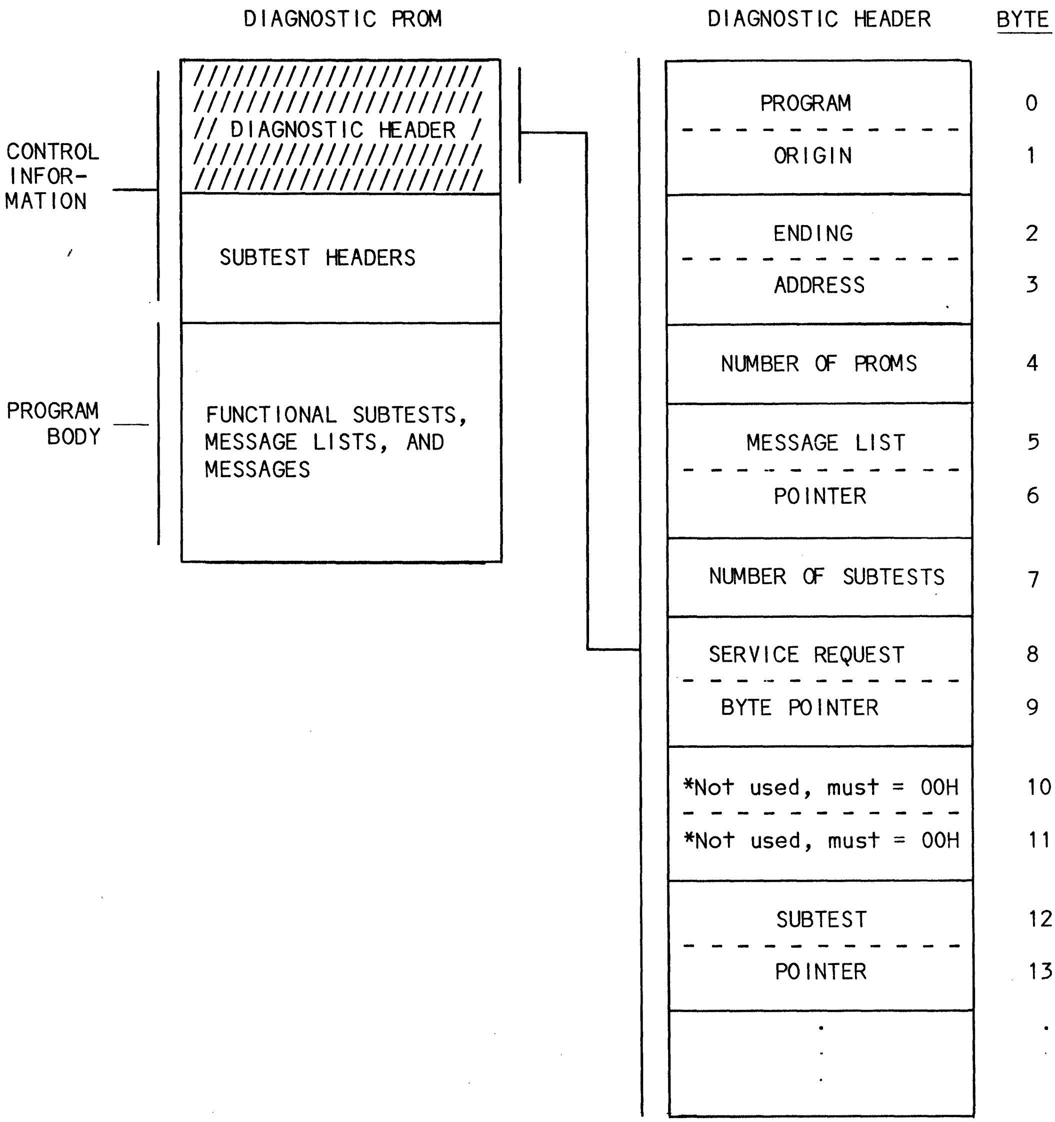
Figure 6-2. Diagnostic Program in PROMs

Control Information

The control information in the PROM consists of a diagnostic header followed by the subtest pointers and the subtest headers. If this information occupies less than 256 bytes, the leftover bytes can not be used because the uSE overlays the first 256 bytes of PROM 1 of the diagnostic program with the shadow RAM during program execution (see figure 6-1). The uSE provides the shadow RAM to the users program at execution time. The uSE can switch the RAM in and out of the working memory space as required for access to the diagnostic program control information and also to service special requests made by the diagnostic program. The information contained in the diagnostic header and subtest header areas is not accessible by the program during execution.

Diagnostic Header

The diagnostic header block contains information about the operating environment of the diagnostic program. The diagnostic header must start at byte 0 of the diagnostic program. Figure 6-3 shows the format of the diagnostic header block and the position of the diagnostic header block in the diagnostic program.



* Bytes 10 and 11 contain filter pointer information that is non-functional in the uSE but active in the uSA. Refer to the MicroSystem Analyzer Users Manual for additional information.

Figure 6-3. Diagnostic Header

A description of the bytes of the diagnostic header follows:

BYTE	DIAGNOSTIC HEADER	DESCRIPTION
0 1	PROGRAM ORIGIN	Two-byte address (hi-byte then low-byte) used by the system to map diagnostic program to the memory space of the UUT. The program origin of the address must fall on one of the following boundaries: Program Size — Boundary One 2708 PROM — 1K * Two 2708 PROMs — 2K * One 2716 PROM — 2K * Two 2716 PROMs — 4K *
2 3	ENDING ADDRESS	Two-byte ending address (hi-byte then low-byte) of the diagnostic program. This address is used to calculate program size.
4	NUMBER OF PROMS	Number of PROMs plugged into the keyboard (either 1 or 2).
5 6	MESSAGE LIST POINTER	Two-byte address pointer (hi-byte then low-byte) to the initial message list that identifies the program. The message size is restricted to 20 characters and is displayed when the PROM/MEM key is pressed. (See message list structure.)
7	NUMBER OF SUBTESTS	The number of subtests in the program. The value range is 1 through 99. (63H max)
8 9	SERVICE REQUEST BYTE POINTER	Two-byte address pointer (hi-byte then low-byte) to the communication service request byte. The address of the service request byte must be in the shadow RAM. (See description of the service requests.)
10 11	(FILTER CRITERIA uSA only) POINTER	
12 13	SUBTEST POINTER	
. . .		

* Boundary 1K = x000, x400, x800, xC00
2K = x000, X800
4K = X000

Diagnostic Header description (continued)

BYTE	DIAGNOSTIC HEADER BLOCK	DESCRIPTION
0 1	PROGRAM ORIGIN	
2 3	ENDING ADDRESS	
4	NUMBER OF PROMS	
5 6	MESSAGE LIST POINTER	
7	NUMBER OF SUBTESTS	
8 9	SERVICE REQUEST BYTE POINTER	
10 11	FILTER CRITERIA POINTER	These two bytes are used as an address pointer (hi-byte, low-byte) to the filter specifications in the MicroSystem Anaylzer (uSA). They are not used by the uSE.
12 13	SUBTEST POINTER	Two-byte address pointers (hi-byte then low-byte) to subtest header block. A subtest pointer is required for each subtest, and a diagnostic program may have as many as 99-subtest pointers.
. . .	. . .	

Subtest Header

The subtest header defines the control for the individual subtest. The most efficient placement of the subtest control blocks is immediately following the diagnostic header block, as this area is overlayed by the shadow RAM. Figure 6-4 shows the position of the subtest header block in the diagnostic program and the format of the subtest header. A description of the bytes of the subtest header is on the following page.

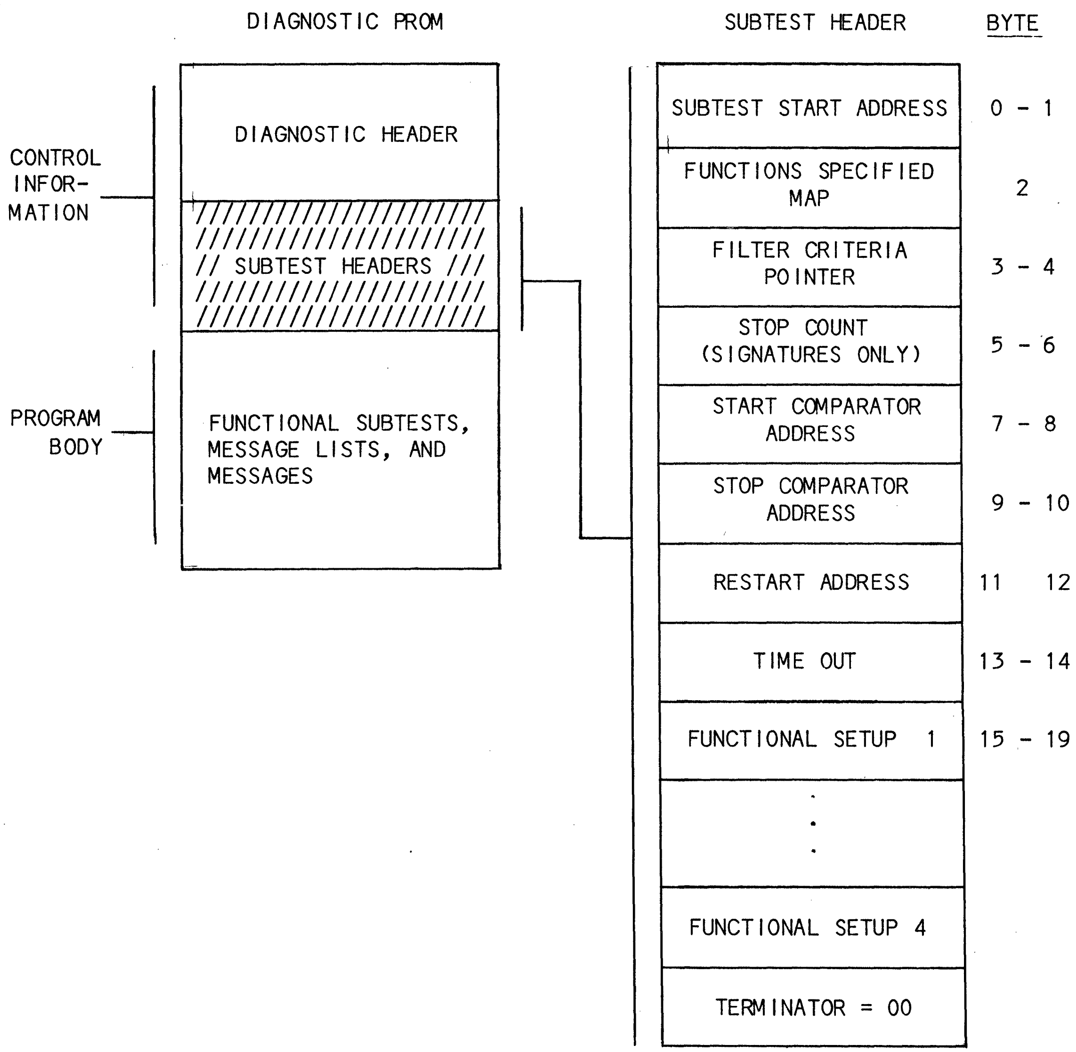
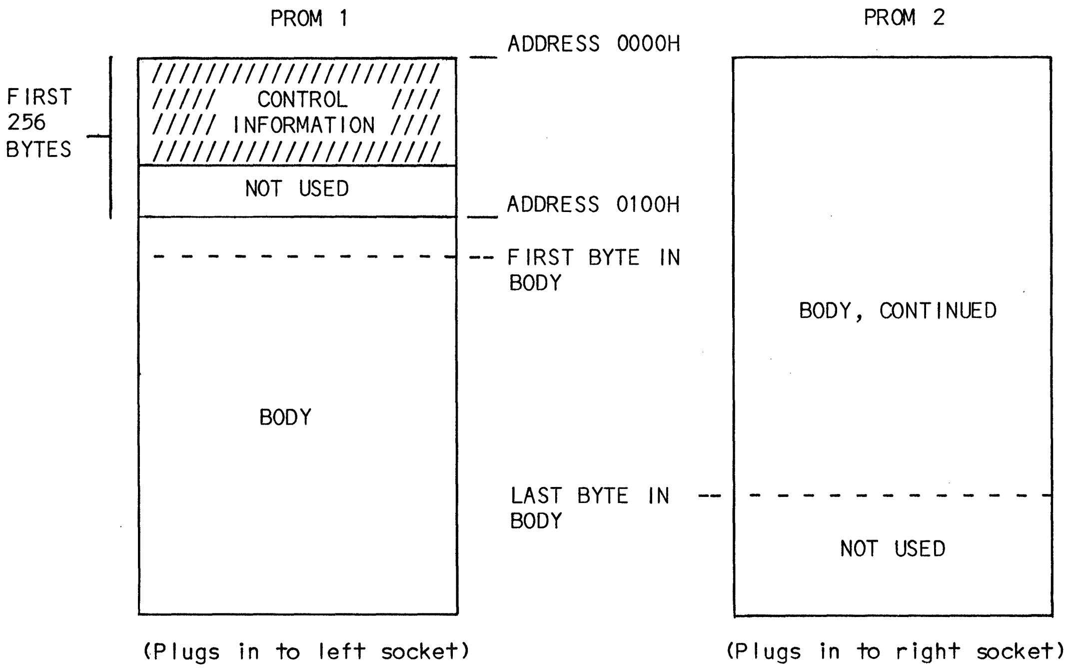


Figure 6-4. Subtest Header

BYTE	SUBTEST HEADER	DESCRIPTION
0 1	SUBTEST START ADDRESS	Starting address of the subtest. (High-byte, low-byte.)
2	FUNCTIONS SPECIFIED MAP	Indicates if any functional setups are present and also specifies how memory is to be mapped. Bit 0 indicates how memory is to be mapped. If it is zero, all memory except the test diagnostic memory is mapped to the UUT. If it is a one, all memory is mapped to the uSA; however, all control, address, and data signals are passed to the UUT. While mapped to the uSA, any memory control and data signals from the UUT are disregarded. This allows fault analysis on the UUT memory and bus structure. Bit 1 is used to indicate if one or more function setup is present. This is used for fault isolation. If it is a one, setups are present. If zero, all information after byte 4 is not used and a new subtest header may be started.
3 4	FILTER CRITERIA POINTER	Not used in the uSE. See the uSA manual for details. (bytes 3 through FUNCTIONAL SETUP N)
5 6	STOP COUNT (SIGNATURES ONLY)	
7 8	START COMPARATOR ADDRESS	
9 10	STOP COMPARATOR ADDRESS	
11 12	RESTART ADDRESS	
13 14	TIME OUT	
15 - 19	FUNCTIONAL SETUP 1	
	. . .	
	FUNCTIONAL SETUP N	
	TERMINATOR = 00	Terminator for subtest header.

Program Body

The bytes following the first 256 bytes are reserved for the body of the program. The body consists of functional subtests, message lists and messages and usually begins at the beginning of the PROM plus 256 bytes (100H). However, if the diagnostic program requires more than 256 bytes of control information, the body starts in the next contiguous byte after the last byte of the control information. Figure 6-5 illustrates an example in which control information occupies less than 256 bytes, and figure 6-6 illustrates an example in which it occupies more than 256 bytes.



NOTE: PROM 2 is needed only if the body of the program is too long to fit into PROM 1.

Figure 6-5. Program Example with Less Than 256 Bytes of Control Information (slashed area)

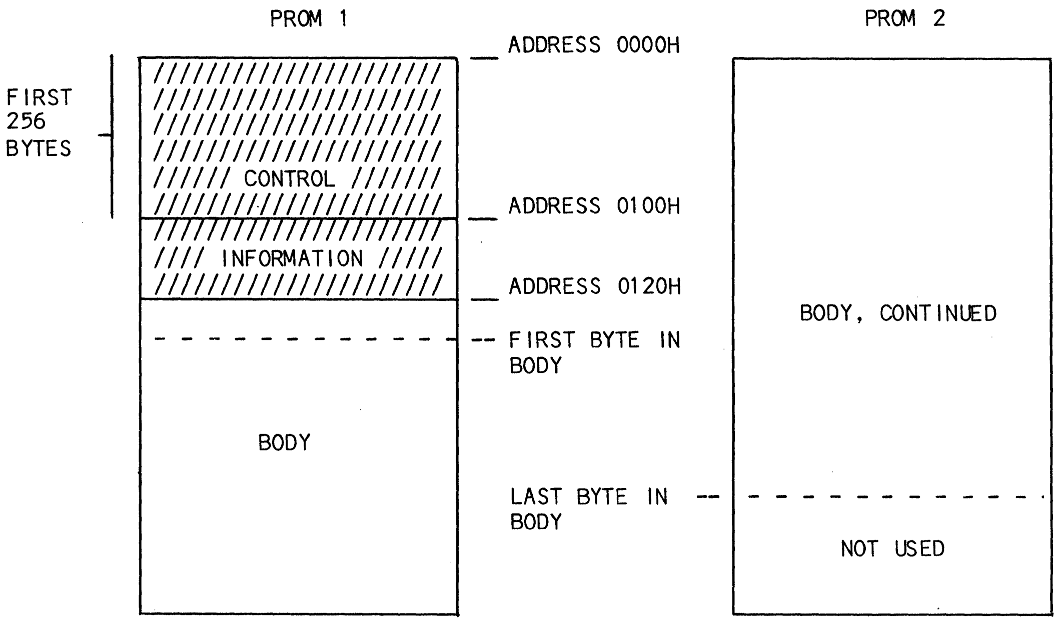


Figure 6-6. Program Example with More than 256 Bytes of Control Information

SHADOW RAM

The uSE operating system uses shadow RAM for data transfer and service requests between the uSE and the diagnostic program.

When the PROM/MEM key is pressed, the shadow RAM is enabled and set to zero. Four bytes of the shadow RAM are used for making a service request (see figure 6-7). When data is stored in the service request byte, the uSE interrupts, pauses the UUT, and determines what action is to be taken.

The data stored at the service request byte communicates to the system: when the subtest is finished, when the diagnostic program is finished with all of the subtests, and if there is a message to be displayed. Also encoded in the service request byte is what the system should do after servicing the requests. The options are: pause the microprocessor under test, wait a period of time, or keep going. If the wait option is selected, the wait time is stored in the wait time byte in the Shadow RAM. If the request was to display a message, then the two bytes following the servie request byte point to the message list. The rest of the shadow RAM is available to the programmer for data storage and stack pointer.

Figure 6-7 shows the format of the shadow RAM.

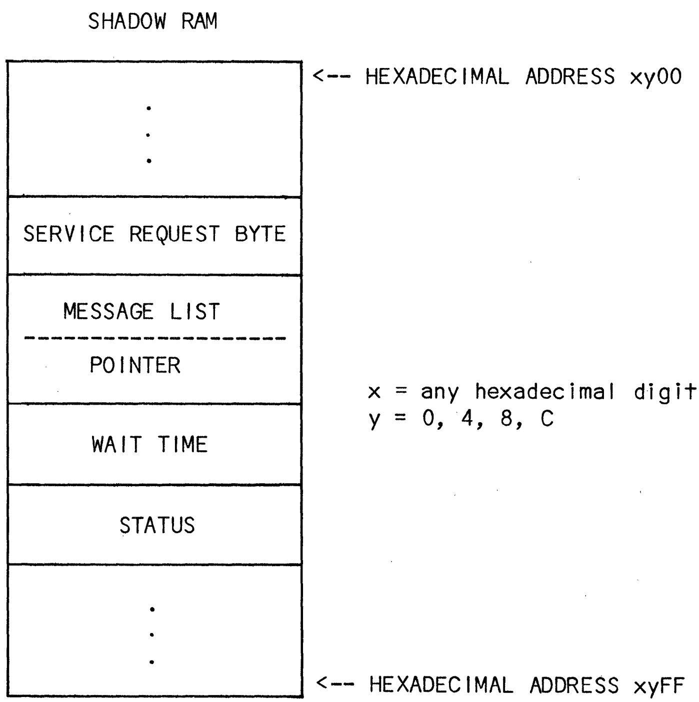
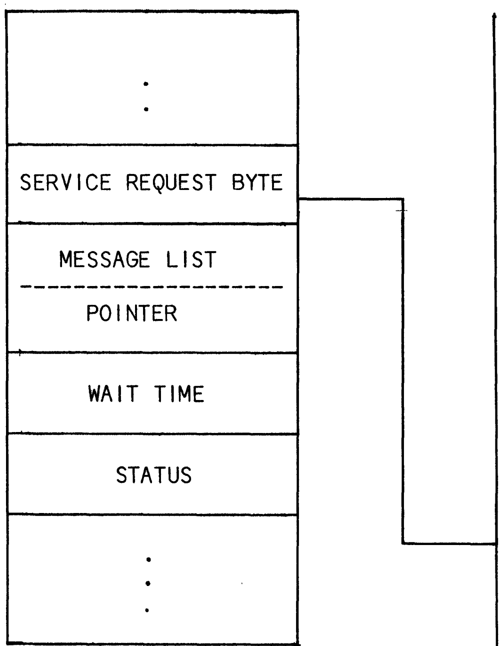


Figure 6-7. Shadow RAM Format

Each byte of the shadow RAM is described below:

SHADOW RAM



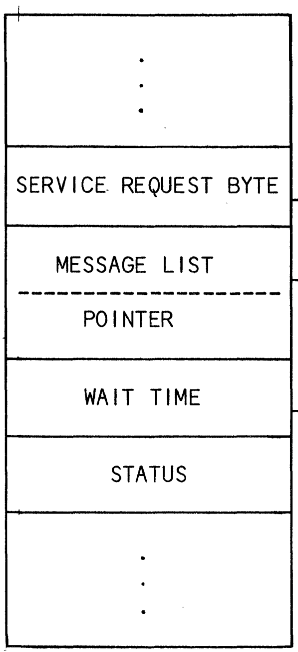
DESCRIPTION

The uSE software provides the following functions for the operating program through the use of a Service Request mechanism. A service is requested by storing data in the form of a coded data byte into the location defined as the Service Request byte in the diagnostic shadow RAM. The storing of data into this defined location causes the operating system to perform the request. The request types are defined below.

Hex Value	Description
01	END SUBTEST
	This signifies the end of the sub-test and one of the following actions will occur, depending on the operator selected subtest.
	If SUBSEL = 0, at the end of each subtest, the subtest number is incremented and the next subtest is executed until the last subtest has been run, at this point the program is restarted at subtest 1.
	If SUBSEL $\neq$ 00, the program loops on the selected subtests.
02	MESSAGE OUTPUT
	The MESSAGE PENDING LED indicator is illuminated and the program is paused for operator inspection of the message. If input is allowed, data can be taken from the appropriate keypad at this time. See the Message List description, page 6-14 for details.

Shadow RAM description (continued)

SHADOW RAM



SERVICE REQUEST (continued)

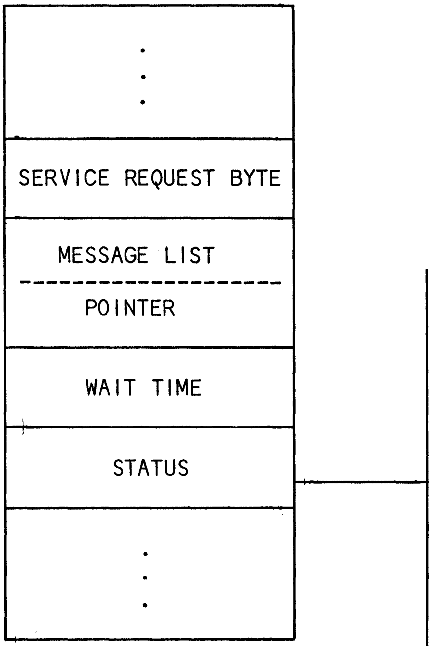
Hex Value	Description
04	END TEST The program is paused and one of the following actions will occur, depending on the operator selected subtest. If SUBSEL = 0, program restarts at subtest 1 when RUN key is pressed. If SUBSEL = 0, selected subtest is repeated when RUN key is pressed.
08	HALT OVERRIDE (or CONTINUE) This value, which can be ORed with any other service request, causes the operating system to perform the requested action and then resume operation. This action can be used for diagnostic loop control and message output.
10	WAIT This value can be used by itself or ORed with the MESSAGE OUTPUT request. After displaying the message, the system waites a specified amount of time before resuming. The time is specified by the WAIT TIME byte.

For message output to the uSE via the Service Request, the two bytes immediately following the SERVICE REQUEST BYTE must contain the absolute address of the associated message list. (Hi-byte the low-byte.)

This value specifies the amount of time to wait. Each hexadecimal digit represents approximately 100 milliseconds.

Shadow RAM description (continued)

SHADOW RAM



DESCRIPTION

The status byte, the fourth byte after the service request byte, is set by the uSA to indicate which fault detection measurement is currently being made. The status byte is updated whenever a function key is pressed.

The format of the status byte is:

Hex Value	Description
00	No Fault Detection Measurement
10	Signature
20	Count
40	Interval
80	Frequency

MESSAGES

Diagnostic programs can initiate message displays and request data input from the operator. The Message List is a control block in the body of the Diagnostic PROM that is used to control and specify display data and input operations. Message list is addressed by a pointer in the Shadow RAM as shown in figure 6-8. Refer to paragraph on Shadow RAM for details on the Message List Pointer.

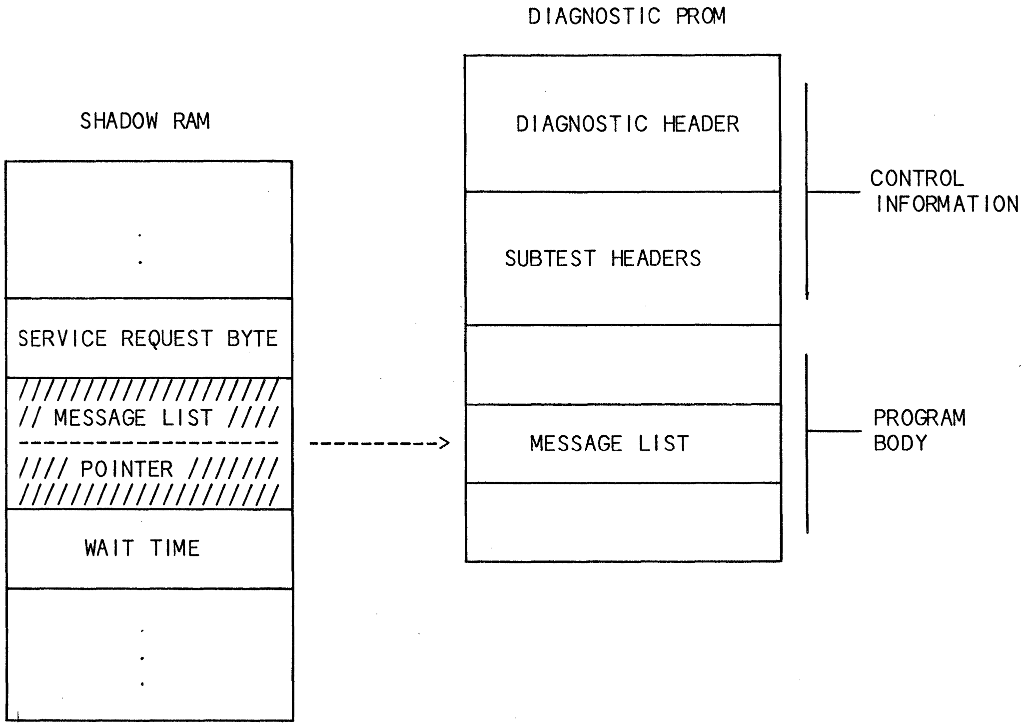


Figure 6-8. Message List

Messages can be from one to twenty characters in length. If more than twenty characters are required, messages may be linked, in this case multiple lines are displayed. The test program initiates a message output via a service request to the uSE. A message can be one of two types: Display and Halt or Display and Continue.

Display and Halt Messages

The MESSAGE PENDING LED indicator is illuminated, a beep is sounded, and the test program execution is halted. When the MSG key is pressed, the message will be displayed. Once the MSG key has been pressed, messages will be automatically displayed until a key is pressed that results in a display change. If more than one display line is associated with the message (linked message), the operator can step through the message by pressing the INCR key. To resume execution of the test, press the RUN key.

Display and Continue Messages

Similar to the Display and Halt Messages except that test execution automatically resumes after the message is displayed. Note a delay may be necessary to guarantee display stability long enough for visual perception. To output a message of this type, the Service Request byte consists of the message output bit ORed with the halt override bit, or the wait bit.

To display a message, the diagnostic program will normally store the wait time. If any message is contained in the wait time byte, the diagnostic program will then store the address of the message list in the message list pointer, and lastly, store the message output request in the service request byte.

Message List

A message list consists of one or more message list entries as shown in figure 6-9.

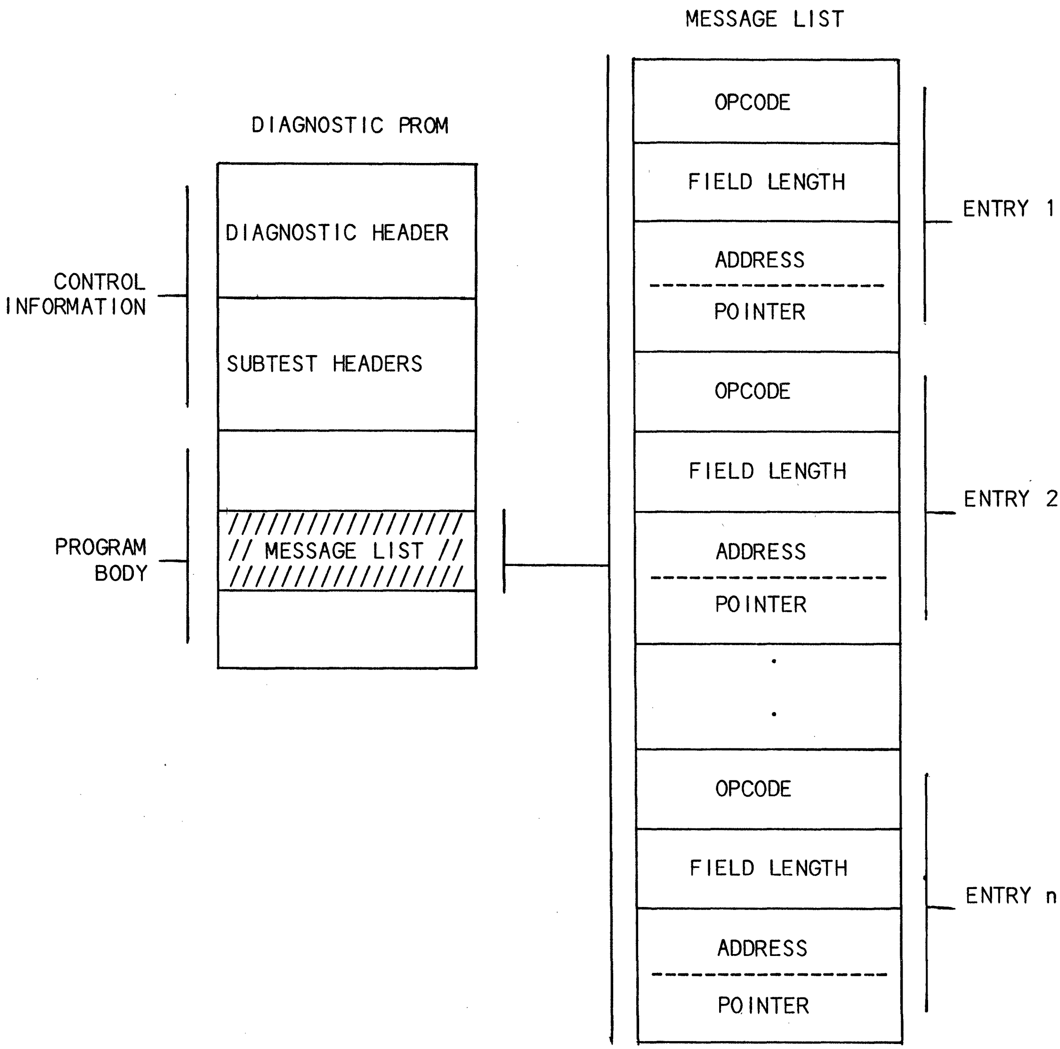


Figure 6-9. Message List

A description of the bytes of the message list follows:

BYTE	MESSAGE LIST ENTRY
0	OPCODE
1	FIELD LENGTH
2	ADDRESS
3	POINTER

DESCRIPTION

OPCODE (byte 0): Specifies operation type and data format. The definition of values for this byte are shown below:

Hex Value	Meaning
00	Output only, input not allowed.
01	Field allows operator input.
02	The data displayed or input is in hexadecimal format.
04	Linked message. One or more additional messages.
0A	The data displayed or input is in binary format.
20	The data displayed or input is in decimal format.
40	The data displayed or input is signed (+ or -).
80	Last entry of MESSAGE LIST.

Any of above may be ORed together.

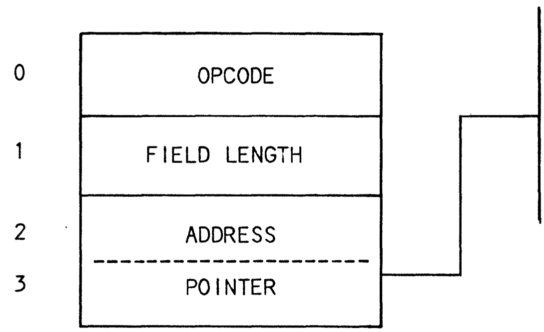
FIELD LENGTH (byte 1): Length of the field displayed. The maximum size of the field for each data type is defined below:

DATA TYPE	SIZE OF FIELD
ASCII	Number of characters displayed
Hexa-decimal	Number of digits displayed
Binary	Number of bits displayed
Decimal	1 to 5 decimal digits

Message List description (continued):

BYTE MESSAGE LIST ENTRY

DESCRIPTION



Address of the associated information. (Hi-byte then low-byte.) The types of information pointed to are pointed to are:

ASCII MESSAGE
LINKED MESSAGE LIST
DATA STORED IN SHADOW RAM

Figure 6-10 shows an example of a simple ASCII message.

BYTE MESSAGE LIST

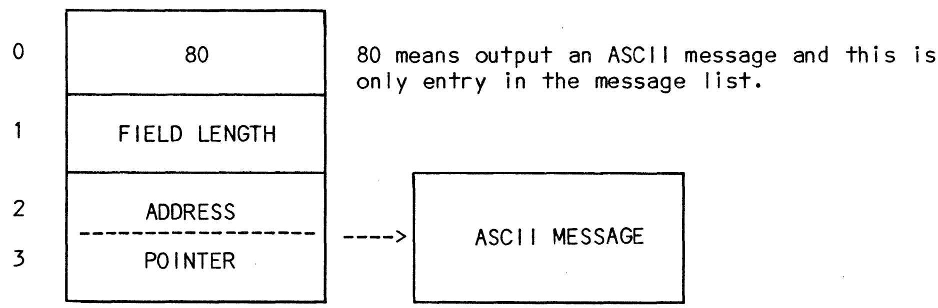


Figure 6-10. Simple ASCII Messages

Figure 6-11 shows a message list with an ASCII message and an hexadecimal data input storage area.

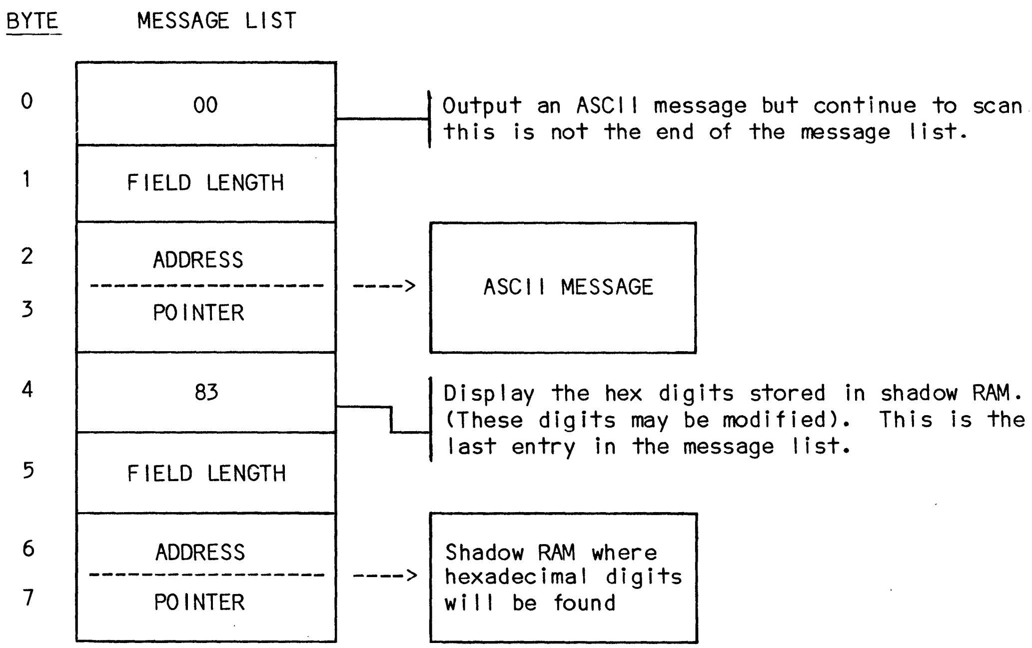


Figure 6-11. ASCII Message and Hexadecimal Data

The last entry on a message list must always be designated as last, even if the last entry is a link entry. See figure 6-12.

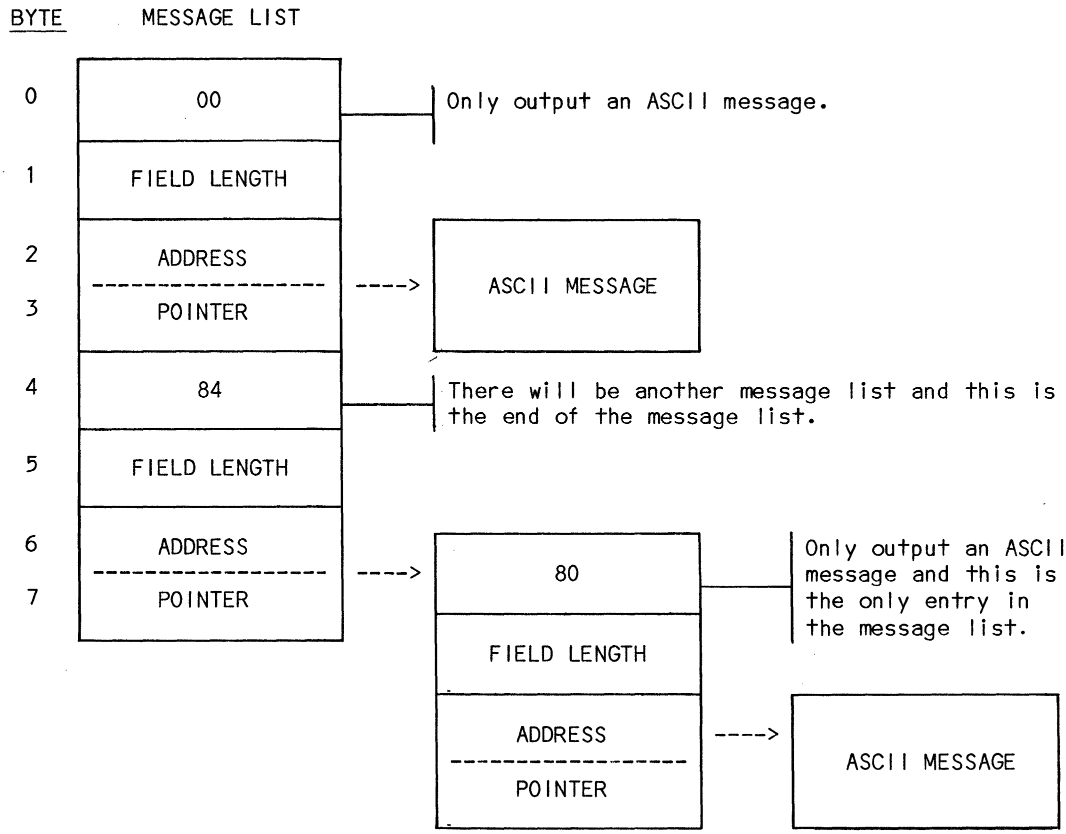


Figure 6-12. Linked List of Two ASCII Messages

To view the second ASCII message, the operator would push the INCR key.

The last message list of a group of linked message lists may also be a linked list and may point to the first (or any other) message list. This allows the operator to continually scroll a number of messages. An example of this is the REGISTER display.

Data Formats

The internal format of data associated with message lists is in either ASCII or hexadecimal format. The internal data types carried in binary are as follows:

Hexadecimal

Each hexadecimal character is represented internally by four bits. The number of characters allowed is limited by the display length and is specified by the Field Length entry of the Message List. The data is carried internally and is right-justified. In the case of an odd number of hexadecimal characters, the most significant nibble (4 bits) of the internal field is set to zero by the uSE.

Binary

Binary is a variation of the hexadecimal format, in that, before the BINARY key is pressed, the data is input/output in hexadecimal format. The data is still internally right-justified in hexadecimal format. The length field entry of the message list must specify the number of binary characters. The number of characters displayed for binary will be the nearest MOD 4 number. For example, if a 10-bit binary field is specified, then three hexadecimal characters will be displayed. If a hexadecimal character is entered that exceeds the specified binary field, then it is treated as an illegal input character and is not accepted.

Binary data is always displayed in the last n character position of the display (right-justified). Therefore, a binary field must be the last field of display.

Decimal

Decimal data is converted to hexadecimal when the ENTER key is pressed. Decimal data is limited to one to five characters (0-65, 535) in length and is always stored in a two-byte field (right-justified) as an unsigned, 16-bit hexadecimal number. When a decimal field is to be displayed, it is converted from hexadecimal format to decimal format. If the value exceeds the range of digits specified in the message list, "ERR" will be displayed (right-justified) and a warning beep will be sounded.

Signed

A signed field is stored internally as an ASCII plus (+) or minus (-).

MOVE USER TO RAM

The following example demonstrates a sequence used when moving a block of memory in the user's system, address 0 thru 07FF to the internal RAM at address 0. All following examples assume the use of a Z80A Emulator Board and pod.

Operation	Display	Explanation
RESTART	Z80 READY 2708	
MOVE	MV PR 0000 0000 0000	Starts out at move PROM to RAM.
INCR	MV UR 0000 0000 0000	Choose move user to RAM.
0, ENTER	MV UR 0000 0000 0000	Starting address.
07FF, ENTER	MV UR 0000 07FF 0000	Move data in user system from address 0-07FF to RAM/MEM.
ENTER	ENABLE RAM @ 0000	RAM not enabled.
RAM/MEM	DIAG RAM-1 ADR=0000	Envoke RAM/MEM.
INCR	DIAG RAM-2 ADR=0000	Don't want diagnostic mode.
INCR	AP RAM-1=0000 S=1K	Want applications mode.
ENTER	AP RAM-1=0000 S=1K	Keep address the same.
2, ENTER	AP RAM-1=0000 S=2K	Set memory size to 2K.

OPERATION EXAMPLES

Operation	Display	Explanation
ENABLE	*AP RAM-1=0000 S=2K	RAM MEM ENABLED LED is illuminated, the * is inserted in the display and the RAM enabled is ready for use.
MOVE	MV PR 0000 0000 0000	Start move PROM to RAM.
INCR	MV UR 0000 0000 0000	Move user to RAM.
ENTER	MOVE IN PROGRESS	Move data to address 0 in RAM/MEM.
	MOVE COMPLETE	(Approximately one minute.)

MOVE PROM TO RAM

To move diagnostics from the front panel PROM to the internal RAM for debug, follow the sequence shown below. In the example, the diagnostic PROM starts at address 2000 and extends thru address 27FF.

Operation	Display	Explanation
RESTART	Z80 READY 2708	
MOVE	MV PR 0000 0000 0000	Invoke move function.
2000, ENTER	MV PR 2000 0000 0000	Choose starting address at 2000H.
27FF, ENTER	MV PR 2000 27FF 0000	Choose ending address at 27FFH.
2	MV PR 2000 27FF 2000	Choose destination starting address at 2000H.
ENTER	ENABLE RAM @ 2000	RAM not enabled.
RAM/MEM	DIAG RAM-1 ADR=0000	Envoke RAM/MEM.
INCR	DIAG RAM-2 ADR=0000	Don't want diagnostic mode.
INCR	AP RAM-1=0000 S=1K	Want applications mode.
2, ENTER, 2, ENTER	AP RAM-1=2000 S=2K	Keep address the same.
ENABLE	*AP RAM-1=2000 S=1K	RAM/MEM/LED is illuminated, * inserted in display and the RAM enabled is ready for use.

Operation	Display	Explanation
MOVE	MV PR 2000 27FF 2000	Move still setup.
ENTER	MOVE IN PROGRESS	Start move.
	MOVE COMPLETE	
RAM/MEM	DIAG RAM-1=2000	Select diagnostic mode for internal RAM.
ENABLE	INITIAL MESSAGE	RAM MEM ENABLE LED is still illuminated.
RUN		Run diagnostic.

IN-CIRCUIT EMULATION FOR LABORATORY DEVELOPMENT APPLICATIONS

1. Turn off the UUT and the uSE and remove the microprocessor from the UUT.

2. Insert the In-Circuit Emulator 40-pin probe into the microprocessor socket in the UUT.

```
* * * * * * * * * * * * * * * * * * * * * *
*                                         *
*                CAUTION                  *
*                                         *
*  Insure that pin 1 on the socket is     *
*  aligned with pin 1 on the probe.       *
*                                         *
* * * * * * * * * * * * * * * * * * * * * *
```

3. Turn on the UUT and the uSE.

4. Press the RESTART key. Some choices are:

   a. To examine and/or modify any of the microprocessor registers or memory locations, use the REGISTER or MEMORY key. See Emulator Supplement for details.

   b. To execute a program in the UUT, set the program counter to the proper address using the REGISTER key, then press the RUN, RUN/DISP or STEP key.

   c. To set one hardware breakpoint, use the BREAK key.

   d. To read or modify an I/O port in the UUT, use the I/O key.

   e. To set the memory in the UUT to a defined value, use the FILL key.

   f. To execute a diagnostic PROM, insert the PROM(s) into the front panel socket(s), press the PROM/MEM key. If all subtests are to be executed, press the RUN key. If a particular subtest is to be run, press the SUBSEL key and choose the subtest, then press the RUN key.

EXECUTING A DIAGNOSTIC PROM

1. Turn off the UUT and the uSE and remove the microprocessor from the UUT.

2. Insert the in-circuit emulator 40-pin probe into the microprocessor socket in the UUT.

```
* * * * * * * * * * * * * * * * * * * * * *
*                                         *
*               C A U T I O N             *
*                                         *
*  Insure that pin 1 on the socket is     *
*  aligned with pin 1 on the probe.       *
*                                         *
* * * * * * * * * * * * * * * * * * * * * *
```

3. Insert the PROM(s) into the socket(s), turn on the UUT and the uSE, and press the RESTART key. The uSA displays:

```
Z80    READY    2708
```

4. Press the PROM/MEM key. The uSE displays an initial message such as:

```
UUT TEST
```

5. Press the RUN key if all subtests are to be executed.

6. Or, if a particular subtest is to be executed, select the subtest with the SUBSEL key, then press the RUN key.

MEMORY EXAMINATION AND MODIFICATION

The MEMORY key is used to read or modify data in the user's memory. The uSE can read and modify memory in the UUT. On a memory "write", the data is entered and then is read back to verify that the proper data was written.

When the MEMORY key is pressed the uSE prompts for the memory address. After the address has been entered and the ENTER key has been pressed, the data contained at that address is displayed.

This is a standard feature of in-circuit emulators and shows how the resources of the system are available through the uSE.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                             *
*                          N O T E                            *
*                                                             *
*  (Execute the entire) load memory sequence that follows     *
*  to insure proper operation of the demonstration.           *
*                                                             *
*                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

This example writes a program to the user RAM. All operations are on the uSE.

Operation	Display	Explanation
RESTART	Z80 READY 2708	Initializes the uSE.
MEMORY	MEM 0000	The memory address can now be entered.
0801, ENTER	MEM 0801 = 81	Set memory address to 0801 and display the address.
3C, ENTER	MEM 0802 = 16	Store 3C (INC A) at address 0801, increment to the next address and display the data.
32, ENTER	MEM 0803 = FA	Store 32 (STA...) at address 0802, increment to the next address and display the data.
09, ENTER	MEM 0804 = 0A	Store 09 (@0809) at address 0803, increment to the next address and display the data.

Operation	Display	Explanation
08, ENTER	MEM 0805=D8	Store 08 at address 0804, increment to the next address and displays the data.
C3, ENTER	MEM 0806=0A	Store C3 (JMP...) at address 0805, increment to the next address and display the data.
01, ENTER	MEM 0807=1D	Store 01 (...TO 0801) at address 0806, increment to the next address and display the data.
08, ENTER	MEM 0808=03	Store 08 at address 0807, increment to the next address and display the data.

The following example shows how to examine consecutive memory locations.

Operation	Display	Explanation
MEMORY	MEM XXXX	
0801, ENTER	MEM 0801=3C	Sets memory address to 0801 and displays the data at that address.

Use the INCR key to verify that the data has been written to the RAM correctly. Press the INCR key to display each successive memory location and compare the data displayed with the data listed below:

Address	Data	Instruction
0801	3C	Increment register "A" by one.
0802	32	Store register A at memory address 0809.
0803	09	
0804	08	
0805	C3	Jump to memory address 0801.
0806	01	
0807	08	

N O T E

If a data byte is incorrect, change it by keying in the correct data and then pressing the ENTER key.

REGISTER EXAMINATION AND MODIFICATION

The REGISTER key is used to examine the internal registers of the microprocessor. The uSE, using in-circuit emulation, has full access to the internal registers of the microprocessor. The microprocessor has been replaced by the uSE. Pressing the REGISTER key allows the internal registers to be examined and the INCR key is used to step through the different registers. The features and benefits of having this key allows the user to start execution at a specified memory address and set internal registers to defined values.

Operation	Display	Explanation
REGISTERS	PN=0000 PL=0000 I=C3	The program counter next (PN and the program counter last (PL) as well as the first byte of the last instruction executed (I) are displayed in hexadecimal.
INCR	SP=0000 CC=00	Advances to the next display. The Stack Pointer (SP) and the Flag Register (CC) are now displayed. The SP is in hex and the CC is in hex.
BINARY	SP=0000 CC=00000000	Display CC in binary.
INCR	RA=00 BC=00 00	Advances to the registers A, B and C (RA, BC) respectively.
INCR	DE=00 00 HL=0000	Advances to register pairs DE and HL.
INCR	IX=0000 IY=0000	Advances to registers IX and IY.
INCR	CC'= 00000000	Advances to the auxiliary flag register (CC') which is displayed in binary.
INCR	RA'=00 BC'=00 00	Advances to the auxiliary A', B', and C' registers (RA', BC').
INCR	DE'=00 00 HL'=00 00	Advances to the auxiliary DE & HL register pairs (DE', HL').

OPERATION EXAMPLES

Operation	Display	Explanation
INCR	IR=00 R=1A IM= 00	Advances to the Interrupt Register, Refresh Register and the Interrupt mode (IR, R, IM).
INCR	IFF2=0 IFF1= 0	Increments to the interrupt flip flops 2 and 1 (IFF2, IFF1).
INCR	PN=0000 PL=0000 I=C3	Increments back to PC next and PC last.
0801, ENTER	PN=6801 PL=0000 I=C3	Set PN (Program Counter Next) to starting address of the program that was previously written into the users memory.

Proceed to the page titled PROCESSOR CONTROL KEYS, but DO NOT push RESTART.

PROCESSOR CONTROL KEYS

The RUN, RUN/DISPLAY, AND STEP keys are used to control how the processor will be operated. In the RUN mode, the uSE is operating the system.

In the RUN/DISPLAY mode, the system under development is interrupted ten times per second and the display is updated during each interrupt. However, when running, the system under development runs at full speed. The STEP key allows one instruction to be executed for each key activation. Pressing any of these keys will cause the slave processor to start executing at the address specified by the program counter next. The three modes of operation allow flexibility in controlling operation of the system under development.

Operation	Display	Explanation
RUN/DISP	PN=080X PL=080X I=XX	The program runs in a loop; the display is updated 10 times/second. PN, PL and I are changing, and the RUN LED is illuminated.
INCR	SP=0000 CC= XXXXX000	The stack pointer is not changing. The condition code is in binary and only certain bits are changing.
INCR	RA=XX BC=00 00	Register A changes and Register B and C do not change.
INCR (8 times)	PN=080X PL=080X I=XX	Advance the display back to the PN, PL, I register and notice all are changing.
STEP	PN=0801 PL=0805 I=C5	Press STEP key a few times; see how PN goes to PL and there is a loop.
INCR, INCR	RA=D5 BC=00 00	Advance the display to the A Register.
STEP	RA=D6 BC=00 00	Press STEP key a few times; show that the A register is being incremented by one on every third push of the STEP key.
RUN	RA=D6 BC=00 00	The uSE is now running full speed.

BREAKPOINT

The uSE has one hardware breakpoint, which is used to interrupt the normal program flow. An address is specified and when this address is referenced, either for a read operation, or a write operation, break will occur.

The uSE monitors the selected address and when a read or write operation takes place, the uSE pauses the slave uP and determines what to do next. The options are (1) the specified address must be referenced several times before further operations take place, (2) on the first reference to the address, the slave will either be paused, or execution will resume at a specified address.

Upon pressing the BREAK key, the uSE will prompt for the break address and the query for further options.

This feature allows the operator to determine if an address is being referenced when it shouldn't be, and from where, or allows the operator to inspect the internal registers after referencing a memory location some number of times.

Operation	Display	Explanation
RESTART	Z80 READY 2708	Initializes uSE.
BREAK	BRKA=0000 OPT=C	
0805, ENTER	PRKA=0805 OPT=C	Sets break address at location 0805, the JUMP Instruction.
BINARY	BRKA=0805 OPT= 1100	Display options in binary.
---->	BRKA=0805 OPT=1100	Move to next bit, break on read reference.
---->	BRKA=0805 OPT=1100	Move to pass count, pass count is the number of times the address is referenced before the break.
1	BRKA=0805 OPT=1110	Select this option. Next bit is jump to another address, (1) or pause (0).
ENTER	BRKA=0805 OPT= 1110	Pause option selected.

Operation	Display	Explanation
INCR	PASS COUNT=0000	
0016, ENTER	PASS COUNT=0016	This sets pass count 0016 (decimal).

This sets breakpoint for write access, or read access on address 0805 after 16 loops through that address.

ENABLE	PASS COUNT=0016	This enables the breakpoint - Notice that BREAK ENABLE LED is on.
REGISTER	PN=0000 PL=0000 I=C3	Use the INCR key to increment to PN, if needed.
0801, ENTER	PN=0801 PL=0000 I=C3	Sets program counter next to start of program.
INCR	SP=0000 CC=00	Increment to Stack Pointer.
INCR	RA=XX BC=00 00	Increment to Register A.
00, ENTER, ENTER, ENTER	RA=00 BC=00 00	Sets Register A to zero and leaves Registers B and C as is.
RUN	BREAK POINT=0805	Break is executed and break address is displayed.

After break, push REGISTER key to show register RA = 10. Press RUN and REGISTER keys a few times to show that 10_{16} is added to register A, everytime the break is encountered.

REAL-TIME TRACE DEMONSTRATION

The Real-Time Trace capacity of the uSE is an extremely powerful debug aid. Because of the level of sophistication of the Real-Time Trace, all of its operatons and flexibility can be somewhat complicated. The information contained below should help ease this situation. The information provides a keystroke-by-keystroke run-through of the various options at the user's command in using the uSE.

Operation	Display	Explanation
RESTART	Z80 READY 2708	The restart key initializes the uSE and allows us now to select the options we want.
TRIG1	T1 EVT ADDR = OFF	After pressing TRIG1 key, this now starts setting up the equation that will be used. The first thing seen is address. Address can be ignored in the off-state.
INCR	T1 EVT ADDR = 0000	Or it can be set equal to a value.
INCR	T1 EVT ADDR = >0000	The address can also be a range, in this case, greater than or equal to a value.
INCR	T1 EVT ADDR = <0000	Or it can be less than or equal to a value.
INCR	T1 EVT ADDR = OFF	Again, the increment key gets back to the address being off.
FLD SEL	T1 EVT BUS = ALL	Field select moves to the next part of the equation. After the proper address range has been established, bus information is presented. All means any bus cycle type will satisfy the equation.
INCR	T1 EVT BUS = FTCH	The fetch is the instruction fetch and only looks at information during instruction fetch.
INCR	T1 EVT BUS=MEM RD/WR	This means any memory operation whether it be read or write.

Operation	Display	Explanation
INCR	T1 EVT BUS=I/O RD/WR	This means any I/O operation, whether it be read or write.
INCR	T1 EVT BUS = ALL RD	This means any bus activity will be a read, whether it's memory or I/O.
INCR	T1 EVT BUS = ALL WRT	The same is true for the write operation.
INCR	T1 EVT BUS = MEM RD	In this case, only memory reads.
INCR	T1 EVT BUS = MEM WRT	Now only memory writes.
INCR	T1 EVT BUS = I/O RD	Now only I/O reads.
INCR	T1 EVT BUS = I/O WRT	Now only I/O writes.
INCR	T1 EVT BUS = ALL	And back to ALL. By having the capability to choose a particular bus cycle, the equation can be conditioned for what the microprocessor is actually doing on the bus at any given time.
FLD SEL	T1 EVT EXT = xxxxxxxx	The field select now allows the 8-external events. These are chosen by the extra clips that come with the Real-Time Trace cable.
1, 0, ENTER	T1 EVT EXT = 10xxxxxx	A 1 or a 0 can be chosen instead of the x. The x means don't care.
x, x, ENTER	T1 EVT EXT = xxxxxxxx	Or they can be set to don't care state.
FLD SEL	T1 EVT DATA = OFF	The field select key moves to the next part of the equation which is what is happening on the data bus. Off means don't care.

Operation	Display	Explanation
INCR	T1 EVT DATA = 00	Or a particular value on the data bus can be specified.
INCR	T1 EVT DATA = >00	Or anything greater than, or equal to, the value can be specified.
INCR	T1 EVT DATA = <00	Or anything less than, or equal to, the value.
INCR	T1 EVT DATA = OFF	Or, again, back to the don't care or off condition.
FLD SEL	T1 PASS = OFF	What has been defined so far is an equation for an event consisting of address information, data information, bus activity, and external events. The field select moves to a field where this event can now be specified to occur "N" number of times, i.e. a pass count on that event.
INCR	T1 PASS = 00000 EVT1	Here a number between 2 and 65,535 can be chosen.
INCR	T1 PASS = OFF	Or the pass count is off.
FLD SEL	T1 DELAY = OFF	Field select now moves to what is called the delay field. Once we have satisfied the event and how many times we wish the event to happen, we can specify the breakpoint not to occur at the point of the defined event, but to delay some number of cycles.
INCR	T1 DELAY = 00000 EVT1	The first choice is to delay n number of additional event 1's. This number again can be from 2 to 65,535 events.
COUNT	COUNT = 00000 EVT1	The count key allows us to define in what units the delay will be timed. This delay is used for both T1 and T2. The count will also count the number of events that have happened. In this case, we have chosen event 1's.

Operation	Display	Explanation
INCR	COUNT = 00000 EVT2	Or choose event 2's.
INCR	COUNT = 00000 TRC	This allows us to position the data in the trace buffer by specifying how many more pieces of trace information we put into the trace buffer. Once the equation is satisfied, the uSE will continue to run and store a specified number of traces and put them into the buffer before the operation is terminated.
INCR	COUNT = 00000 uSEC	Here we can delay so many microseconds after the event.
INCR	COUNT = 00000 MSEC	Or if more time is needed we can delay so many miliseconds.
INCR	COUNT = 00000 FTCH	This allows us to delay a fixed number of instructions.
INCR	COUNT = 00000 MCYC	Or we can delay machine cycles.
INCR	COUNT = 00000 EVT1	Which brings us right back around to event 1.
INCR	COUNT = 00000 EVT2	Event 2, again.
INCR	COUNT = 00000 TRC	We will choose to delay some number of traces.
TRIG1	T1 DELAY = 00000 TRC	Pressing trigger 1 again we now see that the T1 delay is the number of traces. So the count chooses in what unit the delay for T1 is to be measured.
INCR	T1 DELAY = OFF	Again, pressing INCR key will turn off the delay mode.

Operation	Display	Explanation
FLD SEL	T1 EVT MODE = IND	We have now completely specified a trigger and there are two of these; trigger 1 and trigger 2. What we now see is in what mode we will allow these triggers to be used. Independent mode means that trigger 1 is independent from trigger 2. There is no relationship between the two.
INCR	T1 EVT MODE = LIMIT	Limit mode means that both trigger 1 and trigger 2 must happen simultaneously. When using the breakpoint; trigger 1 will be used to stop operation.
INCR	T1 EVT MODE = ARM	The next mode is arm mode. This a sequential operation where T1 must happen before T2. T2 would be used as the breakpoint to stop operation.
INCR	T1 EVT MODE = FRZ	The freeze mode is very similar to the arm mode operation. Again it is sequential and T1 must happen then T2. In this mode tracing begins when the run key is pushed and ceases when T1 occurs but the microprocessor will still continue to operate until T2 occurs. This is useful when you have real time operation and you wish to examine the results up to a given point (T1), but need to allow the program to continue to another point (T2) before emulation can be broken. An example of this would be starting a motor and it must be turned off. Otherwise it will burn up or it will hit the limit switch. What this provides is a means of making sure the proper information is saved in the trace and then allow the operation to finish.
INCR	T1 EVT MODE = IND	This gets us right back to the independent mode.
FLD SEL	T1 EVT ADDR = OFF	The field select has now made a complete circle and we are right back at the T1 event address.
TRIG2	T2 EVT ADDR = OFF	Pressing trigger 2 gives the same information as trigger 1. We can set up the T2 event to be an address; again by pressing INCR key, we could go through equal to a value greater than or less than a value.

Operation	Display	Explanation
FLD SEL	T2 EVT BUS = ALL	Field select moves to the bus qualifier. It is exactly the same as trigger 1.
FLD SEL	T2 EVT EXT = xxxxxxxx	T2 event is now the external data. The eight externals are exactly the same as the T1 event.
FLD SEL	T2 EVT DATA = OFF	Field select now moves to the data domain. Pressing the INCR key will scroll through equal to a value, greater than or less than a value.
FLD SEL	T2 PASS = OFF	Now instead of counting event l's, this mode counts event 2's, because we are dealing with trigger 2. The range on the pass count can be from 2 to 65,535.
INCR	T2 PASS = 00000 EVT2	The INCR key turns off the pass option.
INCR	T2 PASS = OFF	
FLD SEL	T2 DELAY = OFF	Field select moves to the same delay that was set for trigger 1.
INCR	T2 DELAY = 00000 TRC	When setting the delay of trigger 1, we chose trace as the delay. Whatever is chosen by the count key will be the same delay for trigger 1 or trigger 2.
INCR	T2 DELAY = OFF	Again, the INCR key turns off the delay.
FLD SEL	T2 EVT ADDR = OFF	The field select now moves back to the beginning of the equation. Only trigger 1 sets up the mode of operation whether it be independent or not. Trigger 2 just sets up trigger 2 event.

BREAKPOINT

With events or triggers defined as above, these can be used as complex breakpoint definitions. Note the level of sophistication as compared to the simple breakpoint defined on the MicroSystem Emulator in the DISPLAY SELECT area.

Operation	Display	Explanation
BKPT	BKPT SEL = OFF	The breakpoint selects which trigger is going to be used to stop operation. It can be off.
INCR	BKPT SEL = T1	Or pressing the increment key select the T1 event. Notice also that trigger BREAK ENABLE LED is illuminated and the audible alarm sounds.
INCR	BKPT SEL = T2	The INCR key selects T2 instead of T1. The trigger BREAK ENABLE LED remains lighted.
INCR	BKPT SEL = T1 or T2	Now either trigger 1 or trigger 2 is chosen. The BREAK ENABLE LED is still illuminated.
INCR	BKPT SEL = OFF	Or turn off the breakpoint in which case the trigger BREAK ENABLE LED is extinguished.

REAL-TIME TRACE COLLECTION/DISPLAY

The actual Real-Time Trace is a 128x35 bit buffer that collects information during the running of the emulator. By judicious use of setting the breakpoints and use of the trace qualifier, the user can store the information that will provide the insight in finding the key to unlock circuit operation problems found during debug.

The trace stores the information found on the address bus the data bus, and the external probes at each bus cycle that is qualified to be stored in the bus.

Operation	Display	Explanation
TRC QUAL	TRC QUAL = ALL	We can choose what bus activity is to be stored in the 128-word trace memory. This is called the trace qualifier. If we choose all, that says any bus activity will be traced.
INCR	TRC QUAL = FTCH	Now only the instruction fetches will be traced. The programmer can look at his listing and trace his program flow.
INCR	TRC QUAL = ALL RD	Now we choose any read operation to be stored in the trace buffer. This is memory or I/O.
INCR	TRC QUAL = ALL WR	Now any write operation is chosen whether it be memory or I/O.
INCR	TRC QUAL = MEM RD/WR	This chooses any memory operation whether it be read or write.
INCR	TRC QUAL = MEM RD	This chooses just memory read operations to be traced.
INCR	TRC QUAL = MEM WR	This chooses just memory write operations to be traced.
INCR	TRC QUAL = I/O RD/WR	This chooses any I/O operation. This is a powerful feature and one which most trace units don't have.
INCR	TRC QUAL = I/O RD	This chooses just I/O reads.
INCR	TRC QUAL = I/O WR	This chooses I/O writes.

Operation	Display	Explanation
INCR	TRC QUAL = ALL	And again, we're back to ALL. In other words, anything that happens on the bus can be saved.
TRC DISP	TRACE BUFFER EMPTY	Pushing the trace display key allows interrogation of the saved data. If nothing has been saved, this message will be displayed.

ASCII CHARACTER SET

Table A-1 contains the 7-bit ASCII hexadecimal code for each character that can be viewed on the uSE front panel display.

Table A-1. ASCII Character Set

7-bit Hexadecimal Number	Character	7-bit Hexadecimal Number	Character
20	SPACE	40	@
21	!	41	A
22	RIGHT DOUBLE "	42	B
23	ALL SEGMENTS ON	43	C
24	$	44	D
25	%	45	E
26	u	46	F
27	'	47	G
28	(	48	H
29	)	49	I
2A	*	4A	J
2B	+	4B	K
2C	,	4C	L
2D	-	4D	M
2E	.	4E	N
2F	/	4F	O
30	0	50	P
31	1	51	Q
32	2	52	R
33	3	53	S
34	4	54	T
35	5	55	U
36	6	56	V
37	7	57	W
38	8	58	X
39	9	59	Y
3A	:	5A	Z
3B	;	5B	Σ
3C	o	5C	\
3D	=	5D	]
3E	" LEFT DOUBLE	5E	->
3F	?	5F	<-

KEY SUMMARY

KEY	DESCRIPTION	PAGE
ATTN	Allows the operator to regain control of the uSE when it is interfacing with a host computer or a master uSE.	3-23
BINARY	Displays data byte of the current display value in an 8-bit binary mode.	3-17
BKPT	Enables the user to set break points determined by the TRIGGER 1 and TRIGGER 2 equations. Options and Mode are selected by using the FLD SEL key.	3-28
BREAK	Displays and modifies the hardware breakpoint. Options are provided to select read and write, read only, or write only access to a specified memory address. Operator can establish a pass count for each breakpoint and/orjump to an address.	3-10
COUNT	Allows the counting of the various parameters up to a maximum value of 65,534 units.	3-29
DECR	Decrements the memory and I/O addresses.	3-17
DISABLE	Used with the BREAK key to clear breakpoint.	3-17
ENABLE	Used with the BREAK key to enable breakpoint.	3-17
ENTER	Enters display values and terminates the operation.	3-16
FILL	Allows the operator to fill RAM memory between two addresses with a specified data byte.	3-8
DATA ENTRY PAD	Contains the hexadecimal character set 0-9 and A-F. It is used to enter data and address values.	3-15
FLD SEL	Select options and parameters for TRIG 1, TRIG 2, and TRC QUAL keys.	3-29
I/O	Writes and reads the user's microprocessor I/O ports in hex or binary mode.	3-10
INCR	Increments the memory and I/O addresses, linked displays, and register content display.	3-17
MEMORY	Displays or modifies memory contents in hex or binary mode, or dump memory to the display terminal.	3-12
MOVE	Move data from user system RAM to uSE RAM, move uSE RAM to user RAM, move front panel PROM to uSE RAM, move front panel PROM to user RAM, move COMM/RAM to expansion RAM or expansion RAM to COMM/RAM.	3-24

KEY	DESCRIPTION	PAGE
MSG	Displays messages placed in the emulator display area by the diagnostic program. Messages for display are indicated on the front panel via the MESSAGE PENDING indicator. Multiple line messages are displayed with the INCR key.	3-14
PROM/MEM	Overlays front panel PROMs into the user's memory space.	3-4
RAM/MEM	Enables 8K internal RAM.	3-5
REGISTER	Displays or modifies register contents in hex or binary mode. Will also cause a register dump to the display terminal when the TERMINAL key is enabled.	3-13
REMOTE	Allows the uSE to operate as a remote slave unit controlled by a master uSE. Diagnostic programs can be down-loaded via telephone link from a master uSE or a host computer.	3-18
RESET	Forces a hardware reset on the emulator board microprocessor.	3-4
RESTART	Resets the uSE to initialized power on state and tests master RAM, emulator ROM, and system ROM.	3-2
RUN	Starts the execution of the user's microprocessor at a specified address contained in the PC register.	3-3
RUN/DISP	Runs and displays selected memory, registers, and I/O ten times per second.	3-3
SEND	Transmit data in uSE RAM or PROM out serial I/O port.	3-25
STEP	Allows single-step program execution of the user's program. Can also initiate and halt autostep.	3-3
SUBSEL	Enables special subtests if present in front panel PROMs.	3-8
TERMDISP	Displays the next type of data to be output to the display terminal.	3-29
TERMINAL	Enables or disables input/output to a display terminal attached to the uSE via the TERMINAL RS-232 port.	3-21
TEST	Performs a self-test of the uSE, checking RAM, ROM, front panel indicators and internal functions.	3-23
TRC DISP	Allows the user to display the contents of the Real-Time Trace Buffer.	3-28
TRC QUAL	Selects the type of transactions that are stored in the Real-Time Trace Buffer.	3-29
TRIG 1	Defines the trigger equations. When all elements are satisfied, the appropriate trigger pulse is provided on the uSE front panel trigger jack.	3-28

KEY	DESCRIPTION	PAGE
TRIG 2	Operates the same as the TRIG 1 key except for the Mode Parameter.	3-28
-->	Moves cursor left one position.	3-16
<--	Moves cursor right one position.	3-16
X	Enters "don't cares" into the current cursor position when appropriate.	3-16

ERROR MESSAGES

If there is an error on startup, the following message is displayed:

MESSAGE	DESCRIPTION
uSE CONDITION xxxx	xxxx = 0001 = Master RAM Failure
	0002 = Emulator ROM Failure
	0004 = Master ROM Failure
	0008 = Emulator not Responding
	0010 = RTT PROM Failure
	0020 = Shadow RAM Failure
	0040 = Comm/RAM Board, PROM Failure
	0080 = Comm/RAM Board, #1 RAM, Failure
	0100 = Extension RAM, #2 RAM, Failure
	0200 = RTT Board is not installed

The condition codes are reported in hexadecimal so that more than one condition may be reported. For instance, if the Master RAM and the Shadow RAM both failed then the message would be:

```
uSE CONDITION 0021
```

After checking all connections, if the error is not apparent, then call Customer Service at the toll-free hot-line numbers listed below.

National	(800) 538-9320/9321
California	(800) 662-9231

MESSAGE INFORMATION

DATA MESSAGES

Table D-1 shows which of the data can be received and transmitted by each of the remote modes.

Table D-1. Data Messages

MESSAGE	MODE							
	DOWNLOAD		SLAVE		MASTER		PASS	
TYPE	RCV	XMT	RCV	XMT	RCV	XMT	RCV	XMT
Keystroke			X			X		
LED				X	X			
Display				X	X			
Download	X	X	X	X	X	X		
HOST							X	X

KEYSTROKE FUNCTIONS

Table D-2 gives actual values of keystrokes.

Table D-2. Keystroke Functions

Key Type	Hex Value	Key Type	Hex Value
FLD SEL	00	"A"	1E
TRC DISP	01	<--	1F
COUNT	02	SUBSEL	20
TRIG 2	03	PROM/MEM	21
"3"	04	STEP	22
"2"	05	RUN	23
"1"	06	"F"	24
"0"	07	"E"	25
TERM DISP	08	"D"	26
TRC QUAL	09	ENTER	27
BKPT	0A	INCR	29
TRIG 1	0B	DECR	2A
"6"	0C	BINARY	2B
"5"	0D	REMOTE	2D
"4"	0E	TERMINAL	2E
"X"	0F	TEST	2F
"9"	14	ENABLE	31
"8"	15	DISABLE	32
"7"	16	ATTN	36
-->	17	REGISTER	42
FILL	18	BREAK	43
RAM/MEM	19	MOVE	45
RESET	1A	SEND	46
RUN/DISP	1B	I/0	47
"C"	1C	MSG	49
"B"	1D	MEMORY	4A

LED MESSAGES

Bit values of the LED display lights are shown in Table D-3.

Table D-3. LED Bit Values

Bit	LED	Bit	LED
	(Most significant bit)		
15	BEEP (Audible Alarm)	7	OPT MEM ENABLE
14	Unused (must = 0)	6	RUN
13	Unused (must = 0)	5	HALT
12	Unused (must = 0)	4	PROM MEM
11	Unused (must = 0)	3	BREAK
10	Unused (must = 0)	2	TRIG BKPT ENABLE
9	LOCAL	1	MESSAGE PENDING
8	REMOTE ENABLE	0	INPUT ERROR
			(least significant bit)

NOTE: 1 = On, 0 = Off

RECORD FORMATS

LED MESSAGE

TYPE
An exclamation mark (!) (21H) defines an LED message.

VALUE
Four ASCII hex digits representing the 16-bit value.

TEKSUM
Hex sum of the four digits VALUE field.

END-OF-LINE
ASCII carriage return (0DH) and optionally, a line feed (0AH).

TYPE	VALU	TEK-SUM	EOL	<-- FIELD
1	4	2	1-2	<-- NUMBER OF BYTES/FIELD

KEYSTROKE MESSAGE

TYPE
A number sign (#) (23H) defines a keystroke message.

KEYVALUE
Two ASCII hex digits. Send the register key.

TEKSUM
Hex sum of the two digit KEYVALUE field.

END-OF-LINE
ASCII carriage return (0DH) and optionally, a line feed (0AH).

TYPE	KEY-VALU	TEK-SUM	EOL	<-- FIELD
1	4	2	1-2	<-- NUMBER OF BYTES/FIELD

DISPLAY MESSAGE

TYPE
A double quote(") (22H defines a display message.

MESSAGE LENGTH
Number of characters in message.
Range = 1-20 (1-14H).

CURSOR CONTROL
00H = cursor off.
80H = cursor on.

CURSOR POSITION
Range = 0-20 (0-14H).

DECIMAL POSITION
Range = 0-20 (0-14H).

TEKSUM
Hex sum of LENGTH, CURSOR CONTROL, CURSOR POSITION and DECIMAL POSITION.

MESSAGE
ASCII message. 1-20 bytes.

ASUM
Hex sum of ASCII message.

END-OF-LINE
ASCII carriage return (0DH) and optionally, a line feed (0AH).

TYPE	MSG SIZE	CURS CNTL	CURS LOC	DEC LOC	TEK-SUM	1ST MSG BYTE	. . .	LAST MSG BYTE	ASUM	EOL	<-- FIELD
1	2	2	2	2	2	1 - 20			1	1-2	<-- NUMBER OF BYTES/FIELD

DOWNLOAD MESSAGE

START OF RECORD
a slash (/) (2FH) defines a download message.

LOCATION
Hex starting location of data in memory.

BYTE COUNT
Number of bytes in DATA (1-32 bytes).

TEKSUM 1
Hex sum of LOCATION and BYTE COUNT.

DATA
2-64 hex digits of user data, 2 ASCII digits per hex byte.

TEKSUM 2
Hex sum of DATA.

END-OF-LINE
ASCII carriage return (0DH) and optionally, a line feed (0AH).

START REC CHAR	LOC CNTR	BYTE COUNT	1ST TEK SUM	1ST DATA BYTE	. . .	LAST DATA BYTE	2ND TEK SUM	EOL CHARS	<-- FIELD
1	4	2	2	2 - 64			2	1-2	<-- NUMBER OF BYTES/FIELD

FILE TERMINATOR MESSAGE

TYPE
A slash (/) (2FH) defines a file terminator message (special case of the download message).

LOCATION
Memory location where user program begins execution.

BYTE COUNT
Set to 0 to indicate terminating record.

TEKSUM
Sum of LOCATION and BYTE COUNT.

END-OF-LINE
ASCII carriage return (0DH) and, optionally, a line feed (0AH).

TYPE	TRANS ADDR	BYTE COUNT	TEK SUM	EOL CHARS	<-- FIELD
1	4	2	2	1-2	<-- NUMBER OF BYTES/FIELD

PARTS LIST

Table F-1. Parts List Indexed by Description

PART NUMBER	DESCRIPTION
13000033	AC Distribution PCBA
15000149-01	DC Power Supply
13000044	Debug PCBA
13000017-04	Display Board PCBA
1300000-02	Keyboard PCBA
13-148-40	2708 Control Processor PCBA
13-148-41	2716 Control Processor PCBA
13000015-04	COMM/Ram PCBA
90010013-01 & -03	COMM/Ram Cable
90010022	34-Connect Emulator Cable
90010007	50-Connect Emulator Cable
20580460	40-Pin Augat Connector
15000156	Test Clip Adaptor Assembly
16000111	Test Clip Adaptor Contact.
90010047	RTT Cable
90010050	RTT Cable (Trig.)
15000170	RTT Probe Assembly
13000170	RTT PCBA
15000175	RTT Test Clip Kit
15000016	6800 Emulator I/F Assembly
13000005-02	6800 Emulator PCBA
15000016	6802 Emulator I/F Assembly
13000005-02	6802 Emulator PCBA
15000107	6802 Emulator Pod Assembly
15000106	6802 Emulator Probe Assembly
15000018	8080 Emulator I/F Assembly
13000012-02	8080 Emulator PCBA
15000109	8085 Emulator I/F Assembly
15000054	8085 Emulator Probe Assembly
13000108-02	8085 Emulator PCBA
15000099	Z80 Emulator I/F Assembly
15000078	Z80 Emulator Probe Assembly
13000098-02	Z80 Emulator PCBA
15000148	8048/8021 Emulator I/F Assembly
15000147	8048 Emulator Probe Assembly
15000179	8021 Emulator Probe Assembly
13000126	8048/8021 Emulator I/F PCBA
13000125	8048/8021 Emulator Slave PCBA

Table F-2. Parts List Indexed by Part Number

PART NUMBER	DESCRIPTION
13-148-40	2708 Control Processor PCBA
13-148-41	2716 Control Processor PCBA
1300000-02	Keyboard PCBA
13000017-04	Display Board PCBA
13000033	AC Distribution PCBA
13000044	Debug PCBA
15000149-01	DC Power Supply
13000015-04	Comm Ram PCBA
90010013-01 & -03	Comm Ram Cable
90010022	34-Connect Emulator Cable
90010007	50-Connect Emulator Cable
20580460	40-Pin Augat Connector
15000156	Test Clip Adaptor Assembly
16000111	Test Clip Adaptor Contact.
90010047	RTT Cable
90010050	RTT Cable (Trig.)
15000170	RTT Probe Assembly
13000170	RTT PCBA
15000175	RTT Test Clip Kit
15000016	6800 Emulator I/F Assembly
13000005-02	6800 Emulator PCBA
15000016	6802 Emulator I/F Assembly
13000005-02	6802 Emulator PCBA
15000107	6802 Emulator Pod Assembly
15000106	6802 Emulator Probe Assembly
15000018	8080 Emulator I/F Assembly
13000012-02	8080 Emulator PCBA
15000109	8085 Emulator I/F Assembly
15000054	8085 Emulator Probe Assembly
13000108-02	8085 Emulator PCBA
15000099	Z80 Emulator I/F Assembly
15000078	Z80 Emulator Probe Assembly
13000098-02	Z80 Emulator PCBA
15000148	8048/8021 Emulator I/F Assembly
15000147	8048 Emulator Probe Assembly
15000179	8021 Emulator Probe Assembly
13000126	8048/8021 Emulator I/F PCBA
13000125	8048/8021 Emulator Slave PCBA

WARRANTY AND SERVICE PROGRAM

Millennium Systems products were designed and manufactured to strict performance standards and tested to criteria sufficient to provide the user with trouble-free service. Concurrently our customer service policy is designed to provide our customers with prompt, effective repair services both during and after the warranty period.

WARRANTY

The Seller warrants that the products sold hereunder will, at time of shipment be free from defects in material and workmanship and will conform to the Seller's specifications, or as appropriate to the Buyer's specifications, which have been accepted in writing by the Seller for a period of 120 days from date of initial shipment. Seller's obligation for any breach of this warranty shall be limited, at Seller's election, to either refunding the purchase price of the defective products or repairing or replacing the same; provided that Buyer shall have (1) given written notice of nonconformance to Seller within 120 days from date of initial shipement, and (2) obtained a Customer Return Material Authorization Number for non-conforming products, and returned such products FOB Seller's plant or authorized repair center within thirty (30) days from expiration of said 120-day period. In no event shall Seller have any obligation hereunder for defects in respect of which Seller has not received written notice within said 120-day period. The warranty shall not apply to any products which Seller determines have, by Buyer or otherwise, been subjected to operating and/or environmental conditions in excess of the maximum values established therefor in the applicable specifications or otherwise have been subjected to mishandling, misuse or neglect, or to any repair, alteration or use which alters the physical or electrical properties of said products. Transportation charges for the return to the Buyer shall be paid by Seller within the contiguous 48 United States. If Seller determines that the products are not defective, Buyer shall pay Seller all costs of handling and transportation. The warranty outside the contiguous United States excludes all costs of shipping, customs clearance and related charges.

Seller may repair or replace returned products at their sole option. Such repair or replacement shall not extend the warranty period. Buyer's remedies are expressly limited to Seller's obligations as stated in this paragraph.

In no event will Seller be liable to Buyer or to any customer of Buyer for any incidental or consequential damages or other commercial loss of whatever nature and however occasioned.

THIS WARRANTY EXTENDS TO BUYER ONLY AND NOT TO BUYER'S CUSTOMERS OR USERS OF BUYER'S PRODUCTS AND IS IN LIEU OF ALL OTHER WARRANTIES WHETHER EXPRESS, IMPLIED, OR STATUTORY INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

SERVICE PROGRAM

Should service be required, Millennium Systems Customer Service Program is easy to use. Just follow these steps·

1. Call one of Millennium's hot-lines listed in the preface and ask for the Customer Service Manager who will assist you in providing spare parts and service. Be prepared to provide the Model Number, Serial Number, date of purchase and description of the failure.

2. The Customer Service Manager will issue (if necessary) a Return Material Authorization Number, which would be included on any paperwork accompanying a return. Be sure to pack the defective part securely. Include the RMA number, the name and phone number of the person requesting the repair and ship it prepaid to Millennium Systems.

3. The Customer Service Manager will ask you for a Purchase Order Number (or certified check) for a new or equivalent to new replacement part which Millennium Systems will ship within 48 hours freight prepaid (in contiguous 48 states).

4. If the returned part meets the conditions stated in the warranty, then Millennium Systems will issue a full credit against your Purchase Order. If the warranty has expired or other warranty conditions have not been complied with, Millennium will invoice you for the difference between the spare part price less the return allowance.

A

B

C

D

E

F

G

H

I

K

L

M

O

P

R

S

T

U

W

DOCUMENT REVIEW FORM

MICROSYSTEM EMULATOR
USERS MANUAL

Publication No. 87000040
February 1981
Release 2.0

Dear Customer:

Your comments concerning this document help us to produce better documentation for you.

GENERAL COMMENTS

- [ ] Easy to read?
- [ ] Well organized?
- [ ] Accurate?
- [ ] Complete?
- [ ] Well illustrated?
- [ ] Suitable for your needs?

SPECIFIC COMMENTS AND CORRECTIONS

Reference	Page

Please send your comments to:

MILLENNIUM SYSTEMS, INC.
19050 Pruneridge Avenue
Cupertino, Calif. 95014
ATTN: Publications

MILLENNIUM

A subsidiary of American MicroSystems, Inc.

MILLENNIUM SYSTEMS INC.
19050 Pruneridge Ave., Cupertino, CA 95014
Telephone: (408) 996-9109
TWX # is 910-338-0256

Publication #87000040/R3.0