HEWLETT PACKARD

Installation/Service/Terminal Interface Guide

HP 64782
MC6833x Emulator/Analyzer

Notice

Printing History

New editions are complete revisions of the manual. Many product updates and fixes do not require manual changes, and manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual revisions.

Edition 1	64782-97000, August 1994
Edition 2	64782-97001, September 1996
Edition 3	64782-97002, November 1996
Edition 4	**64782-97003, April 1997**

Safety and Certification and Warranty

Safety information, and certification and warranty information can be found at the end of this manual on the pages before the back cover.

HP 64748C
Emulation Control Card

Egress Panel
(Part of HP 64748C)

Three Unique Cables
(Part of HP 64748C)

External
Power Cable
(Part of
HP 64782)

HP 64782 Probe

Demo Board
(Part of 64782)
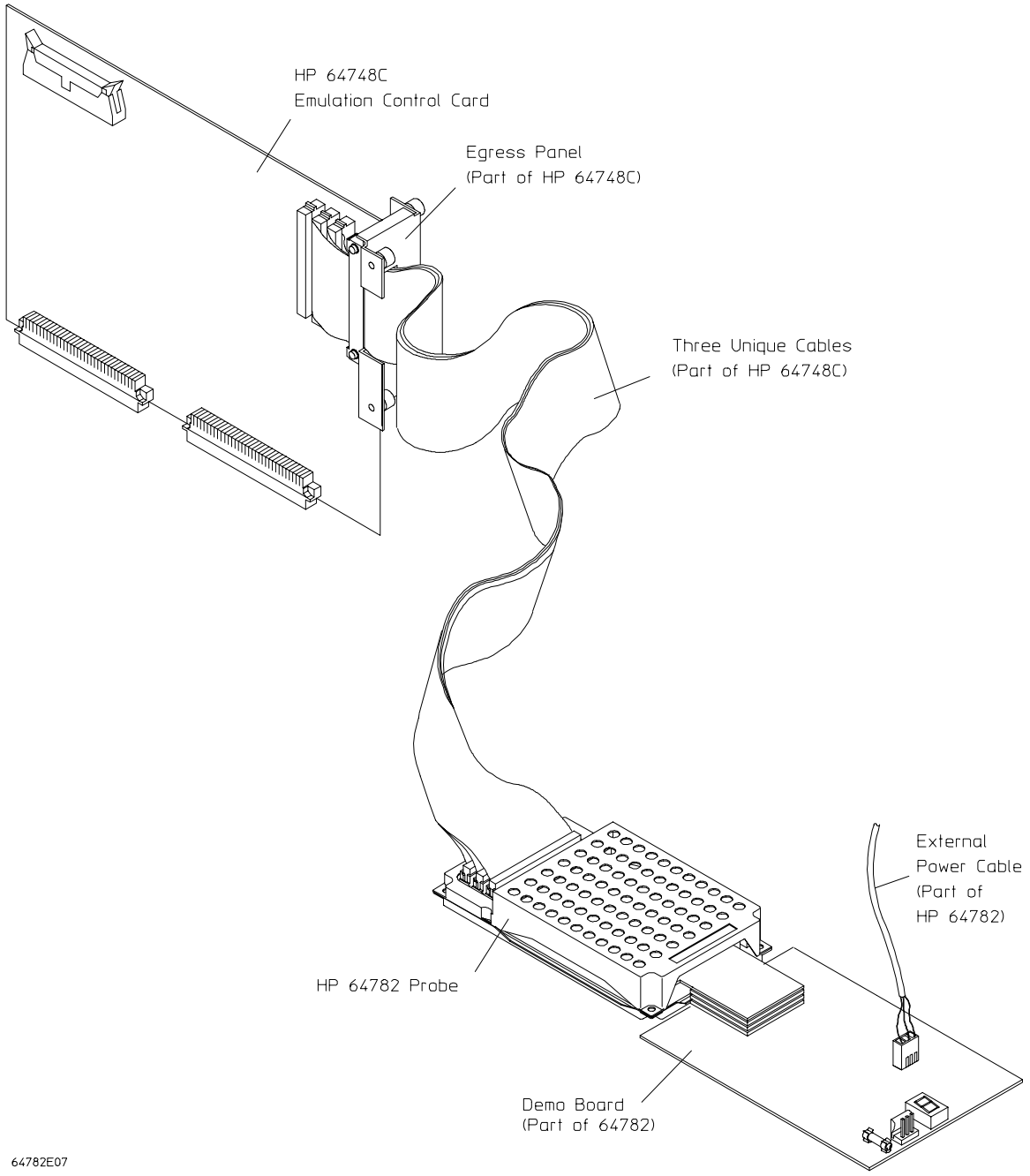
64782E07

The HP 64782 Emulator

Description

The Hewlett-Packard Model 64782 emulator supports the Motorola 6833x microprocessor family. The emulators and supported microprocessors include HP 64782B/MC68331, HP 64782C/MC68332, HP 64782E/MC68334, HP 64782G/MC68336, and HP 64782J/MC68338. In this manual, the HP emulators are referred to by Model Number 64782, and the Motorola microprocessors are referred to by MC6833x, except where specific information applies only to certain individual members of the emulator/microprocessor family.

The emulator supports clock speeds up to 33 MHz.

The emulator supports both 5V and 3.3V operation. The emulator has a PGA-style connector. It can be plugged directly into the PGA connector on the demo board, and it can be plugged into a PQFP or TQFP target system using optional accessories.

The emulator plugs into the modular HP 64700 instrumentation card cage and offers 64 channels of processor bus analysis with the HP 64794A or HP 64704A emulation-bus analyzer. Up to eight megabytes of emulation memory may be installed on the probe. High performance download is achieved through the use of an optional LAN or RS-422 interface. A pair of RS-232 ports and a firmware-resident interface allow debugging of a target system at remote locations.

For software development, the HP AxCASE environment is available on SUN SPARCsystems and on HP workstations. This environment includes an ANSI standard C compiler, assembler/linker, a debugger, the HP Software Performance Analyzer that allows you to optimize your product software, and the HP Branch Validator for test suite verification. The C compiler, assembler/linker, and debugger are also available for MS-DOS systems.

Language support is also available from several third-party vendors. This capability is provided through the HP 64700's ability to consume several industry standard output file formats.

Ada language support is provided on HP 9000 workstations by third-party vendors such as Alsys and Verdix. An Ada application developer can use the HP emulator and any compiler that generates HP/MRI IEEE-695 to do exhaustive, real-time debugging in-circuit or out-of-circuit.

Features

HP 64782 Emulator

- Active probe emulator; supports the fastest processor speeds currently available. At the time this manual was printed, the emulator had been tested to the following processor speeds:
  – 20 MHz for MC68331, MC68332, and MC68334.
  – 16 MHz for MC68336 and MC68338.
- 5V and 3.3V operation
- No wait states to target memory
- Fast termination cycles to target memory
- Unlimited software breakpoints
- Symbolic support
- 36 inch cable and 219 mm (8.8") x 102 mm (4") probe
- Background and foreground monitors
- Simulated I/O with workstation interfaces
- Consumes IEEE-695, HP-OMF, Motorola S-Records, and Extended Tek Hex File formats directly. (Symbols are available with IEEE-695 and HP-OMF formats.)
- Multiprocessor emulation
  – synchronous start of 32 emulation sessions
  – cross triggerable from another emulator, logic analyzer, or oscilloscope
- Demo board and self test module included

Emulation-bus analyzer

- 64-channel emulation-bus analyzer
- Post-processed dequeued trace with symbols
- Eight events, each consisting of address, status, and data comparators
- Events may be sequenced eight levels deep and can be used for complex trigger qualification and selective store

Emulation memory

- Up to 8 Mbytes of emulation memory
- All emulation memory is dual-ported
- Mapping resolution is 256 bytes
- The wait states required for emulation memory depend on processor speed and the access speeds of the SIMMs installed on the emulation probe, as shown below:

SIMM Access Speed	Runs with no Wait States	Requires 1 Wait State
35 nSEC	0 to 16.78 MHz	16.78 MHz to 25 MHz
20 nSEC	0 to 25 MHz	25 MHz to 33 MHz
25 nSEC	0 to 22 MHz	22 MHz to 30 MHz

- Fast termination cycles to emulation memory are supported up to 10 MHz when using 20-nSEC and 25-nSEC SIMMs, and supported up to 8 MHz when using 35-nSEC SIMMs

In This Book

This manual covers the Hewlett-Packard Model 64782 emulator for the Motorola MC6833x family of microprocessors. It is divided into the following parts:

Part 1, "Installation," shows you how to install and connect all of the emulator/analyzer hardware in the card cage, connect the card cage to a host computer in several typical configurations, and connect the emulator probe into the demo board and into your target system.

Part 2, "Service," shows you how to install and update emulator/analyzer firmware, solve problems you may encounter while using the emulator/analyzer, and obtain replacement parts for the emulator/analyzer from Hewlett-Packard.

Part 3, "Terminal Interface Reference," introduces the interfaces available to use with the emulator/analyzer, and shows you how to use the terminal interface for the emulator/analyzer.

This manual replaces the Terminal Interface Reference which accompanied earlier HP emulators. For information on the Terminal Interface, see Part 3.

You should read the book *Concepts of Emulation and Analysis* when you have the chance to do so; it contains a good conceptual introduction to the emulation process, and also describes how an emulation monitor works. Another book, the *HP 64700 Card Cage Installation/Service Guide*, tells you more about installation and configuration of the HP 64700 Card Cage. If you have a problem with the emulator and don't understand how to fix it, a listing of HP Sales and Service offices is in the *Support Services Guide* in the back of this binder.

Contents

Contents

Part 1

Installation Guide

1

Preparing the Emulator

How to connect the emulator probe and power cord.

Preparing the Emulator

This chapter shows you how to install the emulation and analysis hardware in the emulator card cage.

The installation tasks are described in the following steps:

1 Install optional memory modules on the deep analyzer card, if desired.

2 Connect the HP 64782 emulator probe to the HP 64748C emulator control card.

3 Install cards into the HP 64700 card cage.

4 Install emulation memory modules on the emulator probe.

5 Connect a power cord to the HP 64700 Card Cage.

What you need

Equipment supplied

The minimum system contains:

- HP 64782 6833x PGA Emulator Probe
- Demo target system (shipped with the emulator probe).
- HP 64748C Emulation Control card.
- HP 64794A/C/D Emulation-Bus Analyzer (deep analyzer) card, or
  HP 64704A Emulation-Bus Analyzer (1K analyzer) card.
- HP 64700A or HP 64700B Card Cage.

Optional parts are:

- HP 64171A/64172A 256-Kbyte Memory Modules for additional memory
  depth.
- HP 64171B/64172B 1-Mbyte Memory Modules for additional memory depth.
- HP 64173A 4-Mbyte Memory Modules for additional memory depth.
- HP 64708A Software Performance Analyzer.
- HP 64701A LAN Interface card (HP 64700A only).

Equipment and tools needed

In order to install and use the MC6833x emulation system, you need:

- Flat-blade screwdriver with shaft at least 5 inches long (13 mm approx).
- Torx T-10 screwdriver (if installing the optional LAN card for the
  HP 64700A).

The illustrations in this manual show the HP 64700B Card Cage. The locations of
some components may be slightly different if you are using an HP 64700A Card
Cage.

Antistatic precautions

Printed-circuit boards contain electrical components that are easily damaged by small amounts of static electricity. To avoid damage to the emulator boards, follow these guidelines:

- If possible, work at a static-free workstation.
- Handle the boards only by the edges; do not touch components or traces.
- Use a grounding wrist strap that is connected to the HP 64700 chassis.

Note

If you already have a modular HP 64700-Series Card Cage and want to remove the existing emulator and insert an HP 64782 emulator in its place, the HP 64700-Series generic firmware and analyzer firmware may NOT be compatible, and the software will indicate incompatibility. In this event, you must purchase a Flash EPROM board to update the firmware. Instructions for installing this board and programming it from a PC or HP 9000 are provided in the HP 64700 Card Cage Installation/Service manual. Instructions for installing and updating emulator firmware are covered in Chapter 5, "Installing/Updating Emulator Firmware" in this manual.

Note

If you already have a modular HP 64700-Series Card Cage and want to remove the 1K analyzer and install the deep analyzer in its place, the analyzer firmware will be updated by your installation because the analyzer firmware is contained on the analyzer card.

Step 1. Install optional memory modules on Deep Analyzer card (if using the Deep Analyzer)

Observe antistatic precautions

With no optional memory modules installed on the deep analyzer card, the trace memory depth is 8K. If you are going to use the deep analyzer with this default trace memory depth, skip this step.

1 Determine placement of the optional memory modules. Two types of modules may be installed: 256-Kbyte (HP 64172A), and 1-Mbyte (HP 64172B). Either module type may be installed in the banks on the analyzer card. Do not use HP 64171A/B or HP 64173A memory modules; they are too slow.

If you install no memory modules, the deep analyzer will have 8K maximum memory depth.
If you install four 256-Kbyte memory modules, the analyzer will have 64K maximum memory depth.
If you install four 1-Mbyte memory modules, the analyzer will have 256K maximum memory depth.

If you install a combination of 256-Kbyte memory modules and 1-Mbyte memory modules, the analyzer will have 64K maximum memory depth. All four connectors must have memory modules installed before the analyzer depth will be increased.

2 To ensure correct installation of optional memory modules on the deep analyzer card, there is a cutout at one end of the memory modules so they can only be installed the correct way.

To install a memory module:

Align the groove in the memory module with the alignment rib in the connector.

Align the cutout in the memory module with the projection in the connector.

Place the memory module into the connector groove at an angle.

Firmly press the memory module into the connector and make sure it is completely seated.

Rotate the memory module forward so that the pegs on the connector fit into the holes on the memory module.

Make sure the release tabs at each end of the connector snap around the memory module to hold it in place.

Groove in Memory Module and Alignment Rib in Connector

Cutout in Memory Module

Release Tabs
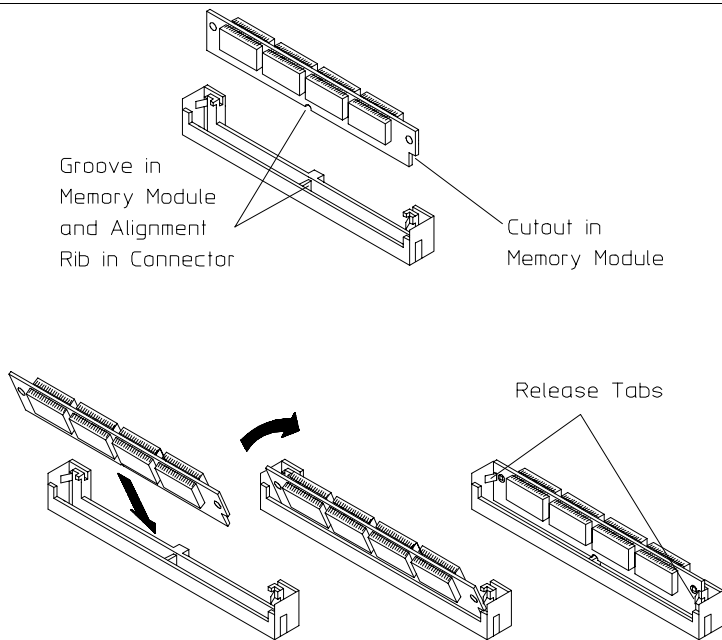
64794E03

Step 2. Connect the Emulator Probe Cables

Three ribbon cables connect the HP 64748C emulation control card to the HP 64782 emulator probe.

The shortest cable connects from J1 of the emulation control card to J3 of the emulator probe. The medium length cable connects from J2 of the emulation control card to J2 of the emulator probe. The longest cable connects from J3 of the emulation control card to J1 of the emulator probe.

Make sure the cable connectors are seated. There are stainless steel clips on the cable connectors; these must be properly latched inside the sockets. Otherwise, the cables will work loose and you will see erratic operation. See illustration next page (step 2).

1 Connect the emulator probe cables to the emulation control card.
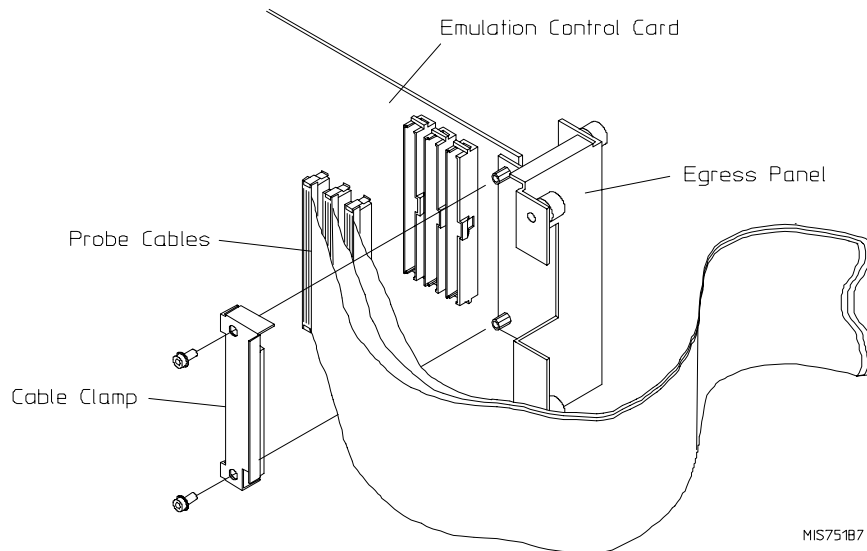
Emulation Control Card

Egress Panel

Probe Cables

Cable Clamp

MIS751B7

2 When inserting cable connectors into the sockets, press inward on the connector clips so that they hook into the sockets as shown. The order of connecting cables was given in step 1.

Push in on clips
so they hook into
socket
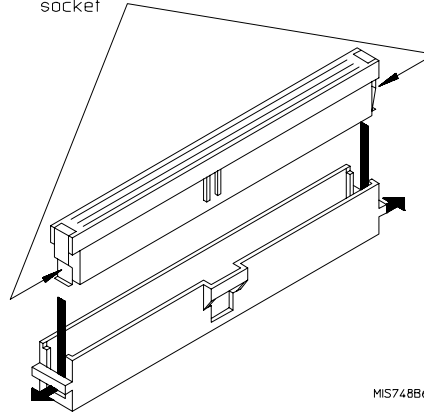
MIS748B6

3 Connect the other ends of the cables to the emulator probe. Make sure the stainless steel clips on the cable connectors are properly latched within the sockets.

Probe Cables

Top Plastic Cover

Bottom
Plastic Cover

Active Probe

Demo Board
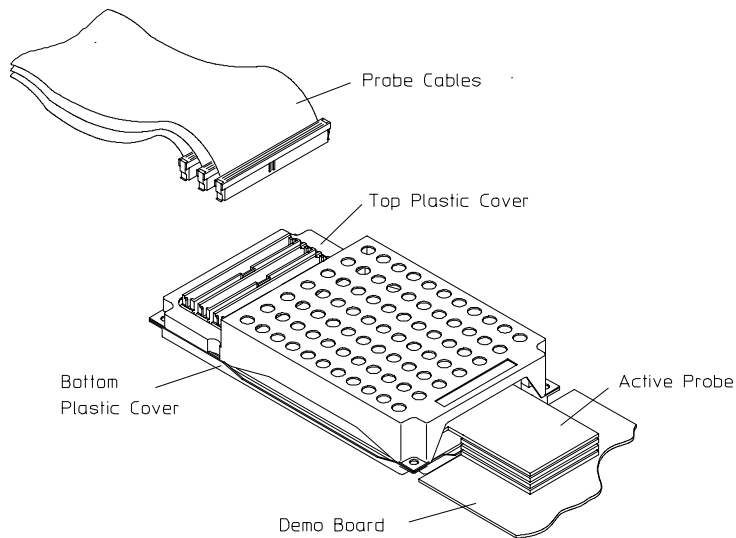
64782E04

Step 3. Install Boards into the HP 64700 Card Cage

WARNING   **Before removing or installing parts in the HP 64700 Card Cage, make sure that the card cage power is off and that the power cord is disconnected.**

CAUTION   Do NOT stand the HP 64700 Card Cage on the rear panel. You could damage the rear panel ports and connectors.

---

1 Use a ground strap when removing or installing boards into the HP 64700 Card Cage to reduce the risk of damage to the circuit cards from static discharge. A jack on the rear panel of the HP 64700 Card Cage is provided for this purpose.

Ground strap plug

64700E07

---

2 Turn the thumb screw and remove the top cover by sliding the cover toward the rear and up.

Loosen thumb screw
and slide cover
to remove.

64700E08

3 Remove the side cover by unsnapping the two latches and lifting off.

Emulator Side Cover

Latches
(On Bottom Panel)

Tab Slots

64783E08
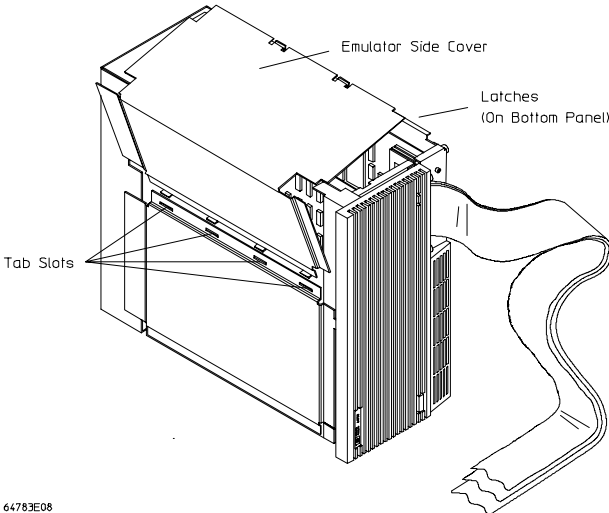
4 Remove the card supports.

Number here
indicates slot 1

Card supports

64700E01

5 First, completely loosen the four egress thumb screws.

To remove emulator cards, insert a flat blade screwdriver in the access hole and eject the emulator cards by rotating the screwdriver.

Emulator Card

Four Egress
Thumb Screws

Probe Cables

With flat blade screwdriver
eject emulator card, Egress
and probe cable as an assembly

64783E06

6 Insert a screw driver into the third slot of the right side of the front bezel, push to release catch, and pull the right side of the bezel about one-half inch away from the front of the HP 64700. Then, do the same thing on the left side of the bezel. When both sides are released, pull the bezel toward you approximately 2 inches.

Be careful because the plastic ears are easily broken on the front bezel.

Front Panel
Without Bezel
Showing Catch

Insert screw driver into third
slot of front bezel, push to
release catch and pull bezel
toward you.

MIS700B9
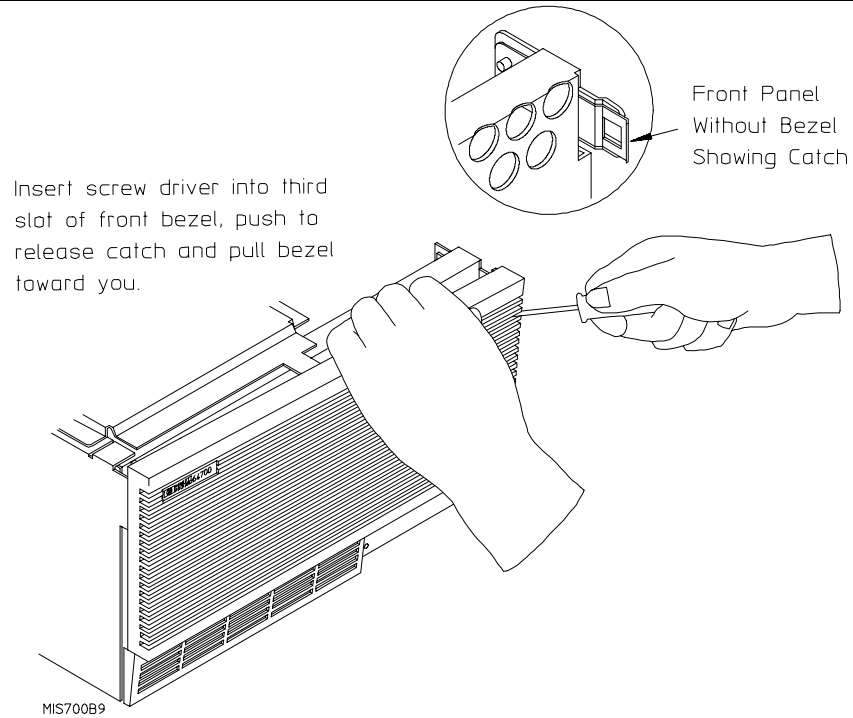
7 Lift the bezel panel to remove. Be careful not to put stress on the power switch extender.

Lift bezel panel and
tip toward you to
remove.
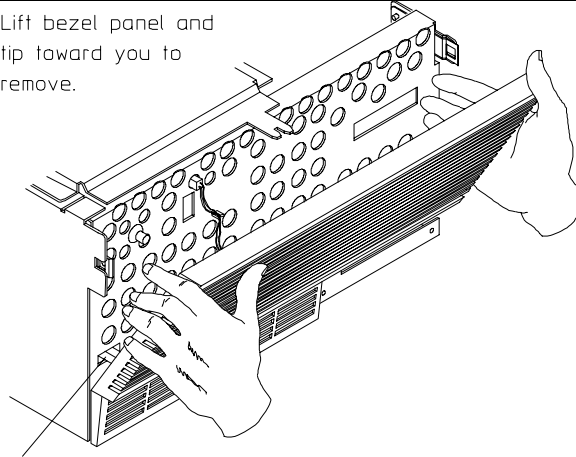
Be careful not to
put stress on power
switch extender.

MIS70011

8 If you're removing an existing analyzer card that provides external analysis, remove the right-angle adapter board by turning the thumb screws counterclockwise.
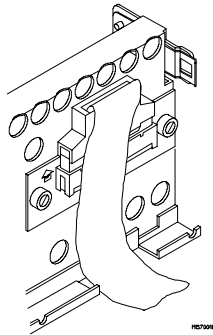
9 To remove the analyzer card, insert a flat blade screwdriver in the access hole and eject the analyzer card by rotating the screwdriver.

MIS76016

Eject Analyzer Card
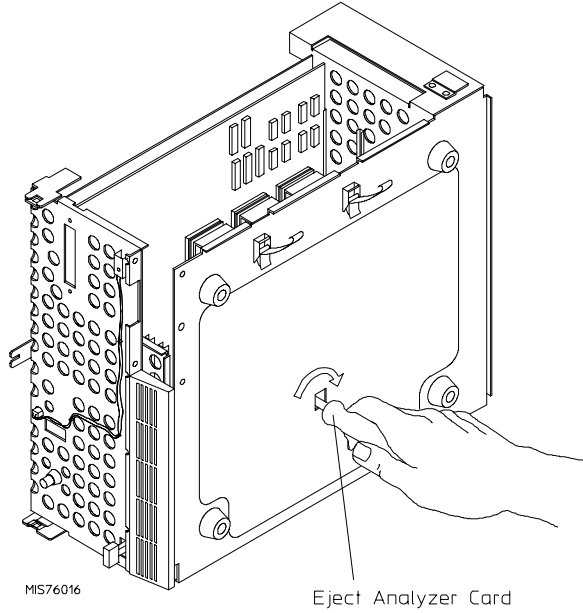
Do not remove the system control board. This board is used in all HP 64700 emulation and analysis systems.

10 Install the analyzer and emulation control cards. The analyzer is installed in the slot next to the system control card. The emulation control card is installed in the second slot from the bottom of the card cage. The software performance analyzer card may occupy any slot between the emulation-bus analyzer and the emulation control card. These cards are identified with labels that show their model numbers and serial numbers.   Note that components on the analyzer card face the opposite direction to the other cards.

To install a card, insert it into the plastic guides. Make sure the connectors are properly aligned; then, press the card into the mother board socket. Ensure that each card is seated all the way into its socket. If the cards can be removed with your fingers, the cards are NOT seated all the way into the mother board sockets.

Attach the ribbon cable from the emulation control card to the analyzer card, and to the software performance analyzer, if installed. Tighten the thumbscrews that hold the emulation control card to the cardcage frame.
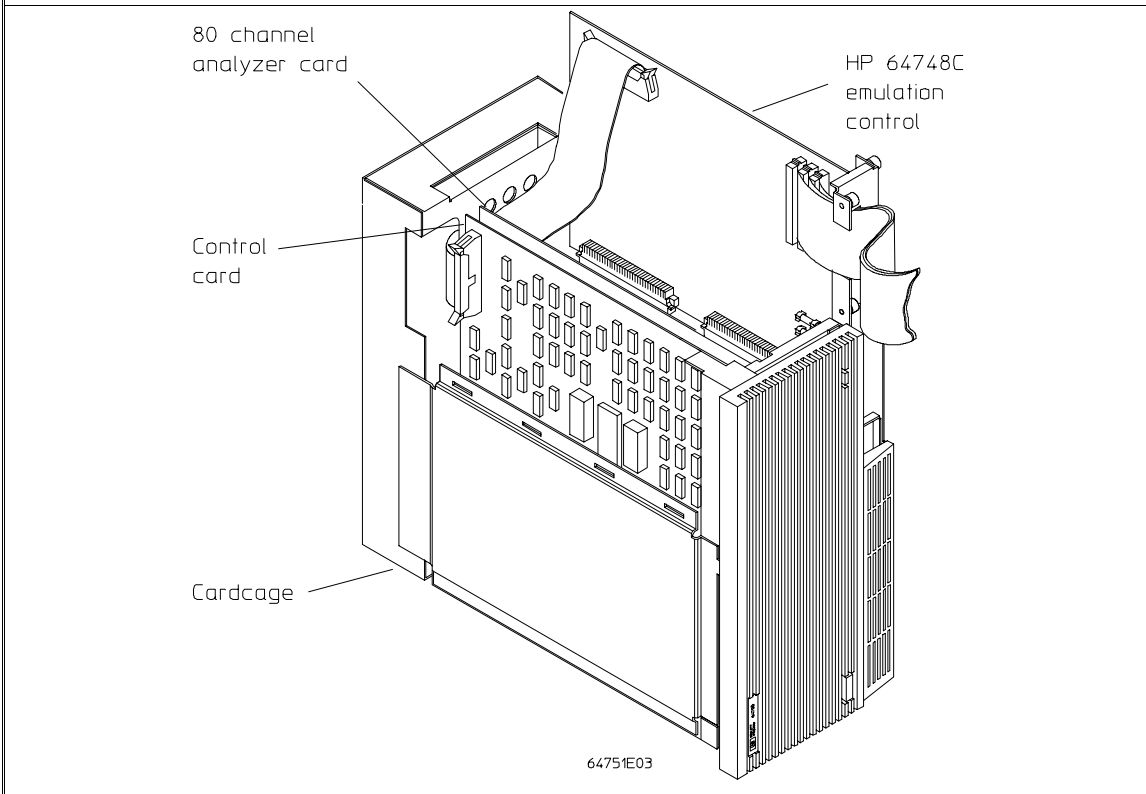
64751E03

11 Connect the +5 V power cable to the connector in the HP 64700 front panel.

Power Connection
for Demo Board

MIS76021
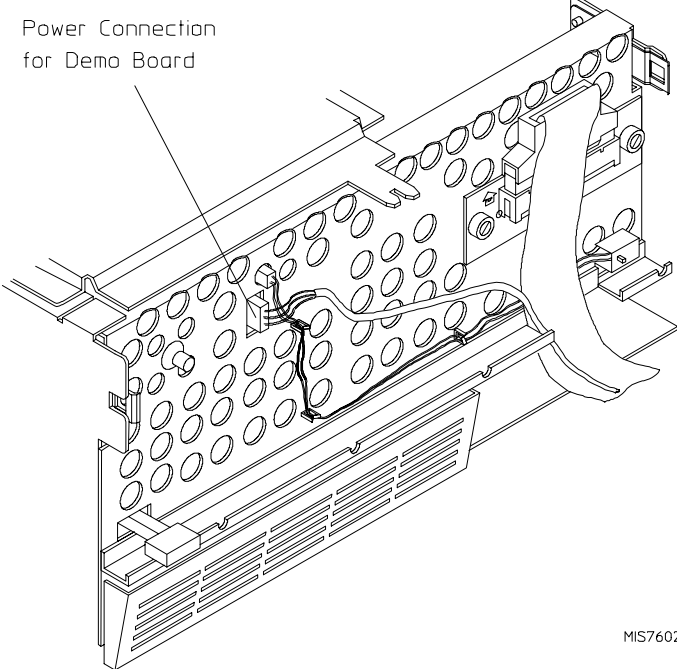
12 To reinstall the front bezel, be sure that the bottom rear groove of the front bezel is aligned with the lip as shown below.

Be sure back groove
of bezel is aligned
with lip.

Push front bezel
into place.
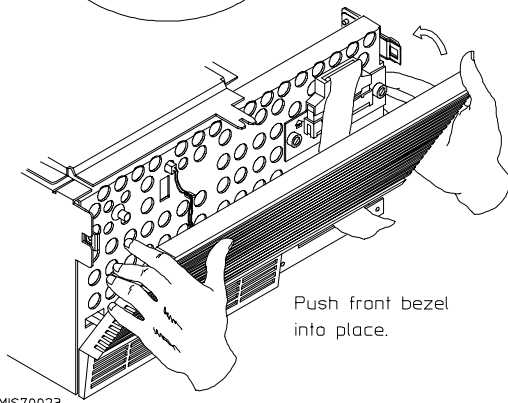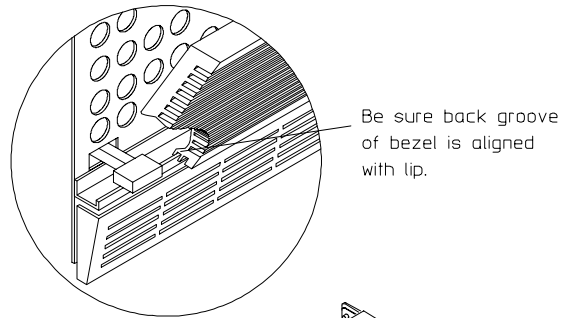
MIS70023

13 This step applies only to the HP 64700A. The HP 64700 has a built-in LAN interface.

Before you install the LAN card, you must remove the rear panel cover plate that is replaced by the bracket shown in the figure below.

To install the LAN card, position the BNC and 15-pin connectors of the LAN card through the openings in the rear panel, press the card into the mother board socket, secure the card with the two Torx T-10 screws, and mount the bracket to the rear panel with the four Torx T-10 screws.
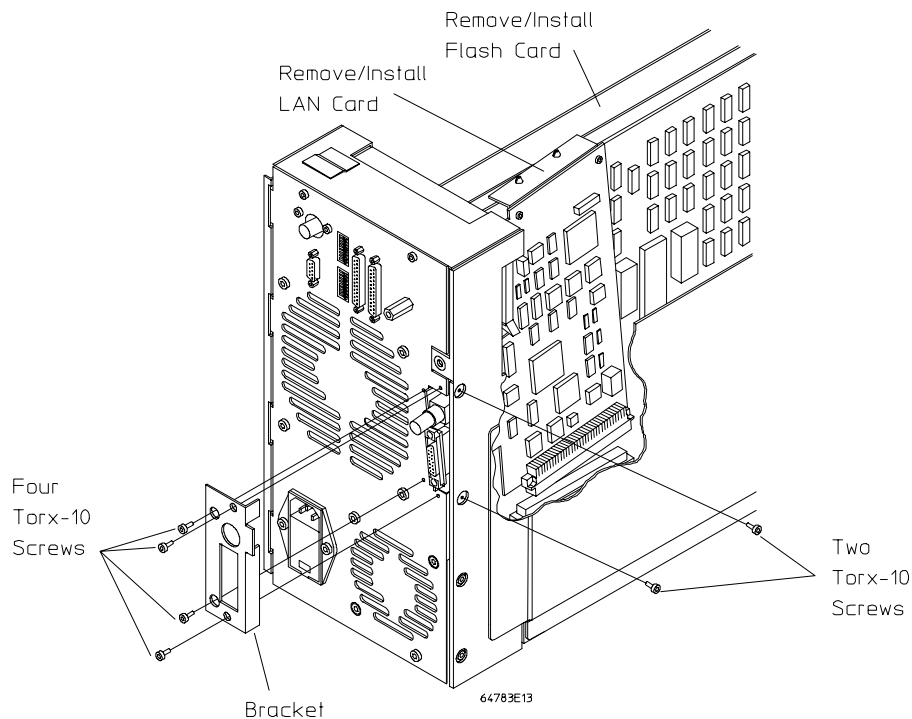
14 This step applies only to the HP 64700A. If you wish to install the flash card (used for updating firmware, see chapter 5), refer to the diagram above. Install the flash card in any available slot between the 80-channel analyzer card and the HP 64748C control card in the cardcage. Insert the flash card in the plastic guides. Make sure the connectors are properly aligned. Then press the card into the mother board sockets. Make sure the card is seated all the way into the sockets.

15 Install the card supports.
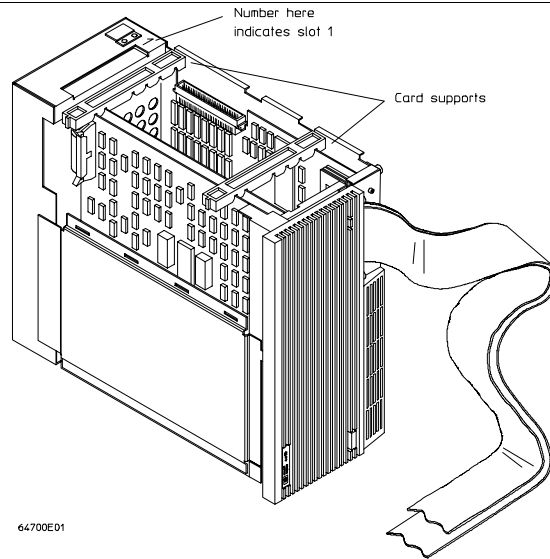
16 To install the side cover, insert the side cover into the tab slots and fasten the two latches.
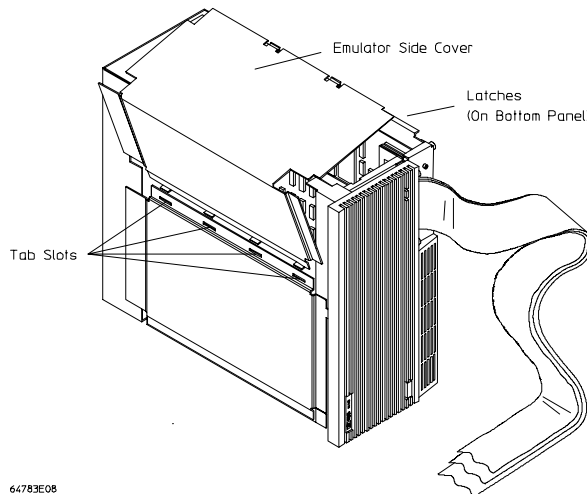
17 Install the top cover in reverse order of its removal, but make sure that the side panels of the top cover are attached to the side clips on the frame.

64700E09

Side clip
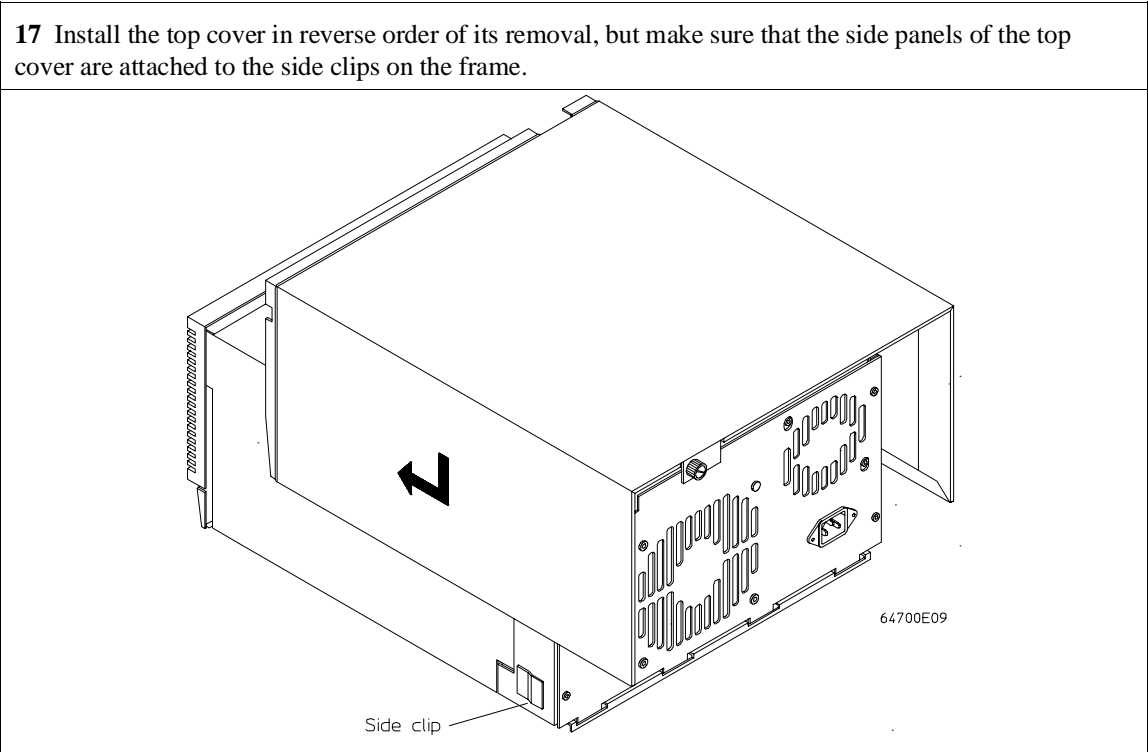
Step 4. Install emulation memory modules on emulator probe

(Observe antistatic precautions)

1 Remove plastic rivets that secure the plastic cover on the top of the emulator probe, and remove the cover. The bottom cover is only removed when you need to replace a defective active probe on the exchange program.

To install rivet
push down on rivet
head

To remove rivet
push up on center
shaft

MIS74810

Memory Bank 1

Memory Bank 0

Add Plastic
Washers to
Four Positions

64782E03

2 Determine the placement of the emulation memory modules. Three types of modules may be installed: 256 Kbyte (HP 64171A/64172A), 1 Mbyte (HP 64171B/64172B), and 4 Mbyte (HP 64173A). Any of the emulation memory modules can be installed in either memory slot on the probe.

Memory in both memory slots is divided into four equal blocks that can be allocated by the memory mapper.

3 Install emulation memory modules on the emulator probe. There is a cutout at one end of the memory modules so they can only be installed the correct way.

To install a memory module:

1 Align the groove in the memory module with the alignment rib in the connector.
2 Align the cutout in the memory module with the projection in the connector.
3 Place the memory module into the connector groove at an angle.
4 Firmly press the memory module into the connector and make sure it is completely seated.
5 Rotate the memory module to the vertical position so that the pegs on the connector fit into the holes on the memory module.
6 Make sure the release tabs at each end of the connector snap around the memory module to hold it in place.

64794E03

4 Replace the plastic cover, and insert new plastic rivets (supplied with the emulator) to secure the cover.

To install rivet
push down on rivet
head

To remove rivet
push up on center
shaft

MIS74810

Memory Bank 1

Memory Bank 0

Add Plastic
Washers to
Four Positions

64782E03
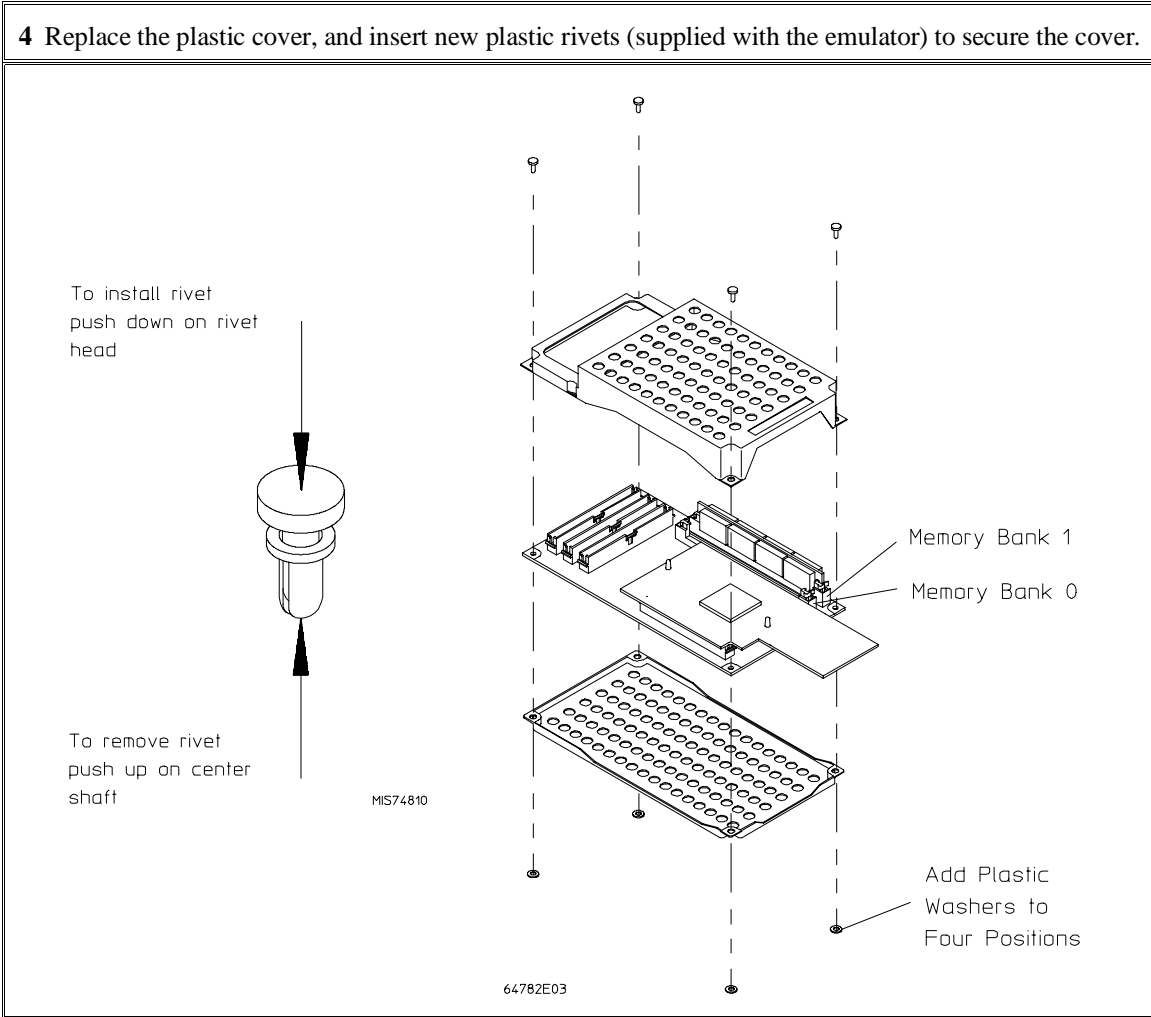
Step 5. Connect the power cord

The HP 64700B automatically selects the 115 Vac or 220 Vac range. In the 115 Vac range, the HP 64700B will draw a maximum of 345 W and 520 VA. In the 220 Vac range, the HP 64700B will draw a maximum of 335 W and 600 VA.

If you have the emulator installed in an HP 64700A card cage, select the line voltage using a thumb-wheel switch inside the power control module on the rear panel of the card cage. The input frequency must be in the range of 48 to 66 Hz. At 115 VAC, the emulator will draw a maximum of 3.0 A. At 220 VAC, it will draw a maximum of 1.5 A.

The HP 64700 is shipped from the factory with a power cord appropriate for your country. You should verify that you have the correct power cable for installation.

If the cable you received is not appropriate for your electrical power outlet type, contact your Hewlett-Packard sales and service office.
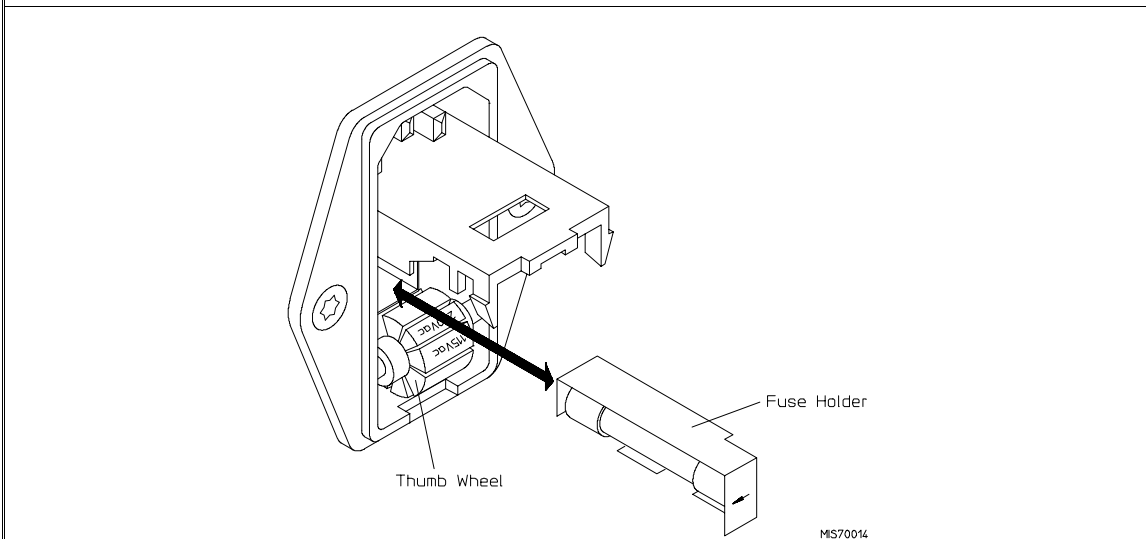
Fuse Holder

Thumb Wheel

MIS70014

1 Connect the power cord and turn on the HP 64700.

The line switch is a pushbutton located at the lower, left-hand corner of the front panel. To turn ON power to the HP 64700, push the line switch button in to the ON (1) position. The power lamp at the lower, right-hand corner of the front panel will light.

2
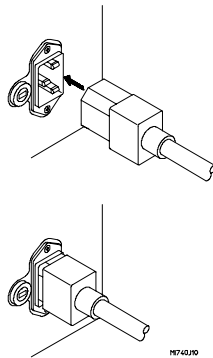
Connecting to a Host Computer

How to connect the emulator to a workstation, PC or terminal.

As you follow the steps in this chapter, you will need to refer to the *64700 Card Cage Installation/Service Guide*.

Step 1: Choose a system configuration

1 Decide how you will connect the emulator to your host computer. Refer to the "Concepts" chapter in the *64700 Card Cage Installation/Service Guide*.

2 If you will be using a LAN connection, continue with the steps in this chapter.

If you will be using a serial connection, refer to the information in the *64700 Card Cage Installation/Service Guide*; you may skip the rest of this chapter.

Step 2: Connect the LAN cable

- Connect the LAN to either the BNC connector or the 15-pin AUI connector.

The card cage can communicate with computers on an IEEE 802.3 or Ethernet Local Area Network. (If you have a 64700A card cage, you need the HP 64701A LAN card to connect to a LAN.) You can use either of two LAN connectors:

- A BNC connector that can be directly connected to a ThinLAN (HP's implementation of IEEE 802.3 Type 10BASE2) cable. The card cage provides the functional equivalent of a Medium Attachment Unit (MAU) for ThinLAN.

- A 15-pin connector for an Attachment Unit Interface (AUI) cable. The AUI cable allows you to connect to an off-board MAU for ThinLAN, a ThickLAN (HP's implementation of IEEE 802.3 Type 10BASE5) connection, or to a Pod for a StarLAN 10 (HP's implementation of IEEE 802.3 Type 10BASE-T) connection.

CAUTION

CORRUPTED DATA! The LAN connection to the BNC will maintain software integrity and can maintain communication when subjected to low levels of Electrostatic Discharge (ESD) directly to the LAN connector.

When operating in an environment where ESD pulses are in excess of 2500 volts, using a ThinMAU adapter (instead of a direct connection) is more reliable and less susceptible to data corruption from ESD to the LAN cable.

Use either the BNC or the AUI connector. Do NOT use both. The LAN interface will not work with both connected at the same time.

Step 3: Install host software

1 If you have not already done so, install the LAN software on your host computer.

2 If you are using a UNIX-based workstation, install the interface software now.

HP supplies the **ipconfig700** command as part of the HP B1471 64700 Operating Environment software. This command will greatly simplify the task of configuring the LAN connection.

Step 4: Configure the LAN parameters

- If you are using a UNIX-based workstation and you have installed the
  ipconfig700 command, see "To configure LAN parameters using 'ipconfig700'" in
  this chapter.

- If you are using an HP-UX workstation, and you have not installed the
  ipconfig700 command, see "To configure LAN parameters using BOOTP" in the
  64700 Card Cage Installation/Service Guide.

- If you are using the HP Real-Time C interface on a PC, see the instructions in your
  Real-Time C interface manual.

- Otherwise, see "To configure LAN parameters using the terminal interface" in this
  chapter.

To configure LAN parameters using "ipconfig700"

If you are using an HP 9000 Series 300/400/700 computer or Sun SPARCsystem computer and you have installed the HP B1471 64700 Operating Environment software, you can configure the LAN parameters with the **ipconfig700** command.

The **ipconfig700** command sets the Internet Address (IP address), Gateway Address, and Subnet Mask on the card cage LAN interface. An Internet Address (also known as an IP address) must be configured for the card cage before a network interface connection can be made.

The **ipconfig700** command cannot be used if your system has a bootp daemon running. If this is the case, use BOOTP to configure LAN parameters.

The following steps need to be taken when configuring the network parameters.

1 Connect the card cage to your network. This connection can be made by using either the 15 Pin AUI connector or the BNC connector.

2 Set the rear panel dip switches to indicate the type of connection that is to be made.
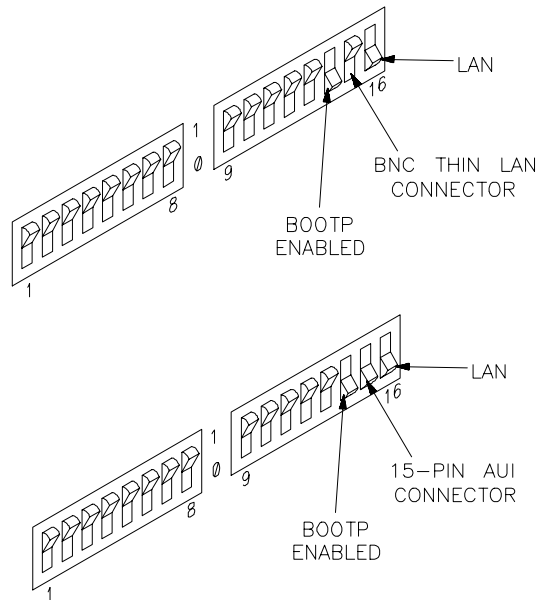
Switch 16 must be set to one (1) indicating that a LAN connection is being made.

Switch 15 should be zero (0) if you are connecting up to the BNC connector or set to one (1) if a 15 pin AUI connection is made.

Switch 14 must be set to one (1) to prepare for the receiving of the network parameters.

Set all other switches to zero (0).

3 Turn ON power to the emulator card cage.

4 Become the root user on the host computer.

5 Enter the **ipconfig700 -l <link> -i <internet> [-g <gateway>] [-s <subnet>]** command.

The **ipconfig700** parameters are:

-l <link> The Link Level Address is entered as 12 character hex ASCII address. This address is configured in each LAN interface shipped from the factory. This address is printed on the rear panel of the card cage. For example, 08000F090B30 is a link level address.

-i <internet> The Internet Address must be obtained from your Local System Administrator. The value is entered in integer dot notation. For example, 192.35.12.6 is an Internet Address.

-g <gateway> The Gateway Address is also an Internet address and is entered in integer dot notation. This entry is optional and will default to 0.0.0.0, meaning all connections are to be made on the local network or subnet. If connections are to be made to workstations on other networks or subnets, this address must be set to the address of the gateway machine. The gateway address must be obtained from your local system administrator.

-s <subnet> The Subnet Mask is also entered in integer dot notation. This entry is optional and will default to 0.0.0.0. The default is valid only on networks that are not subnetted. (A network is subnetted if the host portion of the Internet address is further

partitioned into a subnet portion and a host portion.) If the network is subnetted, a subnet mask is required in order for the emulator to work correctly. The subnet mask should be set to all "1"s in the bits which correspond to the network and subnet portions of the Internet address and all "0"s for the host portion. The subnet mask must be obtained from your local system administrator.

If the **ipconfig700** command is entered without any options, the program interactively prompts for the necessary information.

If the Link Level Address on the rear panel of your card cage read 08000F090F30, and your system administrator gave you the Internet Address 192.35.12.6, you could enter the following command:

```
$ ipconfig700 -l 08000F090B30 -i 192.35.12.6 <RETURN>
```

Since no Gateway Address or Subnet Mask was entered these values would default to 0.0.0.0. When the Internet Address is successfully programmed, **ipconfig700** will ask the emulator to display its version information.

6 Set switch 14 back to zero (0). Do this so the next time power is cycled on the emulator it will not enter a state waiting for network parameters.

7 Verify your emulator is now active and on the network by issuing a **telnet** to the Internet Address. For example:

```
$ telnet 192.35.12.6 <RETURN>
```

This connection will give you access to the emulator's built-in terminal interface. To exit from this telnet session, type <CTRL>d at the emulator prompt.

Once you have entered an Internet Address, and you want to change it to a different number, the best way to accomplish this is to telnet to the emulator and use the terminal interface **lan** command to make the change.

To configure LAN parameters using the terminal interface

1 Set all of the rear panel switches to the down position. This will set the serial port (Port A on a 64700A card cage) to 9600 baud and DCE.

2 Connect an ASCII terminal to the serial port with a 25-pin RS-232 cable.

You can also connect to a computer's RS-232 port and use a terminal emulation program on the computer. Refer to the "Connecting the HP 64700 Using RS-232/RS-422" chapter in the *64700 Card Cage Installation/Service Guide*.

3 Turn ON the emulator card cage. Press the terminal's <RETURN> key a couple times. You should see the "R>" prompt.

4 Display the current LAN configuration values by entering the **lan** command:

```
R>lan
lan -i 0.0.0.0
lan -g 0.0.0.0
lan -s 0.0.0.0
lan -p 6470
Ethernet Address : 08000903212f
```

Note the Ethernet Address, also known as the link-level address. This address is preassigned at the factory, and is printed on the rear panel.

5 Enter the **lan -i <internet> [-g <gateway>] [-s <subnet>] [-p <port>]** command.

The **lan** command parameters are:

-i <internet>   The Internet Address must be obtained from your local system administrator. The value is entered in integer dot notation. For example, 192.35.12.6 is an Internet Address.

-g <gateway>   The Gateway Address is also an Internet Address and is entered in integer dot notation. This entry is optional and will default to 0.0.0.0, meaning all connections are to be made on the local network or subnet. If connections are to be made to workstations on other networks or subnets, this address must be

set to the address of the gateway machine. The gateway address must be obtained from your local system administrator.

-s \<subnet\>   The Subnet Mask (in integer dot notation) is optional and only available when using the HP 64700A. It defaults to 0.0.0.0. The default is valid only on networks that are not subnetted. (A network is subnetted if the host portion of the Internet address is further partitioned into a subnet portion and a host portion.) If the network is subnetted, a subnet mask is required in order for the emulator to work correctly. The subnet mask should be set to all "1"s in the bits which correspond to the network and subnet portions of the Internet address and all "0"s for the host portion. The subnet mask must be obtained from your system administrator.

-p \<port\>   This changes the base TCP service port number. The host computer interfaces communicate with the emulator through two TCP service ports. The default base port number is 6470. The second port has the next higher number (default 6471). If you change the base port, the new value must also be entered in the /etc/services file on the host computer. For example, you could modify the line:

hp64700   6470/tcp

The default numbers (6470, 6471) can be changed if they conflict with some other product on your network. TCP service port numbers must be greater than 1024.

For example, to assign an Internet Address of 192.6.94.2 to the emulator, enter the following command:

R>**lan -i** 192.6.94.2 \<RETURN\>

The Internet Address and any other LAN parameters you change are stored in nonvolatile memory and will take effect the next time the emulator is powered off and back on again.

6 Turn power to the emulator OFF, and connect the the emulator to your network. This connection can be made by using either the 15 Pin AUI connector or the BNC connector.

7 Set the rear panel dip switches to indicate the type of connection that is to be made:
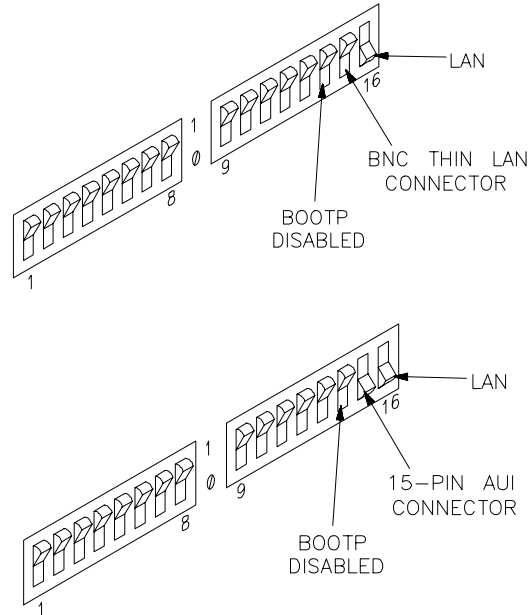
Switch 16 must be set to one (1) indicating that a LAN connection is being made.

Switch 15 should be zero (0) if you are connecting up to the BNC connector or set to one (1) if a 15 pin AUI connection is made.

Switch 14 should be zero (0).

Set all other switches to zero (0).

8 Turn ON power to the emulator card cage.

9 Verify your emulator is now active and on the network by issuing a **telnet** to the Internet Address. For example:

```
$ telnet 192.6.94.2 <RETURN>
```

This connection will give you access to the built-in terminal interface. To exit from this telnet session, type <CTRL>d at the emulator prompt.

The next time LAN parameters need to be configured, telnet to the emulator and use the terminal interface **lan** command.

If "telnet" does not access the emulator

You must use the **telnet** command on the host computer to access the emulator's built-in terminal interface. After powering up the emulator, it takes a minute before the it can be recognized on the network. After a minute, try the **telnet <internet address>** command.

If **telnet** does not make the connection:

☐ Make sure that you have connected the emulator to the proper power source and that the power light is lit.

☐ Make sure that the LAN cable is connected. Refer to your LAN documentation for testing connectivity.

☐ Make sure the rear panel communication configuration switches are set correctly. Switch settings are only used to set communication parameters when power is turned OFF and then ON.

☐ Make sure that the Internet Address is set up correctly. You must use the RS-232/RS-422 port to verify this that the Internet Address is set up correctly. While accessing the emulator via the RS-232/RS-422 port, run performance verification on the LAN interface hardware with the **lanpv** command.

If **telnet** makes the connection, but no terminal interface prompt (for example, R>, M>, U>, etc.) is supplied:

☐ It's possible that the emulator interface software is in the process of running a command (for example, if a repetitive command was initiated from telnet in another window). You can use <CTRL>c to interrupt the repetitive command and get the terminal interface prompt.

☐ It's also possible for there to be a problem with the emulator firmware while the LAN interface is still up and running. In this case, you must cycle power on the emulator card cage.
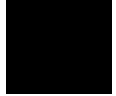
3

Connecting to the Demo Board

How to connect the emulator to the demonstration target system.

Installation

This chapter shows you how to connect the emulator to the demo target system which is shipped with the emulator. It also shows you how to verify installation by starting the emulator/analyzer interface for the first time.

Step 1. Connect the emulator probe to the demo target system

1 With HP 64700 power OFF, connect the emulator probe to the demo target system. When you install the probe into the demo board, be careful not to bend any of the pins. Do not insert the probe of the MC6833x emulator into the demo board socket incorrectly. Be very careful.

Note that pin 1 is located in the center of the line closest to the plastic cover on the MC68331, MC68332, and MC68334 emulator probes. Pin A1 is located in the corner (indicated by the dot on the PGA socket) on the edge furthest from the plastic cover on the MC68336 and MC68338 emulator probes. Correct orientation of the probe and demo board is shown below for all MC6833x emulator probes.

64782E06

2 Connect the power supply wires from the emulator to the demo target system. The 3-wire cable has one power wire and two ground wires. **When attaching the 3-wire cable to the demo target system, make sure the connector is aligned properly so that all three pins are connected**.

Power Connection
For Demo Board
From HP 64700A

Demo Board

64782E05
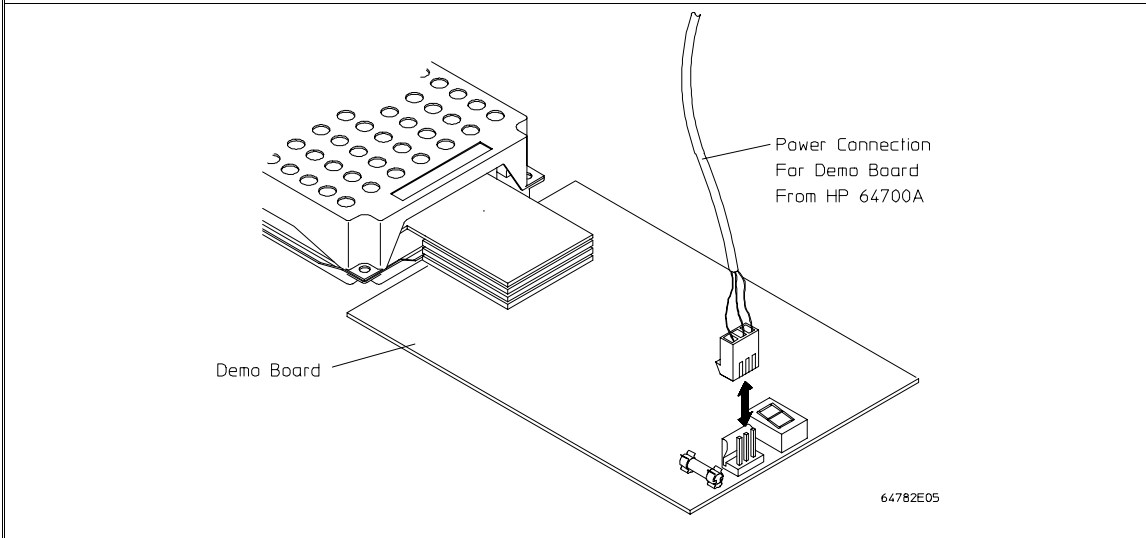
3 If you have not already done so, connect the +5 V power cable to the connector in the HP 64700 front panel. See page 19.

Power Connection
for Demo Board

MIS76021
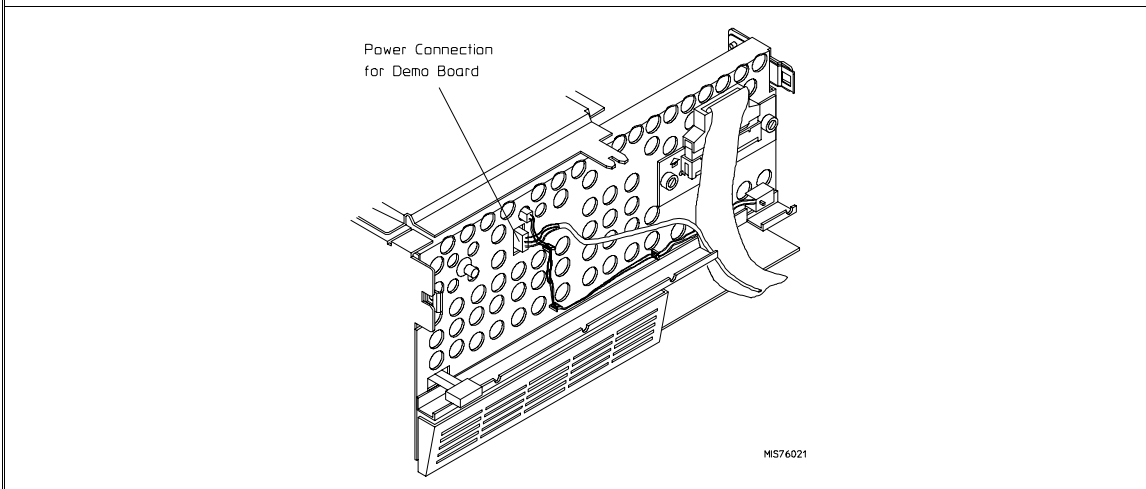
Step 2. Apply power to the HP 64700

1 Connect the power cord and turn on the HP 64700.

The line switch is a push button located at the lower, left-hand corner of the front panel. To turn ON power to the HP 64700, push the line switch button in to the ON (1) position. The power lamp at the lower, right-hand corner of the front panel will light.
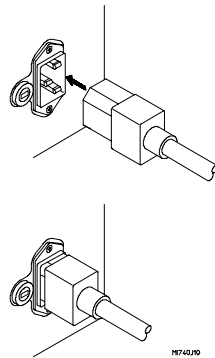
Step 3. Verify the performance of the emulator

1 Establish communication with the emulator from your host or ASCII terminal and
obtain a prompt (such as **R>**).

2 Enter: **pv 1** <return>

The emulator will print the results of its test, followed by a prompt.

3 Enter: **ver** <return>

The emulator will print some version numbers and the status of the memory
modules on the probe, followed by a prompt.

4 Enter: **<CTRL>d**

This command will end the emulation session.

5 If the emulator reported any failures, refer to Chapter 6.

Examples   If you are using a LAN, you can use the telnet capability with the built-in Terminal
Interface:

1 From your host computer enter the command: **telnet <emulator_name>**.

2 Now enter the command: **pv 1**

Note: the HP 64700 telnet capability is not supported by Hewlett-Packard.

Performance verification is discussed in detail in Chapter 8.

4
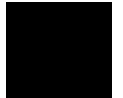
Connecting the Emulator to a Target System

Things you need to know to successfully connect the emulator to a target system and overcome problems you may encounter.

Plugging The Emulator Into A Target System

The following paragraphs help you understand the emulator. Equivalent circuits are shown, followed by a list of devices that you may need to use to overcome mechanical and electrical constraints in your target system.

Understanding an emulator

An emulator is a tool intended for debugging software, and the interactions between software and hardware. Although emulators can help in debugging certain hardware problems, catastrophic problems often require use of other tools, such as a timing analyzer with a preprocessor, or an oscilloscope. To use an emulator effectively, you need to understand its capabilities and limitations, and how it interacts with your target system. This chapter discusses limitations and interactions of an emulator, as they relate to your target system.

An emulator is designed to be electrically and functionally equivalent to the processor it emulates. Some MC6833x signals are electrically isolated from their counterparts on the target system connection. This is done for both electrical and functional reasons. Equivalent circuits of each processor signal are shown later in this chapter. The impact of these circuits is calculated and presented in the emulator specifications beginning on page 131.

Target system design

A target system that is designed around the worst-case specifications of an MC6833x processor should work with the emulator. Ideally, you would use the emulator specifications listed in this manual when designing your target system, instead of the processor specifications. But usually a target system has already been designed and prototyped before the emulator is introduced. If certain circuits in your target system do not allow for variations in the MC6833x specifications, compare the relevant emulator specifications to evaluate their impact on your target system. By keeping the differences between emulator specifications and processor

specifications in mind while you design your target system, you can save hours of debugging time when you plug the emulator into your target system.

Buffering and AC specifications

Most signals going to and from the emulator are not intercepted. Of those that are, for example the data bus, nearly all are gated using analog transmission gates, which introduce very little delay in these signals. Because these gates don't buffer the signal when they are on, the target system and the internal signals of the emulator are effectively connected. This typically results in the target system seeing more capacitance because of emulator circuitry and traces than with the processor alone. The net result is that even though these transmission gates don't add a lump sum delay, the capacitance may slow the edges of the signal, resulting in an extra delay.

The CLKOUT signal is buffered with an FCTA family part, followed by circuitry to allow operation at 3.3 volts. This improves drive capability to the target system and signal quality to the emulator.

DC specifications

Examine the DC specifications of the emulator to evaluate their differences from processor specifications. Again, you can refer to the equivalent circuit diagrams in this chapter for exact details. Because the emulator does not behave exactly like the processor, you may need to examine signal quality and take appropriate steps to compensate for differences.

Clocks

The MC6833x can operate in two basic clock modes: clocked from an oscillator, or clocked from a crystal. The emulator supports both these modes. To provide the most flexibility for supporting these clock modes, the emulator allows operation from either the target system clock source (external) or from an internal clock source.

The CLKOUT clock is the most important signal to the emulator because all system timing is derived from the CLKOUT signal. For proper operation, the CLKOUT clock signal must always be available to the emulator. It is possible to turn off the CLKOUT signal by setting the EXOFF bit in the SIM (or SCIM) Module Configuration Register (MCR). Make certain your target program never sets EXOFF.
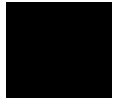
Target power

The emulator uses power from the target system to operate the emulation processor and some pullup resistors. Target power is sensed to make sure the emulator does not drive the target system until the target is powered up.

CAUTION

Possible target system damage. Because of the protections designed into the emulator, always turn power on in the emulator before turning power on in the target system. Always turn power off in the target system before turning off power in the emulator.

Equivalent circuits

The equivalent circuits shown on this page and the next will help you understand
connection requirements between the emulator probe and your target system.

V_{DD}, GND

6833X

64782B02

XTAL, XFC

64782B04

V_{DDSYN}

FERRITE
125Ω@100MHz

0.01μf   0.1μf   0.01μf   0.1μf

V_{DDSYN}

6833X

XFC

64782B03

CLKOUT

I_{OL} = 48mA
I_{OH} = −15mA

6833X

FCTA

5Ω

C=15pf

FCTA

I_I = ±15μA

FCTA

EMULATOR
CLKOUT

64782B05

cf mode = norm

MODCK

reset

6833X

5Ω

$I_I = \pm 10\mu A$

64782B06

$\overline{BKPT}$, FREEZE,
$\overline{IPIPE}$, $\overline{IFETCH}$

cf mode = norm

TARGET V_{DD}

6833X

5Ω

10KΩ

64782B07

V_{STBY}

6833X

0.1μf

64782B08

TSC

cf tsc
cf mode = norm

560Ω

$V_O = 0\text{-}10VDC$

64782B09

EXTAL

cf clk = int

C=8pf   C=5pf

6833X

64782B10

$\overline{HALT}$

TARGET V_{DD}

100KΩ

C=15pf

6833X

$I_I = \pm 5\mu A$

64782B11

$\overline{\text{DSACK0}}$, $\overline{\text{DSACK1}}$, $\overline{\text{BERR}}$,
D[15:0], D[7:0]

TARGET V_{DD}

MAPPING

100KΩ

6833X

C

5Ω

C=8pF

I_{IH}
I_{IL}

	C	I_{IL}	I_{IH}
$\overline{\text{DSACK0, 1}}$	20pf	−10μA	10μA
$\overline{\text{BERR}}$	15pf	−5μA	5μA
D[15:8]	30pf	−20μA	20μA
D[7:0]	25pf	−15μA	15μA
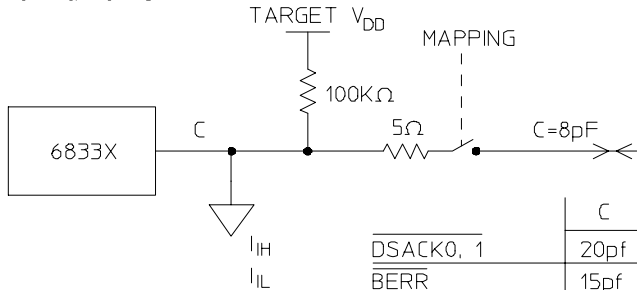
64782B14

$\overline{\text{CSBOOT}}$, $\overline{\text{CS10-0}}$,
A[18:0], R/$\overline{\text{W}}$,
$\overline{\text{AS}}$, $\overline{\text{DS}}$, SIZ0, SIZ1

	C	I_{IL}	I_{IH}
$\overline{\text{CSBOOT}}$, $\overline{\text{CS10-0}}$	40pf	−30μA	30μA
A[18:0]	30pf	−10μA	10μA
R/$\overline{\text{W}}$, SIZ0, SIZ1	15pf	−5μA	5μA
$\overline{\text{AS}}$	15pf	−250μA	25μA
$\overline{\text{DS}}$	15pf	−10μA	10μA

TARGET V_{DD}

100KΩ

6833X

C

I_{IL}
I_{IH}

64782B13

$\overline{\text{RESET}}$

TARGET V_{DD}

10KΩ

6833X

I_{IL} = −260μA
I_{IH} = 35μA

I_{OL} = 24mA
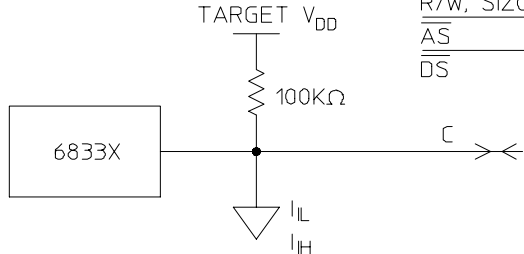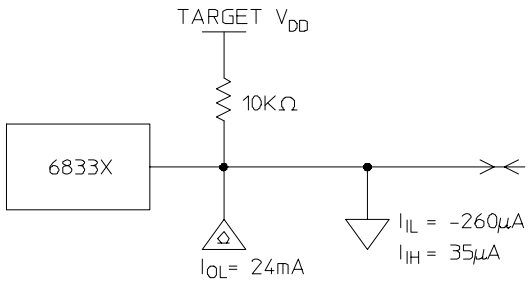
64782B12

$\overline{\text{AVEC}}$, $\overline{\text{RMC}}$, $\overline{\text{IRQ[7:1]}}$
and all other signals
not listed

6833X

C = 8pf

64782B15

55

Connecting the emulator to the target system

Plugging the emulator into a target system can be difficult because of mechanical constraints. If the mechanical constraints cannot be removed so that the emulator can be plugged directly into the target socket, there are several accessories available to help with the connection. These accessories are:

Stacking PGA pin protectors.

- 132-pin for 68331/2/4, HP Part Number 1200-1507
- 160-pin for 68336, HP Part Number 1200-1823

QFP Probe Adapter Assembly.

- 132-pin QFP to PGA flexible adapter for 68331/2/4,
  HP Part Number E3407A.

HP Elastomeric Probing System for 68332 and 68338

- 144-pin TQFP to PGA flexible adapter,
  order the following part numbers:
    - HP E5336A Elastomeric Probe Adapter
    - HP E5338A Flexible Adapter
    - HP E3416A Transition board for 68332
    - HP E5360A Transition board for 68338

HP Elastomeric Probing System for 68336, HP Part Number 64782-63217.

- 160-pin transition board
- 15x15 Header

The specifications given for the emulator do not include the loading of these accessories.

Use the following steps to connect the emulator to the target system:

1 Select the orientation (shown on the next pages) that best suits your target system.
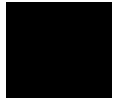
Flexible adapters can be installed in one of four orientations as shown in the following illustrations. This allows flexibility in attaching the emulator probe when target system components interfere.

CAUTION	**Equipment damage**. Serious damage to the target system or emulator can result from incorrect connection. The connections between the emulator probe, probe adapter, and MC6833x microprocessor are delicate and must be done with care.

2 Ensure proper alignment by taking note of the following on the illustrations:

- position of Pin 1 on the 6833x microprocessor
- position of little pin on the retainer (if applicable)
- position of little hole on the probe adapter (if applicable)
- color/bar code on both ends of the flexible adapter
- position of indicator on the transition socket
- position of Pin A1 on the emulator

3 If you are using a QFP Probe Adapter Assembly, refer to the operating note supplied with the probe adapter for specific instructions when making the connections.

4 If you are using a TQFP Elastomeric Probing System, you will need to adhere a retainer to the TQFP before installing the probe adapter. Refer to the installation guide supplied with the probing system when making the connections.

(a)

Pin 1 side

Emulator
Probe

Pin 1 side

Transition
Socket

Pin 1 side

QFP Flexible
Adapter

Pin 1

68331/2/4
132-pin QFP

(b)

Pin 1

Pin 1 side

Pin 1 side

(c)

Pin 1 side

Pin 1 side

Pin 1

(d)

Pin 1 side

Pin 1 side

Pin 1
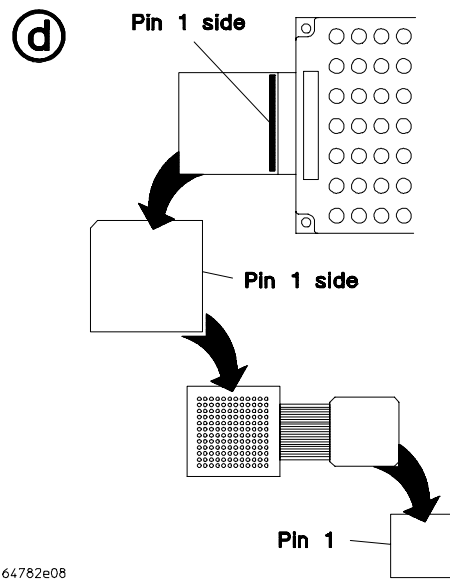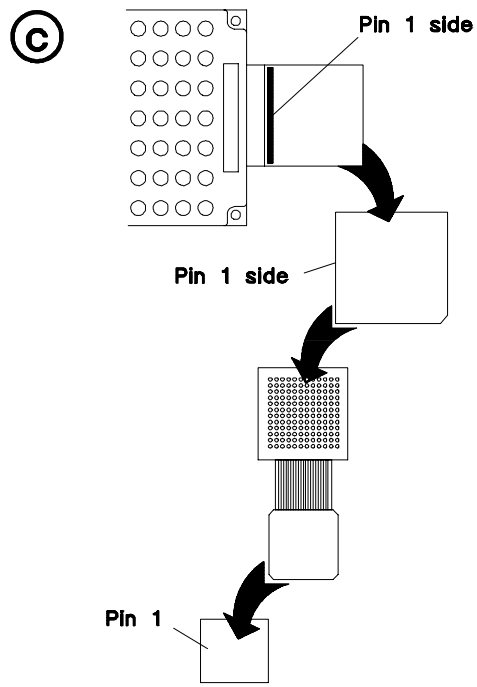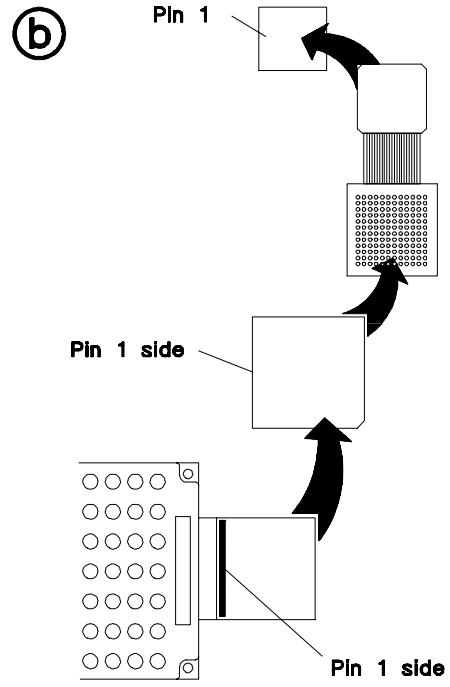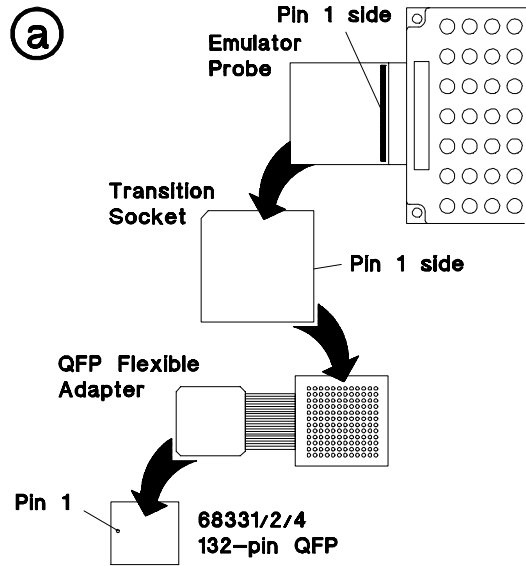
64782e08

Probing a 68331/2/4 microprocessor 132-pin QFP

58
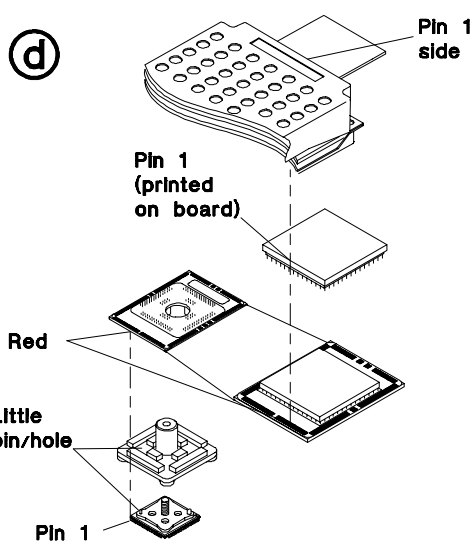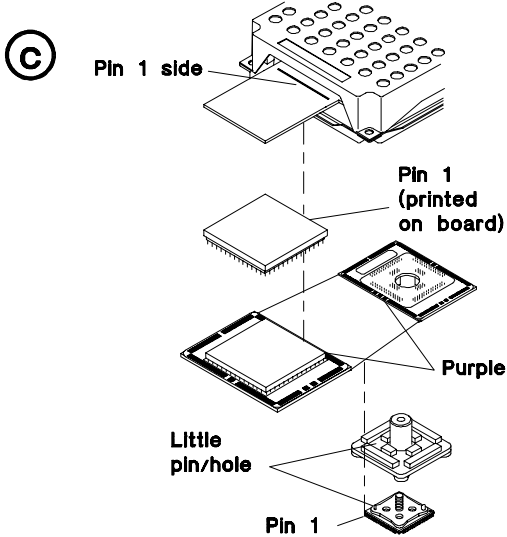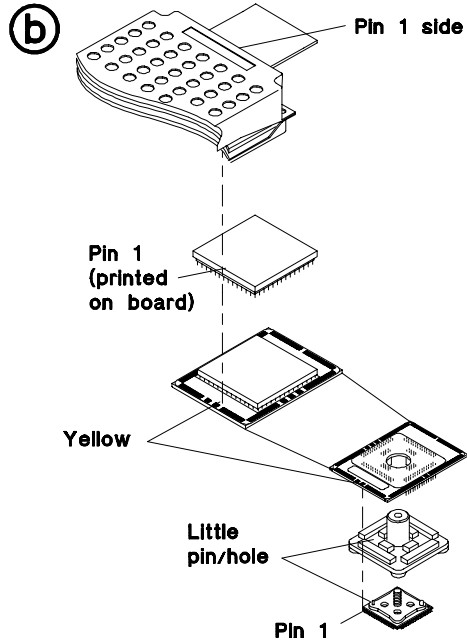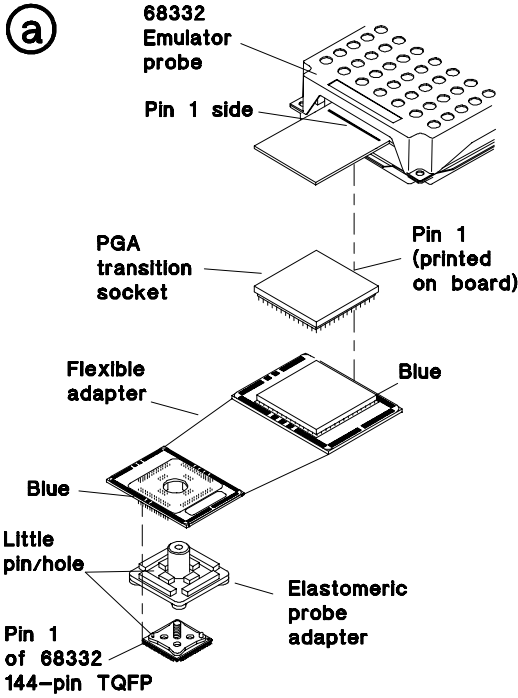
Probing a 68332 microprocessor 144-pin TQFP

(a) 68336 Emulator probe — Pin 1 side — PGA transition socket — Pin 1 side — Flexible adapter — Blue — Blue — Little pin/hole — Elastomeric probe adapter — Pin 1 of 68336 160-pin PQFP

(b) Pin 1 side — Yellow — Little pin/hole — Pin 1

(c) Pin 1 side — Purple — Little pin/hole — Pin 1

(d) Pin 1 side — Red — Little pin/hole — Pin 1
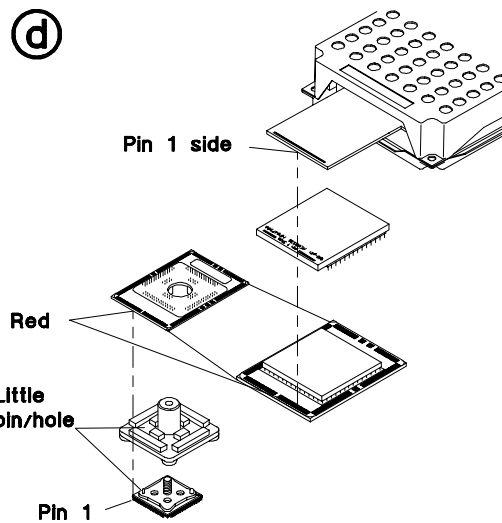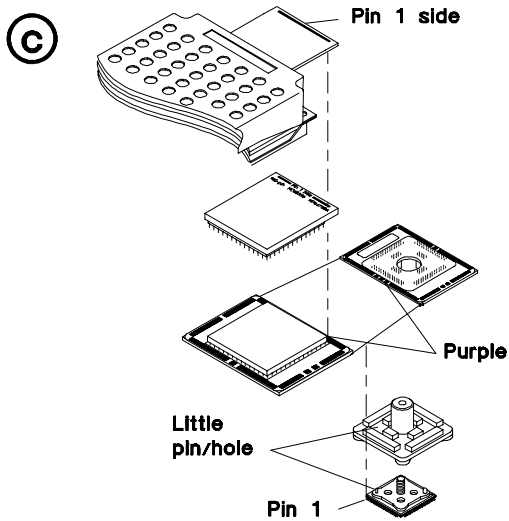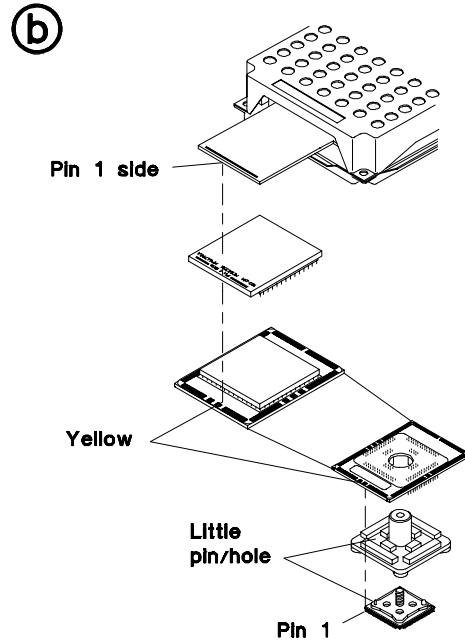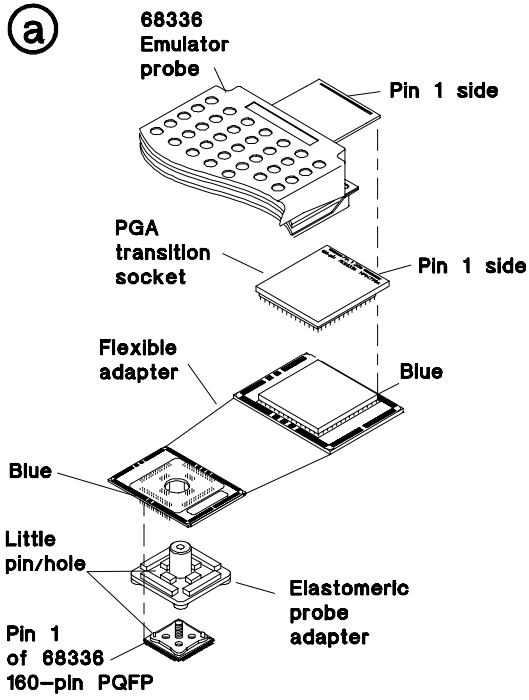
64798e13

Probing a 68336 microprocessor 160-pin PQFP

a

68338
Emulator
probe

Pin 1 side

PGA
transition
socket

Alignment
tab

Flexible
adapter

Blue

Blue

Little
pin/hole

Elastomeric
probe
adapter

Pin 1
of 68338
144-pin TQFP

b

Pin 1 side

Alignment
tab

Yellow

Little
pin/hole

Pin 1

c

Pin 1 side

Alignment
tab

Purple

Little
pin/hole

Pin 1

d

Pin 1 side

Alignment
tab

Red

Little
pin/hole

Pin 1
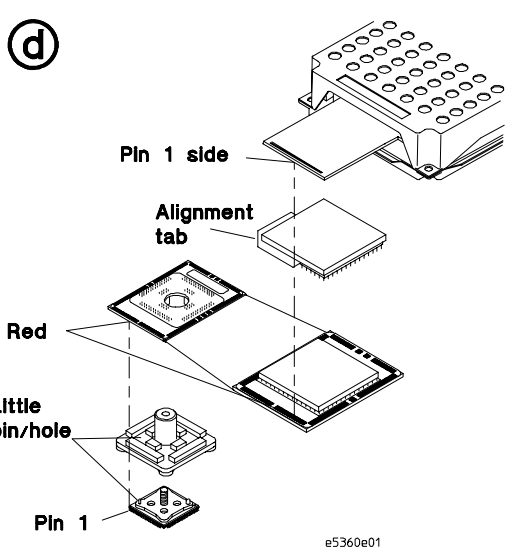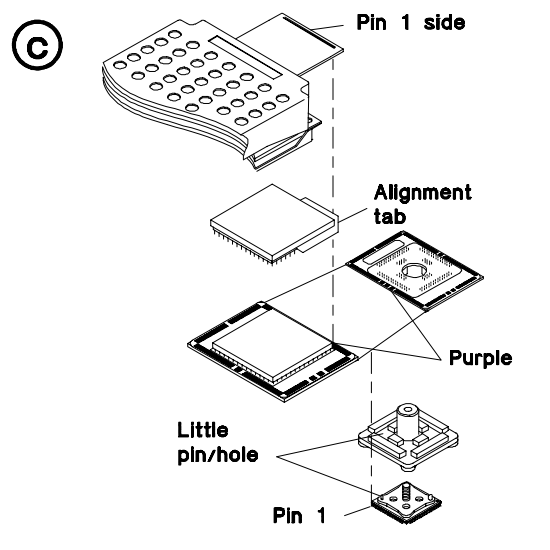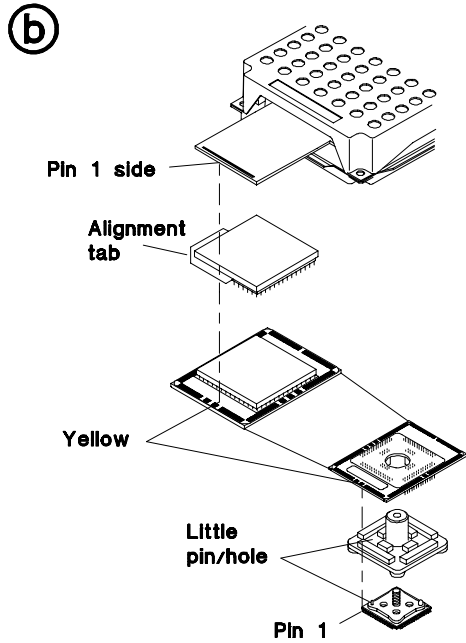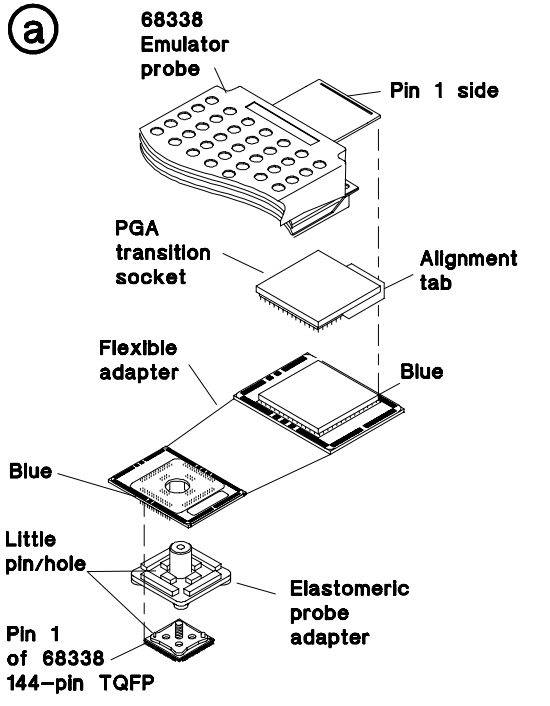
e5360e01

Probing a 68338 microprocessor 144-pin TQFP

Verifying Operation Of The Emulator In Your Target System

When connecting an emulator into a new target system, the step-by-step approach described in the remainder of this chapter will help you get your system running most quickly. This is a logical procedure that starts out with the most simple requirements and moves toward compete functionality, allowing for verification of installation at each step in the process. This not only helps debug problems if they arise, but builds confidence that the emulator is functioning correctly in your target system.

To begin, run the performance verification procedure described on page 103.

This procedure uses the built-in terminal interface. Once your target system is operating with the emulator, you should begin using one of the graphical user interfaces.

Additional equipment

Some additional equipment may be required to make measurements of MC6833x signals. It will help to have an oscilloscope and high speed timing analyzer to use during these procedures. A 250-MHz timing analyzer may be fast enough, but if you can obtain a faster timing analyzer, faster is better. The oscilloscope should have a single-shot bandwidth greater than 500 MHz. You may also need to cross trigger these instruments from the emulator. If there are no trigger inputs to the timing analyzer, you can probably use a timing channel. The BNC trigger output of the HP 64700 emulation card cage provides a rising edge TTL signal.

Probing

When making measurements, remember that signals need to be probed at the right place for the measurement being made. The emulator specifications are referenced to the target socket connector on the probe. This is where measurements should be made to verify compliance with the specifications. When probing setup and hold times to circuits in the target system, make the appropriate measurements at the circuits. This will keep connection accessories from impacting the true measurements. Always use ground leads to get the most accurate measurements possible. The wide trace that runs along the perimeter of the emulator probe (around the target connector) is a good ground that is easy to connect to when probing the target connector.

Selecting a clock source

The emulator is designed to support the two clock modes of MC6833x processors. If your target system provides a system frequency oscillator, select external clock in the emulator configuration. If your target system has a cyrstal and uses the clock synthesizer inside the MC6833x processor, select internal emulation clock.

When the emulator is configured for external clock, the target system clock is received on the EXTAL input to the MC6833x processor. The XTAL output from the MC6833x is not connected to the target system because it is specified to be a no connect in this mode. While /RESET is low, the MC6833x processor is disconnected from the target system MODCK line and MODCK is pulled low at the MC6833x processor. This instructs the MC6833x to use the oscillator signal as CLKOUT. The V_{DDSYN} power supply line from the target system is connected to the MC6833x processor through an LC filter on the emulator board.

When the emulator is configured for internal clock, the MC6833x is connected to a crystal oscillator circuit on the emulator board. The crystal frequency is 32.768 kHz for the MC68332 and others that use a crystal in the lower frequence range, and 4.194304 MHz for the MC68336 and others that use a crystal in the higher frequence range. While /RESET is low, the MC6833x processor is disconnected from the target system MODCK line and MODCK is pulled high at the MC6833x processor. This instructs the MC6833x to use its on-chip clock synthesizer to generate CLKOUT.

Running the emulator configured like the processor

To determine whether the loading and timing changes of the emulator impact your target system, configure the emulator to work like the target system processor. This procedure uses no emulation monitor and no emulation memory. For this test, the only emulation feature that is operating is the emulation-bus analyzer. The emulation-bus analyzer is passive, like a preprocessor.

This plug-in procedure uses the built-in terminal interface. Once your target system is operating with the emulator, you should begin using one of the graphical user interfaces.

If your target system can run a program without the emulator, follow this procedure. Otherwise, go to the next section.

1 Turn on power to the emulator.

2 Check the emulator prompt by pressing <RETURN>.

The prompt should be "p>". A prompt of "->" indicates a software compatibility problem. Correct problems indicated in error messages (seen in the emulator error log) or check the software version using the **ver** command for more information.

3 Configure the emulator by entering the following commands:

cf mode=norm
cf clk=int   (for crystal operation) OR **cf clk=ext**   (for oscillator operation)
cf mon=bg
cf rstcfg=ext

4 Set up the emulation-bus analyzer to capture all MC6833x system cycles.

tck -u
tg any
tsto any
tp s
t

5 Execute your program with the command: **r rst**.

This tells the emulator to deassert reset so that the emulator does not interfere with the target system powerup reset.

6 Power on the target system.

7 Verify correct operation.

The target system should run just as if the processor was being used. If your target system performs any I/O, check it to see of your system performs it correctly. If your target system appears to work correctly, allow it to reach its stable operating temperature and test it again.

If the target system appears to work correctly, go to the paragraph titled, "Installing the Monitor", later in this chapter. Otherwise, verify operation of the target system as described next.

To verify operation of the target system

Get the prompt by pressing <RETURN>, or use the command **es** to get more information about the emulator status. If the system is working the prompt will normally be "U>", but there are a few situations where the system will be working properly and the prompt will be something different. If the bus is taken away from the MC6833x often or for long periods of time, the emulator can display the "g>" prompt or alternate between "g>" and "U>". If the MC6833x is running code in its internal memory for long periods of time, the emulator may display the "b>" prompt. The emulator may alternate between any of these prompts during normal operation.

All other prompts usually indicate a problem. Even the "g>" or "b>" prompts can indicate a problem. To understand problems indicated by the prompts, you need to know whether bus cycles were executed, how many bus cycles were executed, what type of bus cycles were executed, and whether the target system is still executing bus cycles. You can tell the difference between these conditions by checking the trace status to see if any bus cycles were captured. The analyzer may have states in its internal pipeline that will not be reported until the trace is halted.

```
b>th;ts
 Emulation trace halted
 --- Emulation Trace Status ---
 User trace halted                    <- trace status
 Arm ignored
 Trigger not in memory
 Arm to trigger ?
 States 0 (0) ?..?                    <- number of states captured
 Sequence term 2
 Occurrence left 1
b>
```

If the trace status indicates that the trace was halted, look at the number of states collected to decide how many bus cycles were executed. If the status indicates that the user trace was completed, a large number of states were executed. If this is the case, it may help to take another trace to see if bus cycles are still being executed. Again, view the trace status to determine if bus cycles are executing.

If the "p>" prompt remains after target powerup, check:

- mechanical installation of the probe.
- blown fuses.
- target system power supply voltage.

If the prompt is "c>", mechanical installation may be causing the problem, but the most likely cause is a problem with the clock. If using a clock oscillator, check EXTAL clock quality. Look at the voltage levels, edges, and duty cycle. If the clock looks suspect, compare it to the target system clock without the emulator.

If the input clock is not the problem, check the CLKOUT signal. Is it operating correctly?

If the prompt is "r>", either the target system never released the reset pin, or some program error condition caused the target system to reset itself. If no bus cycles were captured by the analyzer, the target system never released reset. You need to find out which conditions must occur in order to release reset, and then investigate those conditions to determine why reset isn't being released.

Consider the example of a multicard system whose design uses a master card to start slave cards after verifying that they are installed in the system by reading checksums from their ROMs. The reset of each slave card is released when its checksum is read correctly. If the emulator interferes with the reading of the checksum from one of the slave cards, its reset will not be released.

One thing to keep in mind is that the emulator does not trace alternate bus master cycles while it is reset.

If any bus cycles were executed before the reset occurred, then something caused the target system to reassert the reset condition. Usually, this is caused by some type of fault detected by the system. Typical faults include the target system attempting to access a restricted address range, or a watchdog timeout. Refer to "Interpreting the Trace List", later in this chapter, to help understand what caused the reset.

The default programming of the 6833x chip will cause it to periodically reset the processor unless the watchdog address is appropriately accessed.

If the prompt is "b>", and there are no cycles in the trace list, the processor never attempted to run any bus cycles even if other indications show it should have. This could indicate problems with power, clock, or signal transitions, especially the reset signal. Items to check include: - Check power supply voltage levels. - Make sure the power up is monotonic. - Check clock quality. - Check that the reset

signal meets its required assertion time after power up and clock stabilization. - Check signal quality on the reset signal, especially the signal transitions.

If some cycles were captured in the trace list, but no cycles are occurring now, check for setup and hold violations on the processor signals. The "b>" prompt is not a normal condition for the processor when you find no functional reason. It usually indicates that the processor has malfunctioned.

If bus cycles are occurring, then the "b>" prompt only indicates that bus cycles are infrequent. A type of system that would exhibit this behavior would be an interrupt-driven system. When done processing an interrupt, the system could execute a STOP instruction to wait for the next interrupt. If the interrupts were infrequent a "b>" prompt would be displayed.

If the prompt is "w>", the emulator has stopped in the middle of a bus cycle.

To troubleshoot the above problem, you need to know if the target system provides bus termination for the address. If the answer is no, then the target program must have run incorrectly. The emulation-bus analyzer will have to be used to investigate further. If the answer is yes, then you will need to determine the reason the bus cycle did not complete, as described next.

There are many reasons why bus cycle interaction between a target system and an emulator may fail. A common cause is that no chip select has been programmed to assert DSACK internally for the current bus cycle. If the DSACK is generated external to the processor, another cause is that the target system missed the start-of-cycle indication from the emulator, or that the emulator missed the cycle-termination indication from the target system.

A basic MC6833x bus cycle starts with the transfer start signal, $\overline{AS}$, and possibly one or more chip selects. The $\overline{AS}$ signal stays low throughout the cycle and is deasserted between cycles. The end of the cycle occurs when the processor samples a transfer acknowledge DSACK1 or DSACK0 or both, or a transfer error acknowledge BERR on the falling edge of the clock. A typical system may sample $\overline{AS}$ on the rising clock edge and then generate a DSACK signal an integral number of clocks later. Wait states are added to a cycle by delaying when the $\overline{DSACK}$ is asserted.

If there is no functional reason why the bus cycle would not complete, check the timing relationships between the various bus cycle control signals. Probably the first measurement you will want to make is to see if the assertion of $\overline{DSACK}$ is within the emulator specification.
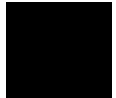
If there are no cycles in the trace list, then the processor stopped during the first bus cycle. In this case, set up the trace using $\overline{AS}$ as the trigger because the cycle of interest is the first cycle. If there are only a few cycles in the trace list, you can use the same technique if the oscilloscope or timing analyzer has enough depth.

If there are many cycles in the trace list before the processor stalled, use a different method of triggering. There are a number of different approaches that can be used. The most direct method is to trigger on a condition of $\overline{AS}$ low for a period of time greater than the length of a memory cycle. Another method is to determine if the system always stops at the same address. Use this address as the trigger. One drawback to this method is that you may have to probe a large number of signals to get a unique address.

A better way is to use the emulation-bus analyzer to generate a trigger. Unfortunately, because the cycle never finishes, the emulation-bus analyzer will not capture this address, so something preceding this event must be used as the trigger. Examine the trace list to find a unique event to use as the trigger. Once you have specified the trigger, you need to configure the emulator to drive the trigger out. The real trick to cross-triggering is to correlate the trigger event to the captured data. In this type of measurement, the correlation is easy because the signals of interest stop transitioning shortly after the trigger occurs.

tg addr=00badad
tp c
tgout trig2
bnct -r trig2
t

Once you have a trace of the offending cycle, verify that $\overline{DSACK}$ is present for the duration required to terminate a cycle. If $\overline{DSACK}$ is not asserted at all, it could be an indication that the target system missed the $\overline{AS}$. Set up your oscilloscope or logic analyzer to make a measurement on your cycle start circuitry to determine why the target system did not respond to the cycle.

If the prompt is "g>" and there are no cycles in the trace list, the target system never gave the bus to the processor. Check the bus arbitration signals for proper functionality and timing. Refer to the bus arbitration diagram below. Remember that the analyzer does not trace alternate bus master cycles while the emulator is reset, but it does once the emulator is running.

When trying to determine why the bus is not being granted to the processor, you will need to determine why either the bus arbitration circuitry or an alternate bus master is not behaving correctly. The processor is the bus master; therefore, it grants the bus with $\overline{BG}$ in response to a target system bus request $\overline{BR}$. The

processor does not grant the bus until it is idle so bus ownership changes as soon as the $\overline{BG}$ signal is asserted. The alternate master asserts $\overline{BGACK}$ to claim the bus before deasserting $\overline{BR}$.

If the bus is granted by the processor but is not being returned, check the bus arbitration signals $\overline{BR}$, $\overline{BG}$, and $\overline{BGACK}$. If the bus is never released, the alternate bus master may be stuck in the middle of a cycle. Check the cycle strobes $\overline{AS}$, $\overline{DSACK}$, and $\overline{BERR}$. These strobes do not have to be asserted during alternate master accesses, but if $\overline{AS}$ is shown to the processor, it will generate memory control signals and may even provide $\overline{DSACK}$ for the alternate master bus cycle.

If some cycles are shown in the trace list, but no cycles are occurring now, the processor executed some cycles before getting stuck in a DMA cycle. Examine the bus arbitration signals and cycle strobes around where the target system gets stuck. Use the same techniques to set up a trigger as were described for measuring a bus cycle that stops before it is complete.

If bus cycles are occurring, then the "g>" prompt indicates that a high percentage of the bus activity is by alternate bus masters.

Interpreting the trace list

In some cases, a problem caused by an errant bus cycle may not show up until many cycles later. The emulation-bus analyzer must be used to track back thru the sequence of events to the faulty bus cycle. Data problems often give this delayed appearance, but there may be other causes.

If the "h>" prompt is shown, indicating a double bus fault, and if there are only four states in the tracelist, this indicates a problem with fetching of the initial vectors.

```
h>tl
   Line    addr,H  6833x Mnemonic
   -----   ------  ----------------------------------------
      0    000000  $0000       data word rd (ds16)
      1    000002  $0000       data word rd (ds16)
      2    000004  $0BAD       data word rd (ds16)
      3    000006  $ADDR       data word rd (ds16)
      4
h>
```

The first four cycles in the trace list are the initial stack pointer and the initial program counter. The initial program counter must be even or the processor will

immediately double bus fault.   Verify that the data captured by the analyzer is what is expected.

If the data for the vectors is wrong, a trace should be set up to check for access problems during the fetch of the initial vectors. If the data is completely incorrect, suspect an address or strobe timing problem. If only a few bits are wrong or if the data in the trace is correct, suspect a data timing problem.

If the tracelist contains many cycles, start from the end and work backwards to understand what caused the double bus fault. If the trace was completed before the processor stopped, modify the trace specification to "trigger on nothing" so that the last bus cycles preceding the double bus fault will be captured.

Wait until the emulator status shows a double bus fault, and then halt the trace.

tg never

reset the target system

es
th
tl -23

```
h>tl
   Line    addr,H  6833x Mnemonic
   -----   ------  ----------------------------------
    -23    000100  Illegal Instruction: $4AFC        <- illegal inst
    -22    000102  NOP
    -21    0003fe  $0010    data word wr (ds16)       <- illegal inst stack
    -20    0003f8  $2700    data word wr (ds16)
    -19    0003fa  $0000    data word wr (ds16)
    -18    0003fc  $0100    data word wr (ds16)
    -17    000010  $0011    data word rd (ds16)
    -16    000012  $1111    data word rd (ds16)       <- odd vector
    -15    0003f6  $00D6    data word wr (ds16)
    -14    0003f4  $0000    data word wr (ds16)
    -13    0003f0  $0000    data word wr (ds16)
    -12    0003f2  $0100    data word wr (ds16)
    -11    0003ec  $0011    data word wr (ds16)       <- address error stack
    -10    0003ee  $1111    data word wr (ds16)
     -9    0003e8  $0011    data word wr (ds16)
     -8    0003ea  $1111    data word wr (ds16)
     -7    0003e6  $C00C    data word wr (ds16)
     -6    0003e2  $0000    data word wr (ds16)
     -5    0003e4  $0100    data word wr (ds16)
     -4    0003e0  $2700    data word wr (ds16)
     -3    00000c  $0033    data word rd (ds16)       <- odd vector
     -2    00000e  $3333    data word rd (ds16)
     -1
h>
```

A double bus fault occurs when the processor encounters an exception that prevents processing of a previous exception. An example of a double bus fault is shown above. This original exception occurred because the target system tried to execute an illegal instruction. During processing of the illegal instruction exception, the processor encountered another exception.

This exception was an address error caused because the vector supplied for the illegal instruction handler was odd. The double bus fault occurred when the vector supplied for the address error handler was also odd. Other things that can cause a double bus fault are bus errors that occur during exception stacking or vector fetch. Keep in mind that bus errors can happen because the target system asserts $\overline{BERR}$.

Once you have found the cause of the double bus fault, you need to determine the root cause of the problem. In some cases, the exception is a normal part of execution, but the subsequent faults indicate a problem. In some cases, the first fault indicates a problem directly, such as when the program has already malfunctioned, and the fault is caused by an unintentional access.

At this point, the problem is to find the faulty bus cycle that eventually caused a recognizable problem. The same situation exists if the processor stops execution at an address that should not have been executed, or if a program is simply running code incorrectly.

There are really only two ways to go about determining what is wrong. - One is to try to trace back the terminal error condition to a faulty bus cycle. - The other is to start at the beginning of the trace, or at some other known point, and work forward, comparing the trace to the execution that was expected while looking for the point where execution first becomes unexpected. A listing of the program or a tracelist captured by a preprocessor could be used for this comparison.

When you find a suspected bus cycle, set up a trigger on it so that you can make a timing measurement on the cycle. When looking for clues or shortcuts to the problem, keep in mind that a system is usually made up of many different types of memory devices: ROM, EEPROM, SRAM, DRAM, and peripheral ports. Each of these devices may have different timing characteristics. Also, keep in mind that unique characteristics of a bus cycle, such as size and number of wait states, may result in unique timing requirements.
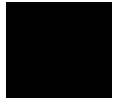
Fixing timing problems

When a timing problem is identified, you must decide how to fix it. First, examine the signal to make sure that signal quality is not affecting the timing. Look for AC or DC drive problems or reflections caused by transmission line problems. If you can find no other solution to the problem, you may have to decrease the clock speed.

If the timing problem only occurs during data accesses, another possible solution is to add wait states to the memory access. This assumes that the problem is caused by the amount of time it takes to access the memories in the system and is not a problem with a setup time to a synchronous circuit. A reliable indicator of this type of problem is when the data setup time to the emulator is being missed.

Another possible solution to data access problems is to use faster memories when using the emulator.

Installing the emulator in a target system without known good software

If you do not have a program in ROM on your target system that you can run to electrically test the emulator, you will need to create a test environment. The initial step of this is to use the emulator's memory to install a simple program that will run from reset. To do this, proceed as follows:

1 Turn on emulator power.

2 Check the prompt by pressing <RETURN>.

The prompt should be "p>". A "->" prompt indicates a software compatibility problem. Correct problems indicated in error messages or check the version (using the "ver" command) for more information.

3 Configure the emulator by entering the following commands:

cf mon=bg
cf mode=norm

cf clk=int
cf rstcfg=int,auto

4 Map memory with the following command:

map 0..0fff eram

This maps a block of emulation memory starting at address 0 so that the reset vectors will be accessed from this block.

5 Load a program with the following commands:

mo -ax -dl
m 0=0f00,100
mo -dw
m 100=60fe

This sets up reset vectors SSP=0f00 and PC=100. It then loads the most simple program imaginable: jump to self.

6 Setup a trace to capture all MC6833x cycles, as follows:

tck -u
tg any
tsto any
tp c
t

7 Execute **r rst**.

This tells the emulator to deassert reset so that the emulator does not interfere with the target system powerup reset.

8 Power on the target system.

9 Verify correct operation.

The prompt should be "U>". If you examine the trace, the program should be looping at address 100.

If the target system appears to work properly, go ahead to the paragraph titled "Installing a Monitor". If you suspect problems, return to "Verifying System Operation" in the previous paragraphs. Keep in mind that the emulator must receive strobes from the target system for emulation memory accesses to complete.

Also, because these cycles are from internal emulation memory, the data on the target system will not be the same as the data seen by the processor.

Installing Emulator Features

Once the emulator is transparently running in the target system, it is time to start adding other emulator features. Dividing the installation of features into two tasks is the easiest way to debug problems. The monitor is the facility that provides the majority of the emulator's features, but some features like the reset circuitry do not require the monitor. The first feature to be installed does not depend on the monitor.
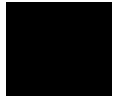
Evaluating the reset facilities

Now is a good time to use the emulator to find out how the emulator reset interacts with your target system. The first question to answer is whether or not the emulator reset command is adequate to reset your target system. Perform the following steps:

1 Run your target program by following the procedure in the previous steps.

2 Reset the emulation processor and run your program using the emulator commands:

r rst

Note that the **r rst** command pulses the processor reset line.

3 Verify correct operation.

If your program does not run correctly after performing the above procedure, your target system has other circuitry besides the processor that must be reset. The emulator only resets the emulation processor when it responds to a reset command. Other circuitry on your target system does not get reset. The following sequence determines if an additional reset circuit is required.

1 Run your target program following the procedure in the previous steps.

2 Reset the emulation processor and run your target program using these emulator commands:

rst

Reset the target system using whatever facility is available.

r rst

3 Verify correct operation of the target system.

An example of a target system that requires an additional reset circuit is one that normally has RAM starting at address 0, but for the first two bus cycles after reset, maps ROM to this area instead to provide the initial vectors. If this remapping does not occur, the system will attempt to fetch these vectors out of RAM, which will fail.

One additional thing to keep in mind is that your target system can initiate a reset without the knowledge of the emulator. A reset that is initiated by your target system will reset the emulator. If the emulator was running your target program at the time of the reset, then when your system releases reset, the emulator will run as if an r rst command had been issued. If the emulator was executing in the monitor at the time of the reset, it will return to the monitor when the reset is released.

Another resetting method that may be more convenient than the first method requires use of the monitor. This method works well for target systems such as those in the example above. This method resets the emulator into the monitor instead of running the target system program immediately. Once in the monitor, the initial stack pointer and initial PC can be loaded into the appropriate registers, and then a run of the target program can be initiated. This method will be illustrated in the next section.

Installing the background monitor

The emulator allows you to choose between use of a background and foreground monitor.

The background monitor is normally used unless interrupts need to be serviced while in the monitor. Other conditions associated with the 6833x communication facilities may also require the use of the foreground monitor.

The background monitor does not show cycles to your target system. It uses the built in BDM mode of the processor. Therefore, the background monitor is transparent to your target system. Even though the background monitor does not show any cycles to the target system, the initial vector fetch cycles are shown to the target system. Memory access cycles caused by the background monitor are called background cycles. All other cycles are called foreground cycles.
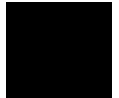
Resetting into the background monitor

There are three ways to initially get into the background monitor. The first of these ways is to enter the monitor from reset. Perform the following command sequence to enter the monitor:

1 Reset the emulator and the target system if necessary using any reset procedure you determined to work adequately.

2 Configure the emulator by entering the following command:

   cf mon=bg

3 Set up a trace to capture all MC6833x cycles, including background monitor cycles, by entering the following commands:

   tck -b
   tsto any
   tg any
   t

4 Execute the command: **rst -m**. This tells the emulator to release reset, but enter the monitor.

5 Verify that the emulator is in the monitor.

The prompt should be "M>", indicating that operation is in the monitor. There is not much that can go wrong up to this point because everything required has already been verified.

If you get the "?>" prompt or something other than the "M>" prompt, this indicates something went wrong with monitor operation. This may indicate problems with the clock or reset signals. Because the emulator provides all control signals for the background monitor, typically problems are with signals that can prevent the processor from running bus cycles.

Testing memory accesses with the background monitor

Once the background monitor looks like it is running properly, you can use it to test accesses to different ranges of memory in your target system. This may be an easier way to diagnose problems than by running a program that accesses each memory range. It is also easy to check accesses of different sizes using the monitor.

mo -aw -dl
m 0badad=12345678

When accesses to your target memory do not execute exactly right, the monitor attempts to diagnose these problems and resolve them so the monitor program does not malfunction. However, the monitor does not read back write cycles to check the integrity of the data written. When testing memory accesses, you must check the data to make sure it is correct.

```
M>m 0badad
  0000badad  ffdf00ff
```

If your target memory does not respond to a bus cycle, the monitor will force termination of the cycle and report this error message:

```
!ERROR  700! Target memory access failed
```

Running a program from the background monitor

Once you are satisfied that the monitor is working and memory in your target system can be accessed correctly, you can use the monitor to run your target program. Proceed as follows:

1 Reset into the monitor.

2 Load a program, if necessary.

3 Initialize the initial stack pointer and initial program counter.

This may not be necessary because the emulator will attempt to read the initial stack pointer and initial program counter from address 0 and 4 and initialize these registers. Alternatively, they can be set using the following commands.

reg ssp=<initial SSP>
reg pc=<target program starting address>

   If these values are not known, they can be found by taking a trace of the program running from reset, as was done in the previous sections.

1 Take a trace of the program running, using the following commands:

   tg addr=<target program starting address>
   t

2 Run the program with the command:

   r

3 Verify correct operation of the program.

From this point on, most of the problems will be discussed from a functional point of view instead of a parametric point of view. If any of the functional problems discussed below identify a problem that looks parametric, use the debugging techniques of the previous procedures to isolate the problem.
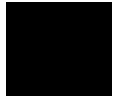
Breaking into the background monitor

The next thing to try with the background monitor is to see if you can break into it from your target program. The emulator uses the BKPT signal to break into the monitor. No stacking is required to enter the monitor. Program execution will stop abruptly once the BKPT signal takes affect. The background monitor may access foreground memory during its operation.

While the emulator is executing in the background monitor, no target interrupts are serviced. The interrupt signals from the target system are ignored. If the interrupt signals are still present when the monitor is exited, they will be serviced according to normal interrupt priorities.

Exiting the background monitor

If the procedures described in the preceding paragraphs gave satisfactory results, you should be able to resume execution of the target program.

r

If the target system and emulator do not work correctly after exiting the background monitor, the problem may be because your target system is real-time sensitive. Perhaps interrupts that needed to be serviced to keep the target system running were delayed or ignored during execution of the background monitor. Under these conditions, things such as data overrun could cause problems in the target system. If you suspect your target system is real-time sensitive, use the foreground monitor.

Software breakpoint entry into the background monitor

The background monitor can also be entered via a software breakpoint. The emulator will respond to any BGND instruction in the code, regardless of whether breakpoints are enabled, or whether or not the BGND instruction was inserted by the emulator. Breakpoints are enabled by the following command.

bc -e bp

Set breakpoints only on the initial word of an instruction (operation word); otherwise, they will be assumed to be extension words and not be executed. Instead, they might alter an earlier instruction, unintentionally. The emulator places a breakpoint by modifying memory to insert a breakpoint instruction at the address specified. If the memory at the address specified is ROM or cannot be modified for some other reason, the breakpoint cannot be set.

b
bp <instruction address>

When a BGND instruction is executed, the processor immediately transitions to the background monitor.

Stepping with the background monitor

The last feature of the background monitor which needs to be evaluated is the single-stepping facility. The emulator uses the BKPT signal to reenter the monitor when the first instruction outside the monitor is executed.

b
tsto any
tg any
t
s

```
0000100@sp  4e71       NOP
PC = 0000102@sp
```

81

A typical trace of a single step is shown below:

```
  Line   addr,H  6833x Mnemonic
 -----   ------  ----------------------------------------
     0   00010c  MOVE.L   D2,-A7
     1   00010e  NOP
     2   000110  MOVEQ    #$00000000,D2
     3   0001f4  $0000     data word wr (ds16)
     4
```

One way to watch what the emulator is doing during a step, is to set up the analyzer to trace only foreground cycles and to store everything. Use the following commands:

tck -u
tsto any
tg any
t
s

When stepping over instructions that cause the processor to take exceptions, the trace list can look very different. Most exceptions create their stack frame before entering the monitor.

Installing the foreground monitor

The foreground monitor supports interrupts and customization, but imposes on your target system more than the background monitor. The foreground monitor occupies a 4-Kbyte block in your target memory space. Emulation memory must be mapped for this 4-Kbyte block. The target system cannot use this address range for anything. The cycles strobes AS and DS are shown to the target system during foreground monitor cycles. The monitor needs to be placed in an address range where it will not interfere with target system operation.

If the monitor is placed in an address range where the target system responds with a DSACK, interlock the monitor to the target strobes. The target system must not respond with BERR for this address range. If the monitor is placed in an address range where the target system does not respond with any strobes, do not interlock the monitor. If in doubt, interlock the foreground monitor to the target system. It will be obvious if this is the wrong thing to do because the monitor will stop operating immediately.

If there is not a suitable address range in which to put the monitor, the system protection schemes may need to be modified to create a place for the monitor. This may be as simple as modifying the 6833x SIM programming, or it may require modifying a hardware protection scheme to allow placement of the monitor.

The foreground monitor, in contrast to the background monitor, allows servicing of interrupts. When the foreground monitor is not busy performing some action, interrupts are allowed. The interrupt routine must return control to the monitor within a reasonable period of time or the monitor may timeout if it attempts to do something. The level of interrupt that can be recognized by the monitor can be controlled through a configuration question:

cf monintr=0

The foreground monitor is entered and exited through the background monitor, therefore there are short periods of time when interrupts are not serviced.

Resetting into the foreground monitor

If you have successfully established operation of the background monitor, or if you have decided that you cannot use the background monitor because you require interrupt servicing, then it is time to evaluate the foreground monitor. The first thing to do is to enter the foreground monitor from reset. Perform the following command sequence to enter the monitor.

1 Reset the emulator, and the target system if necessary, using whatever reset procedure you determined to work.

2 Configure the emulator, as follows:

    cf mon=fg
    cf monaddr=addr as appropriate for the target system
    map addr..addr+0fff eram dsi
    cf monintr=0

3 Set up a trace to capture all MC6833x cycles. You must trace background cycles in order to see foreground monitor operation.

tg any
tsto any
tck -ub
t

4 Execute the command: **rst -m**

This tells the emulator to release reset, but enter the monitor.

5 Verify that the emulator is in the monitor.

The prompt should be "M>", indicating correct operation in the monitor. There is not much that can go wrong up to this point because everything required has already been verified.

If you get a "w>" prompt for a monitor address, you may have incorrectly interlocked the monitor to the target system. If the monitor was correctly interlocked, check to see if there is a timing problem with the target terminations for the monitor address range.

If you get the "b>" prompt or something other than the "M>" prompt, suspect a failure in monitor operation. These prompts may indicate problems with the clock or reset signals. If the monitor is interlocked, it may also indicate that the target system responded with a bus error for a monitor access.

Dealing with keep-alive circuitry by using the custom foreground monitor

The foreground monitor will periodically service the internal 6833x watchdog timer if it is enabled via the SIM registers, but you may have problems with other keep-alive circuitry located in the target system. Because the foreground monitor cycles are shown to the target system, bus cycle activity monitors should not be a problem. Also, because interrupts can be serviced within a reasonable period of time, any keep-alive circuits that depend on interrupts should not be a problem.

Keep-alive circuits that require a certain address to be accessed probably will fail when you are using the foreground monitor. The keep-alive problem will most likely show up immediately when using the foreground monitor. If the monitor is interlocked, it will be affected immediately if a keep-alive circuit causes a bus error. If a keep-alive circuit generates an interrupt or a reset, it should also be immediately obvious. If reset is only temporarily asserted, it may not be so obvious because the emulator will return to the monitor when it is released.

If you suspect a problem with a keep-alive circuit, try using the custom foreground monitor. This monitor can be customized to take the required actions to satisfy a keep-alive circuit. Look in the graphical interface manuals for information on using the custom foreground monitor. Retry your reset into the monitor with the customized foreground monitor.

If keep-alive circuits cannot be accommodated by using the available emulator features, you may need to disable them for emulation.

Testing memory access with the foreground monitor

Once the foreground monitor looks like it is running properly, you can use it to test accesses to different ranges of memory in your target system. This may be an easier way to diagnose problems than by running a program that accesses each memory range. It is also easy to check accesses of different sizes using the monitor.

mo -ax -dl
m 0badad=12345678

When accesses to your target memory are not performed exactly right, the monitor attempts to diagnose these problems and resolve them so the monitor program does not malfunction. However, the monitor does not read back write cycles to check the integrity of the data written. When testing memory accesses, check the data to make sure it is correct.

```
M>m 0badad
  00badad  ffdf00ff
```

If your target memory does not respond to a bus cycle, the monitor will force termination of the cycle and report this error message.

```
!ERROR  700! Target memory access failed
```

Running a program from the foreground monitor

Once you are satisfied that the monitor is working and that memory in your target system can be accessed correctly, you can use the monitor to run your target program. Use the following procedure:

1 Reset into the monitor.

2 Load a program, if necessary.

3 Initialize the initial stack pointer and initial program counter.

This may not be necessary because the emulator will attempt to read the initial stack pointer and initial program counter from address 0 and 4 and initialize these registers. Alternatively, they can be set using the following commands.

reg ssp=<initial SSP>
reg pc=<starting address of target program>

If you do not know these values, you can find them by taking a trace of the program running from reset as done in the previous sections.

1 Take a trace of the program as it is running, using the following commands:

tg addr=<starting address of target program>
t

2 Run the program with the command:

r

3 Verify correct operation of the program.

Breaking into the foreground monitor

The next thing to try with the foreground monitor is to see if you can break into it from your target program. The emulator uses the BKPT signal to break into the monitor. No stacking is required to enter the monitor. Program execution will stop abruptly once the BKPT signal takes affect. The background monitor automatically transfers control to the foreground monitor.

While the emulator is transitioning into the foreground monitor, interrupts are temporarily blocked. Once in the foreground monitor, the interrupt mask level is lowered to the greater of the "monintr" configuration setting or the target program mask level.

Entry into the foreground monitor can be traced by using the following trigger specification.

tck -ub
tp c
tg addr=<addr range of monitor>
t
b

Exiting the foreground monitor

If the tests of the preceding paragraphs operate correctly, you should be able to resume execution of the target program. You may want to take a trace of the monitor exit to verify that everything is working correctly. Use the run command:

r

Software breakpoint entry into the foreground monitor

Software breakpoints are handled the same as for the background monitor. Test them by using the same procedure as for the background monitor.

Stepping with the foreground monitor

Stepping for the foreground monitor is handled in the same way as the background monitor. Test stepping using the same procedure as for the background monitor.

Installing emulation memory
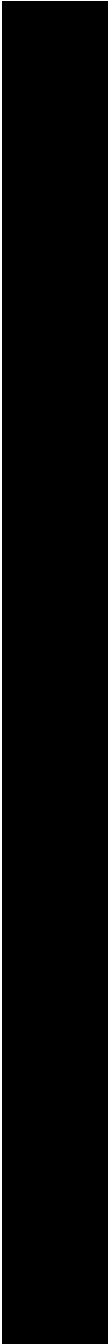
The last feature of the emulator that you need to integrate is the emulation memory. Emulation memory is intended to overlay ROM in the target system. This allows changes to target programs to be quickly loaded into a system. Emulation memory is completely dual ported.

If emulation memory is placed over existing target memory, interlock it to the target memory strobes. This ensures that the target memory control circuits remain in sync with the emulator. If there are no strobes that respond in the address range where emulation memory is placed, then do not interlock. When interlocked, both the DSACK and BERR signals are sampled.

Part 2

Service

5

Installing/Updating Emulator Firmware

How to update the emulator firmware using a workstation or PC.

When to update the firmware

You need to update the emulator firmware if:

- you ordered the HP 64782 and the HP 64748C separately,

- you are using an HP 64748C that has been previously used with a different emulator probe,

- you are upgrading to a newer version of the MC6833x processor, or

- you received a firmware update disk from HP.

If you ordered the HP 64782 MC6833x emulator probe and the HP 64748C emulation control card together, the control card contains the correct firmware for the HP 64782.

Updating the firmware using a workstation

If you have installed emulator interface software on your UNIX workstation, enter the following command at the operating system prompt:

```
progflash
```

The program will prompt you for further information. Further information on the **progflash** command can be found in the user interface manual.

Updating the firmware using a PC

The firmware, and the program that downloads it into the control card, are included with the MC6833x emulator probe on the following MS-DOS format floppy disks:

- 6833x EMULATION FIRMWARE

- 64700 SW UTIL

The steps to install or update the emulator firmware are:

1 Connect the HP 64700 card cage to the RS-232 port on your PC.

2 Install the firmware update utility and the emulator firmware.

3 Run "progflash" to update emulator firmware.

The above steps are described in detail on the following pages.
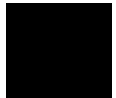
Step 1. Connect the HP 64700 to a PC host computer

1 Make a note of the present settings of the data communications configuration switches on the HP 64700 card cage. These switches are labeled "COMM CONFIG" (communications configuration). You may want to restore these switch settings after you finish updating the firmware.

Set the COMM CONFIG switches to the zero or open position, as shown in the illustration. These switches establish communication over the RS-232 at 9600 baud. Switch 16 is the LAN/RS-232 switch.
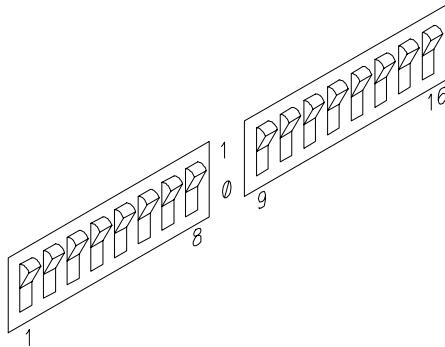
Note that switch settings are read during the power up routine of the HP 64700 card cage. Any changes made to the switches after power-up will not be read until you turn the HP 64700 off and back on again.

3 Connect the RS-232 cable.

Recommended cables are HP 13242N (25-pin male to 25-pin male) or HP 24542M (9-pin female to 25-pin male) which are equivalent to a MODEM cable.

To connect cable to the HP 64700, align the 25-pin male connector with port A and insert it until it is firmly seated. Tighten the holding screws on each side of the cable connector with a small, flat blade screwdriver. This will ensure that the cable pins and shield hood make good contact with the HP 64700 connector, and will also guard against accidental disconnection of the cable.

See which COM port the cable is connected to on your PC. Some PCs may label COM ports as Serial ports (eg: COM1 may be labeled Serial A).

Tighten Screws with
Small Flat Blade Screwdriver
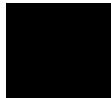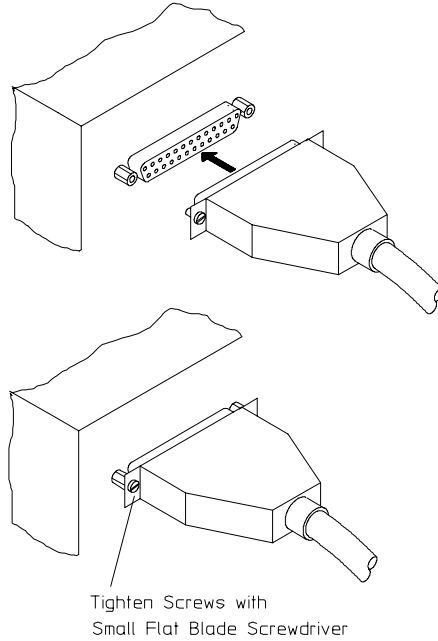
Step 2: Install the firmware update utility

Your HP Vectra PC or IBM PC AT compatible computer must have MS-DOS 3.1 or greater and a fixed disk drive. The firmware update utility and the 64782 firmware require about 360 Kbytes of disk space.

1 Insert the 64700 SW UTIL disk into drive A.

2 Change MS-DOS prompt to drive A: by typing "A:" at the MS-DOS prompt.

For example:

```
C> A: <RETURN>
A>
```

3 Type "INSTALL" at the MS-DOS prompt.

For example:

```
A> INSTALL <RETURN>
```

After confirming that you want to continue with the installation, the install program will give you the option of changing the default drive and/or subdirectory where the software will reside. The defaults are:

```
Drive = C:
Directory Path = C:\HP64700
```

Follow the remaining instructions to install the firmware update utility and the 64782 firmware. These instructions include editing your CONFIG.SYS and AUTOEXEC.BAT files. Details follow in the next steps.

4 After completing the install program, use the PC editor of your choice and edit the \CONFIG.SYS file to include these lines:

```
BREAK=ON
FILES=20
```

BREAK=ON allows the system to check for two break conditions: <CTRL><Break>, and <CTRL>c.

FILES=20 allows 20 files to be accessed concurrently. This number must be at LEAST 20 to allow the firmware update utility to operate properly.

5 Edit the AUTOEXEC.BAT file to add:

```
C:\HP64700\BIN (to the end of the PATH variable)
SET HPTABLES=C:\HP64700\TABLES (as a new line)
SET HPBIN=C:\HP64700\BIN (as a new line)
```

Part of an example AUTOEXEC.BAT file resembles:

```
ECHO OFF
SET HPTABLES=C:\HP64700\TABLES
PATH=C:\DOS;C:\HP64700\BIN
```

6 The default \HP64700\TABLES\64700TAB file contains entries to establish the communications connection for COM1 and COM2.

```
EMUL_COM1 unknown COM1 OFF 9600 NONE ON 1 8
EMUL_COM2 unknown COM2 OFF 9600 NONE ON 1 8
```

You may want to edit the \HP64700\TABLES\64700TAB file to change the name of the emulator beside the serial communications port you are using from "unknown" to "m6833x". You don't have to change this field in order to update the emulator firmware, but if you leave it as it is, your screen will say "unknown" instead of "m6833x" when you run PROGFLASH to update your firmware.

If you are using the COM3 or COM4 ports to update your emulator firmware, you will need to edit the \HP64700\TABLES\64700TAB file. Either add another line or modify one of the existing lines. For example:

```
EMUL_COM3 m6833x  COM3 OFF 9600 NONE ON 1 8
EMUL_COM4 unknown COM4 OFF 9600 NONE ON 1 8
```

Software is now installed on your PC and you are ready to update the emulator firmware. Reboot the PC to enable the changes made to the CONFIG.SYS and AUTOEXEC.BAT files. To reboot, press the <CTRL><ALT><DEL> keys simultaneously.

Step 3: Run "progflash" to update emulator firmware

- Enter the following command:

```
PROGFLASH
```

The PROGFLASH command downloads code from files on the PC host computer into Flash EPROM memory in the HP 64700. The full syntax is:

```
PROGFLASH [-V] [EMUL_NAME] [PRODUCT]
```

The -V option means "verbose". It causes progress status messages to be displayed during operation.

The EMUL_NAME option is the logical emulator name as specified in the \HP64700\TABLES\64700TAB file. Note that this name may be "unknown" if you didn't edit this file, but PROGFLASH will still work properly.

The PRODUCT option names the product whose firmware is to be updated (eg: 64782).

If you enter the PROGFLASH command without options, it becomes interactive. If you don't include the EMUL_NAME option, PROGFLASH displays the logical names listed in the \HP64700\TABLES\64700TAB file and asks you to choose one. If you don't include the PRODUCT option, PROGFLASH displays the products which have firmware update files on the system and asks you to choose one. You can abort the interactive PROGFLASH command by pressing <CTRL>c.

PROGFLASH will return 0 if it is successful; otherwise, it will return a nonzero (error) and a message will be written on the standard error output. If an error occurs, return to Step 1 of this procedure and change the setting of switch S4 (the DCE/DTE switch). Then cycle power on the card cage and try this "Step 3" procedure again.

You can verify the update by displaying the firmware version information.

Examples   To install or update the HP 64782 emulator firmware in the HP 64700 card cage
that is connected to the COM1 port:

```
C> PROGFLASH <RETURN>


                HP64700S006 A.00.04 24Feb92
                      64700 SW UTIL

         A Hewlett-Packard Software Product
         Copyright Hewlett-Packard Co. 1991


  All Rights Reserved. Reproduction, adaptation, or translation without prior
  written  permission  is prohibited, except as allowed under copyright laws.

                   RESTRICTED RIGHTS LEGEND

     Use , duplication , or disclosure  by the  Government is  subject to
     restrictions as set forth in subparagraph (c) (1) (II) of the Rights
     in Technical Data and Computer Software clause at  DFARS 52.227-7013.
     HEWLETT-PACKARD Company , 3000 Hanover St. , Palo Alto, CA 94304-1181

             Logical Name          Processor
           1 EMUL_COM1             m6833x
           2 EMUL_COM2             m68000

Number of Emulator to Update? (intr (usually cntl C or DEL) to abort)
```

To update firmware in the HP 64700 that is connected to the COM1 port, enter "1".

```
        Product
        1 64782

Number of Product to Update? (intr (usually cntl C or DEL) to abort)
```

To update the HP 64782 MC6833x emulator firmware, enter "1".

```
Enable progress messages? [y/n] (y)
```

To enable status messages, enter "y".

```
Reading configuration from '/hp64700/update\64782.CFG'
System firmware revision required = A.03.00
ROM identifier address = 2FFFF0H
Required hardware identifier = 1FFFH, 12FFH, 1201H, 1202H, 1203H, 1204H, 1205H, 1206H,
1207H
Control ROM start address = 280000H
Control ROM size = 40000H
Control ROM width = 16
Programming voltage control address = 2FFFFEH
Programming voltage control value = FFFFH
Programming voltage control mask = 0H

Rebooting HP64700...
Downloading flash programming code: '/hp64700/lib/npf.X'
Checking Hardware id code...
Erasing Flash ROM
Downloading ROM code: '/hp64700/update/64782.X'
    Code start 280000H
    Code size 30156H
Finishing up...

Rebooting HP64700...
Flash programming SUCCEEDED
C:\>
```
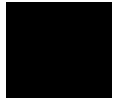
You could perform the same update as in the previous example with the following command:

```
C> PROGFLASH -V EMUL_COM1 64782 <RETURN>
```

6

Solving Problems

What to do when the emulator does not behave as expected..

Sometime during your use of the emulator, you will encounter a problem that is not adequately explained by an error message or obvious target system symptoms. This chapter explains how to solve some of these problems.

To verify the performance of the emulator

1 If you have a special configuration or session in progress, save it now. This procedure will cause your session to be lost.

2 Turn off power to the HP 64700 Card Cage.

3 Plug the emulator probe into the Demo Board. (See Chapter 3.)

4 Connect Demo Board power cable from the Demo Board to the HP 64700 Card Cage front panel. (See page 19.)

5 Turn on power to the HP 64700 Card Cage.

6 Establish communication with the emulator from your host or ASCII terminal and obtain a prompt (such as **R>**). (See Chapter 2.)

7 Enter: **pv 1** <return>

There are different hardware system configurations for the HP 64700 Series system. For information on hardware configurations, refer to the HP 64700 Installation/Service manual.
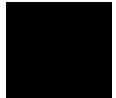
Examples   If you are using a LAN, you can use the telnet capability with the Terminal Interface:

1 From your host computer enter the command: **telnet <emulator_name>**.

2 Now enter the command: **pv 1**

Note: the HP 64700 telnet capability is not supported by Hewlett-Packard.

After about a minute, the emulator should display a list of tests which were performed, and whether they were passed or failed.

If you have an emulation failure, you can replace the assembly that failed through your local Hewlett-Packard representative, and through the Support Materials Organization (SMO). Refer to the list of replacable parts in chapter 7.

When your performance verification test is complete, use the keyboard **<CTRL>d** keys to end the emulation session.

To verify installation of memory modules in the deep analyzer card or in the emulation probe, type the ver command as follows:

M>**ver**

The emulator should display a message similar to the following:

```
            Copyright (c) Hewlett-Packard Co. 1987
All Rights Reserved.  Reproduction, adaptation, or translation without prior
written permission is prohibited, except as allowed under copyright laws.

  HP64700 Series Emulation System
    Version:   A.04.00 22Oct92

  HP64782C Motorola 68332 Emulator
    Version:   A.00.00 01Jun94
    Control:   HP64748C  Emulation Control Board
    Speed:     25.1 MHz
    Memory:    256 Kbytes
      Bank 0:  HP64171A (35ns) or HP64172A (20ns) 256 Kbyte Memory Module

  HP64740 Emulation Analyzer
    Version:   A.02.02 13Mar91

  HP64701A LAN Interface
    Version:   A.00.04 21Oct91
```

What is pv doing to the Emulator?

The performance verification procedures (PV) provide a thorough check of the functionality of all of the products installed in the HP 64700 Card Cage. The PV procedures change the emulator configuration, content of the processor internal registers, and the processor internal memory (including TPU internal memory). After PV, you will need to reload the emulation configuration and program. The following is an example Test Suite for an HP 64700 Card Cage containing an HP 64782 Emulator. To see the exact Test Suite for the HP 64782 Emulator and associated modules in your HP 64700 Card Cage, enter the command: **pv -l**.

```
 Tests available in Emulator Subsystem:
   test # 1 (ABG Type Map)
   test # 2 (ABG 68k RAM -Host Side)
   test # 3 (ABG 68k RAM - CPU Side)
   test # 4 (Host Side SIMM Data)
   test # 5 (Host Side SIMM Address)
   test # 6 (In-Circuit Verification)
   test # 7 (Emulator Reset)
   test # 8 (Background Monitor)
   test # 9 (Processor Address Lines)
   test #10 (Target Reset and Halt)
   test #11 (Target Data)
   test #12 (Target Address)
   test #13 (Chip Selects [5:0] and CSBOOT)
   test #14 (Sim Ports E and F)
   test #15 (Target Run and BERR)
   test #16 (Demo RAM Cell Test)
   test #17 (SIMM Cell Test)
   test #18 (LED Test)
   test #19 (Tri-State Control)
   test #20 (Guarded Break Test)
   test #21 (ROM Break Test)
   test #22 (Step Break Test)
   test #23 (Software Break Test)
   test #24 (Analysis Trace Test)
   test #25 (Analysis Break Test)
   test #26 (CMB Test)
   test #27 (QSM Port)
   test #28 (TPU Port)
   test #29 (QADC Port)
   test #30 (CTM4 Port)
```

Troubleshooting

The test results for all of these modules are indicated by a simple PASS/FAIL message. The PASS message gives a high level of confidence that all major functions and signals are operating because it includes a loopback test that includes read and write tests to the demo board. The demo board also stimulates inputs to the emulator.

A FAIL message on the other hand indicates that one or more of the tested functions is NOT working. In this event, an HP field representative can either swap

assemblies to isolate the failure to an individual board, or replace all the major assemblies shown in the replaceable parts list. The emulation memory modules and plastic cover are not part of the probe assembly. The emulation memory modules must be ordered separately and the plastic covers should be removed from the probe assembly before replacing the probe assembly.

To ensure software compatibility

There are various sets of firmware resident in the assemblies contained in the HP 64700 Card Cage. It is important to ensure that all the versions are compatible among the products you have installed. You can determine which versions of firmware you have by entering the terminal interface **ver** command.

There are at least four assemblies that have separate firmware in the HP 64700 Card Cage. These assemblies are:

- Host Controller card
- Emulator card
- Analyzer card
- Local Area Network card

If you purchased a complete Emulation/Analysis System from HP, you can be assured that all the products contained in the HP 64700 Card Cage contain compatible firmware at the time of sale. Software compatibility problems can occur when you swap the host controller card, emulator card, analyzer card, or local area network (LAN) card from one HP 64700 Card Cage to another, or from a recently purchased subassembly.

For example, you might purchase only the emulator subassembly (Emulation Control Card, Probe, and interconnecting ribbon cable) and replace the original emulator subassembly with the one you just purchased. In this case, the host controller may contain a version of firmware that is older than required to operate the new emulator; hence, compatibility problems can be caused by a newer emulator. All emulators will work with the latest software versions. The emulator software will warn you of incompatible software.

This emulator and the LAN card (if present) have Flash EPROMs that can be updated with current versions of firmware. Other products (assemblies) that do not use the Flash EPROM technology can also be updated with the latest firmware by adding a Flash EPROM card that is available from Support Materials Organization

(SMO). A Flash EPROM card can be inserted in an available slot in the card cage to override old versions of firmware in products with conventional EPROMs. The host controller in the card cage is already programmed to look for a Flash EPROM card if one is inserted into the card cage.

The latest versions of firmware for the host controller card, analyzer card, and LAN card, along with a program called *progflash*, are part of the B1471 software for the HP 9000 workstation and Sun SPARCsystems and the HP 64700 M6833x Real-Time C Debugger software for PCs.

When you load all your new versions of software onto your host computer, you are now ready to load the new version of firmware from your host computer to the assemblies that are in the HP 64700 Card Cage.

To load the new firmware, you use the *progflash* command. The progflash command must be run from a PC or workstation; it displays a list of card cages and subassemblies in each card cage on your system. From these lists, you can select which product to update. For information on using the *progflash* command, and updating your HP 64700 Series firmware, refer to the chapter 5, "Installing/Updating Emulator Firmware."

To display the emulator status

The emulator status is displayed on a status line in the graphical interface. If you need to display the emulator status using the built-in Terminal Interface:

- Display the emulator status by typing: **es**

The emulation prompts can usually tell you most information about the emulator's status: whether the emulator is reset, running a user program, or running in monitor. If you need more information than is given by the prompt, you can use the **es** command.

Example

```
R>es
  M68332--Emulation reset
R>
```

To check the version of the Terminal Interface software

☐ Type **ver** to display the version numbers of the Terminal Interface system software and emulator software.

The MC6833x emulator firmware must be used with the correct version of the emulation system and emulation analyzer firmware. See the paragraph titled, "To ensure software compatibility" earlier in this chapter for more information.

If the emulator appears to be malfunctioning

☐ Check to make sure that the cables connecting the Emulation Control Board to the Emulation Probe are connected correctly.

☐ Run the performance verification procedure as described in this chapter. If the emulator fails this test, contact your Hewlett-Packard representative.

☐ If the emulator passes the performance verification procedure, look for other reasons for the problem. Performance Verification is a thorough test, but it cannot find every hardware failure in the emulator. It is a good indication that the emulator is functioning correctly, but if you are still convinced the emulator is malfunctioning, contact your local Hewlett-Packard representative.

If you suspect the emulator is broken

☐ Run the Performance Verification procedure as described in the first paragraph of this chapter. If either the emulator or analyzer fail the performance verification, check the installation of those modules. If the installation is correct, contact your local HP Sales and Service office for assistance.

If you're having problems with DMA

☐ Check to make sure that your external DMA process doesn't access memory ranges mapped to emulation RAM (**eram**) or emulation ROM (**erom**). External DMA to emulation memory resources is not supported.

If you're having problems with TPU

☐ Motorola makes several versions of TPU microcode for certain 6833x microprocessors. HP uses one of the versions in its 6833x emulators. If you are using the default TPU functionality supplied by Motorola in the 6833x processor you are emulating, and if the 6833x processor version in the emulator is different from the 6833x processor version you are using in your target system, there may be a mismatch in TPU functionality between the emulator and target TPU.

To overcome this problem, you can develop your desired functionality and load it into on-chip RAM, and enable the TPU to use the code in RAM instead of the default microcode supplied by Motorola.

At the end of your development process, use the emulator in the ACT mode to test functionality of your target system. The ACT mode disables the emulation processor and enables the target processor. In this mode, you can test to see if you are obtaining the desired TPU functionality with your target TPU.

Note that when you reset the processor, the TPU reverts to use of the Motorola TPU microcode instead of the code you have placed in on-chip RAM. When you run performance verification within the emulator, any TPU code you have written and loaded into RAM is lost. The performance verification procedure writes special TPU code into on-chip RAM for use in testing the TPU.

Refer to the Motorola User's Manual for your MC6833x processor for information about how to load microcode into on-chip RAM and how to enable the TPU.

If the analyzer won't trigger

☐ Instruction fetches from internal 6833x resources normally aren't visible to the analyzer. You can force them to be visible by enabling *show cycles*. The SIM register MCR allows you to set show cycles. To obtain the show cycles feature of the processor, set the two least-significant bits in the third byte of register SIM_MCR to "11". For example:

```
R>reg emsim_mcr=63cf
R>rst -m
M>
```

If the analyzer triggers on a program address when it should not

☐ Check to see if the analyzer is triggering on an instruction prefetch. The analyzer cannot distinguish between prefetch and execution because the processor does not provide that information. Usually your actual trigger address is within four words of the address where trigger is occurring.

☐ Try to pad the program code with NOP instructions to move the trigger address away from the other code so that it won't be prefetched until it is time to trigger.

☐ You may be able to insert a write instruction to a meaningless variable in your code immediately following the trigger address. Then you can trigger on a write to the address of the variable. Write transactions never appear in instruction prefetches.

If trace disassembly appears to be partially incorrect

☐ Check to see if the analyzer began disassembly of the trace on a long-word boundary but the instruction started on the low word within the long word. This will make disassembly incorrect. You can start disassembly on the low word within the long word by use of **tl -d -ol <trace list line number>**.

☐ If the trace list seems correct for a few states after disassembly starts, and then it seems incorrect, restart disassembly of the trace at the low word where disassembly first becomes incorrect **tl -ol <trace list line number>**.

☐ If an instruction seems to have incorrect data associated with it, you can read down the trace list to see if you can find correct data for the instruction on another line. You can cause the disassembler to realign the instruction with the correct data by entering a command like **tl -d -ol <trace list line number containing instruction> <trace list line number containing data>**. For example, **tl -d -0l 38 47.**

If you see unexplained states in the trace list

☐ Check to see that the sequence, storage, and trigger specifications are set up to exclude the states you don't need.

☐ If you are using the built-in terminal interface, try using the **tl <instruction_state> <operand_state>** command to inform the dequeuer which operand state belongs with the named instruction state.

☐ If you are using the built-in terminal interface, try using the **-ol** option to the **tl** command to begin disassembly from the low word of the starting state, instead of the high word.

☐ Check to see if some of the program in the range covered by the trace might have executed from internal processor RAM memory. If so, these cycles won't appear in the trace list, which will confuse the disassembler. Use the *show cycles* feature of the processor to make internal RAM cycles visible to the analyzer. Obtain the

show cycles feature by setting the two least-significant bits in the third byte of register SIM_MCR to "11".

If you see exclamation marks "!" in count columns of the trace lists

☐ This is a normal condition. It indicates the counter overflowed (began again at 0) before the present state was captured. The exclamation mark warns you that the counter value may not be accurate because the analyzer is unable to determine how many times the counter overflowed between the preceding state and the state where the exclamation mark is shown.

If you were to scroll through a trace list of the entire trace memory in relative count mode, a "!" would be seen beside the first state after each occurrence of counter overflow (each 22.9 minutes). If you were to scroll through the entire trace memory in absolute count, the "!" would be seen beside every state after the first occurrence of counter overflow.

If you see negative time or state counts in trace lists

☐ If counter overflow occurs during a trace measurement, you may see a count of negative time or negative states in trace lists using the absolute time count mode. This indicates that the counter value stored with the trigger state was greater than the counter value stored with the present state. In absolute time counts, negative times will continue to be seen until a state is captured whose counter value is greater than the trigger state counter value. In relative time counts, the counter value is corrected so no negative time is seen.

If you do not see the counter overflow indication "!" where you expected to see it in a trace list

☐ This may be a normal indication. If you scroll through a reduced portion of the trace memory, one that contains no counter overflow, no counter overflow indication will be seen, even if counter overflow occurred before the line range you specified in your **display/store/copy** command. The routine that reads trace memory to compose a trace list only reads the portion of the trace memory you specify in your **display/store/copy** command.

If the Performance Verification reports massive pv failures

☐ Check to make sure mapper chip U84 on Emulation Control Card Subassembly HP 64748C is installed properly in its socket.

7

Parts List

Parts List

What is an Exchange Part?

Exchange parts are shown on the parts list. A defective part can be returned to HP for repair in exchange for a rebuilt part.

Probe (exchange)

To replace the Probe on the exchange program, you must remove certain parts, and return only that part considered an exchange part. When returning the Probe, you must remove the:

- cable assembly.
- top and bottom plastic covers.
- SRAM modules.
- demo board.

Emulation Control Card (exchange)

To replace the Emulation Control Card on the exchange program, you must remove certain parts, and return only that part considered an exchange part. When returning the Emulation Control Card, you must remove the:

- ribbon cable that connects the Emulation Control Card to the analyzer card.
- cable assembly.
- egress panel.

Main Assembly		
Component Part	**New**	**Exchange**
HP 64782 Probe and Demo Board		
6833x Firmware Floppy	64782-17006	
64700 SW UTIL	64700-17534	
MC68331 Probe Board Assy for HP 64782B	64782-62101	64782-69111
MC68332 Probe Board Assy for HP 64782C	64782-62102	64782-69112
MC68334 Probe Board Assy for HP 64782E	64782-62108	64782-69108
MC68336 Probe Board Assy for HP 64782G	64782-62106	64782-69116
MC68338 Probe Board Assy for HP 64782J	64782-62107	64782-69118
(Order the following parts separately:)		
Top Plastic Cover	64782-04103	
Bottom Plastic Cover	64761-04102	
External +5V Power Cable	5181-0201	
Plastic Rivets Kit (rivets and washers)	64748-68700	
PGA Pin Protector	5181-0206	
Demo Board for MC68331, MC68332, and MC68334	64782-66509	
Demo Board for MC68336	64782-66511	
Demo Board for MC68338	64782-66512	
(Order the following parts separately:)		
Fuse 3A 250V	2110-0837	
Fuse Holder, clip type, two required	2110-0825	
LED lamp	1990-0485	

Main Assembly		
Component Part	**New**	**Exchange**
HP 64748C Emulation Control Card Subassembly		
Egress Panel	64748-00205	
Bracket (used with Egress Panel)	64748-01201	
Spacer, Hex M3X6	0515-1146	
Screw, Machine M3X8	0515-0372	
Cable-100 36"	64748-61601	
Cable-100 37"	64748-61602	
Cable-100 38"	64748-61603	
Cable Clamp	64744-01201	
Rubber Strip	64744-81001	
Emulation Control Card	64748-66515	64748-69515
(without external cable or egress panel)		
Wrist strap	9300-1405	
HP 64171A 256-Kbyte, 35-nS SRAM Module	64171-66503	64171-69503
HP 64172A 256-Kbyte, 20-nS SRAM Module	64172A	64172-69501
HP 64171B 1-Mbyte, 35-nS SRAM Module	64171-66502	64171-69502
HP 64172B 1-Mbyte, 20-nS SRAM Module	64172B	64172-69502
HP 64173A 4-Mbyte, 25-nS SRAM Module	64173A	64173-69501
HP 64794A Emulation-Bus Analyzer (deep) card	64794-66502	64794-69502
34-pin ribbon cable	64708-61601	
Analyzer Card HP 64740 with 1K memory depth	64740-66526	64740-69526
34-pin ribbon cable	64708-61601	

Main Assembly		
Component Part	**New**	**Exchange**
Probing Accessories (not included with emulator, order separately)		
Stacking PGA Pin Protectors		
132-pin for 68331/2/4	1200-1507	
160-pin for 68336	1200-1823	
QFP Probe Adapter Assembly		
132-pin QFP to PGA flexible adapter for 68331/2/4	E3407A	
HP Elastomeric Probing System for 68332 and 68338		
144-pin TQFP Elastomeric Probe Adapter	E5336A	
Flexible Adapter	E5338A	
Transition board for 68332 (144-pin)	E3416A	
Transition board for 68338 (144-pin)	E5360A	
HP Elastomeric Probing System for 68336		
Transition board for 68336 (160-pin) includes a 15x15 header	64782-63217	
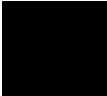
Part 3

Terminal Interface Reference
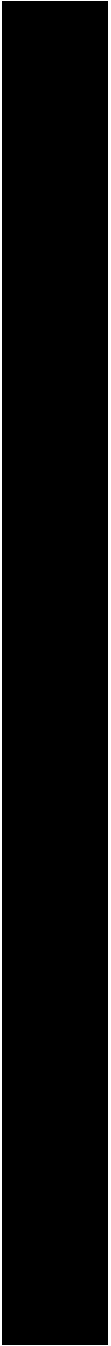
8

Using the Terminal Interface

An introduction to the emulator's built-in terminal interface.

The emulator has a built-in, host-independent Terminal Interface. The Terminal Interface provides all the commands you need to make emulation and analysis measurements. The interface includes tools for emulator initialization, command entry and recall, and command help.

When to Use the Terminal Interface

Hewlett-Packard suggests that you control the emulator with a graphical interface on a host computer, either a PC or a workstation (page 129). You may need to use the Terminal Interface, however, for tasks such as:

- Troubleshooting emulator hardware.

- Troubleshooting the connection between the emulator and your host computer.

- Developing a custom interface (such as a debugger) for the emulator.

- Using the emulator when a PC or workstation is not available, but a terminal or terminal emulator is available.

- Accessing emulator or analyzer features which are not supported by the graphical interfaces. In this case, it is best to issue the Terminal Interface commands from within the graphical interface.

- Using the emulator if you are already an expert user of the Terminal Interface and you do not need the improved ease of use of the graphical interfaces.

Learning About the Terminal Interface

You should be able to find most of what you need to know about the Terminal Interface commands from the on-line help.

To start the Terminal Interface

- To connect the emulator to a terminal or host computer, see Chapter 2.

- If the emulator is connected to a LAN, type the following command to log in to the emulator:

```
$ telnet hostname
```

Where <hostname> is the name of the emulator. You could use the Internet Protocol (IP) address (or internet address) in place of the emulator name, if desired.

Note   The "telnet" capability of the HP64700 card cage is unsupported. It is provided at no cost. Hewlett-Packard makes no warranty on its quality or fitness for a particular purpose.

- If the emulator is connected to your host computer or terminal via a serial port, press <return> a few times and you should see an emulator prompt.

The Terminal Interface is active when you connect a terminal or terminal emulator to the emulator via a serial or LAN connection.

Example   If the IP address of your emulator is 15.35.226.210, type:

```
$ telnet 15.35.226.210
```

You should see messages similar to:

```
Trying...
Connected to 15.35.226.210
Escape character is '^]'
```

After you connect to the emulator, you should see a prompt similar to:

```
R>
```

To view a list of available commands

1 Display the main help menu by typing: **help**

The main help menu lists groups of commands.

2 Display the commands in one of the groups by typing **help** followed by the name of the command group.

You can see a list of all of the commands by typing **help ***. This list may be too long to fit on your terminal display.

Example To display the main help menu, enter:

R>**help**

The emulator will list the command groups:

```
help  - display help information

  help <group>          - print help for desired group
  help -s <group>       - print short help for desired group
  help <command>        - print help for desired command
  help                  - print this help screen

--- VALID <group> NAMES ---
  gram     - system grammar
  proc     - processor specific grammar

  sys      - system commands
  emul     - emulation commands
  trc      - analyzer trace commands
  *        - all command groups
```

To display help information for the emulation command group, enter:

R>**help emul**

The emulator will list the emulation commands:

```
emul - emulation commands
  ------------------------------------------------------------------------------
  b......break to monitor  cp.....copy memory      mo.....modes
  bc.....break condition   dump...dump memory      r......run user code
  bp.....breakpoints       es.....emulation status reg....registers
  cf.....configuration     info...config info      rst....reset
  cim....copy target image io.....input/output     rx.....run at CMB execute
  ck.....check config      load...load memory      s......step
  cmb....CMB interaction   m......memory           ser....search memory
  cov....coverage          map....memory mapper    sync...sync sim and ems
```

To view help on individual commands

- To display help information for a particular command, type: **help <command_name>**

The emulator will display the command syntax, description, and examples.

Examples

To display help information for the processor step command, enter:

R>**help s**

The emulator will display:

```
s - step emulation processor

  s                     - step one from current PC
  s <count>             - step <count> from current PC
  s <count> $           - step <count> from current PC
  s <count> <addr>      - step <count> from <addr>
  s -q <count> <addr>   - step <count> from <addr>, quiet mode
  s -w <count> <addr>   - step <count> from <addr>, whisper mode

  --- NOTES ---
   STEPCOUNT MUST BE SPECIFIED IF ADDRESS IS SPECIFIED!
   If <addr> is not specified, default is to step from current PC.
   A <count> of 0 implies step forever.
```

To view help on command syntax

- Type: **help gram**

To get detailed information about the Terminal Interface

- Refer to a Terminal Interface manual for another HP emulator, such as the
  *MC68040/EC040/LC040 Emulator Analyzer (HP 64783A/B) User's Guide for the
  Terminal Interface*.

  If you use the Terminal Interface as your main interface to the emulator, you can
  find more information in any of HP's older Terminal Interface manuals. These
  manuals describe each command in detail, with syntax diagrams, parameter
  descriptions, and examples. These manuals also describe how to use the Terminal
  Interface to load, run, and debug programs.

Graphical Interfaces

HP provides a variety of interfaces for this emulator, including:

- The built-in Terminal Interface.

- A graphical interface for MS-Windows.

- A graphical emulator/analyzer interface for HP and Sun workstations.

- A graphical debugger for HP and Sun workstations.

HP also provides other instruments and software tools for developing, debugging, and optimizing embedded systems.

Note   Ask your HP sales representative for more information about these interfaces. Other interfaces may also be available from HP.

Why use a graphical interface?

- Use your host computer's windowing environment.

- Reduce many commands (such as setting breakpoints or stepping through source lines) to a single mouse-click.

- Show different kinds of information about your target system at the same time (for example, register contents, memory contents, and high-level source code) in separate windows on the display.

- Simplify emulator configuration by clearly displaying configuration options and their relationship to one another.

- Automate frequent tasks through menu selections, user-defined action keys, and command files.

- Work with compilers and assemblers to shorten the edit-compile-execute cycle.
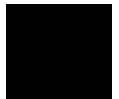
9

Specifications and Characteristics

Processor Compatibility

The HP 64782 emulator supports the Motorola 6833x family microprocessors operating at clock speeds up to 33 MHz.

The emulator supports both 5V and 3.3V operation. The emulator plugs directly into a PGA socket, and it can be plugged into a PQFP target system using optional accessories.

Electrical

Maximum clock speed

The maximum clock speed of the emulator is 33 MHz (with limitations shown in the table):

In	Zero Wait State	One Wait State
Target memory/Fast termination cycle	0 to 33 MHz	N/A
Emulation memory using: - 64171 (35 ns SIMM) - 64172 (20 ns SIMM) - 64173 (25 ns SIMM)	0 to 16.78 MHz 0 to 25 MHz 0 to 22 MHz	16.78 to 25 MHz 25 to 33 MHz 22 to 30 MHz

At the time this manual was printed:

- The MC68331, MC68332, and MC68334 emulators had been tested at 20 MHz.
- The MC68336 and MC68338 emulators had been tested at 16.78 MHz.

Contact your HP 64000 Field Engineer for current supported processor speeds.

HP 64782 Maximum Ratings

Characteristic	Symbol	Value	Unit
Supply Voltage	V_{CC}	–0.3 to +5.5	V
Input Voltage	V_{in}	–0.5 to +5.5	V
Maximum Operating Ambient Temperature	T_A	45	^{o}C
Minimum Operating Ambient Temperature	T_A	0	^{o}C
Storage Temperature Range	T_{stg}	–40 to +70	^{o}C
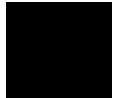
HP 64782 Electrical Specifications

HP 64782 — DC ELECTRICAL SPECIFICATIONS $(V_{DD}$ and $V_{DDSYN} = 5.0$ Vdc $\pm10\%$, $V_{SS} = 0$ Vdc$)$					
Characteristic	**Signal**	**Symbol**	**Min**	**Max**	**Unit**
Input High Voltage		V_{IH}	$0.7*V_{DD}$	$V_{DD}+0.3$	V
Input Low Voltage		V_{IL}	$V_{SS}-0.3$	$0.2*V_{DD}$	V
Undershoot			—	1.0	V
Hi-Z (Off-State) Leakage Current @ $V_{IN} = $ GND or V_{DD}		I_{OZ}	-20	20	µA
Signal Low Input Current $(V_{IL} = 0.8$ V$)$	$\overline{RESET}$ All other signals	I_{IL}	— —	0.8 100	mA µA
Signal High Input Current $(V_{IH} = V_{DD})$	All Signals	I_{IH}	—	50	µA
Output High Voltage $I_{OH} = -0.8$ mA, $V_{DD} = 4.5$ V	All Signals Except $\overline{HALT}$ and $\overline{RESET}$	V_{OH}	VDD-0.8	—	V
Output Low Voltage $I_{OL} = 1.6$ mA	$\overline{FREEZE}$, IPIPE, $\overline{IFETCH}$, D15-0	V_{OL}	—	0.4	V
$I_{OL} = 5.3$ mA	$\overline{CSBOOT}$, $\overline{CS}$10-0, $\overline{DSACK[1:0]}$,SIZ[1:0] $\overline{RMC}$, $\overline{AVEC}$, R/W, $\overline{AS}$, $\overline{DS}$		—	0.4	V
$I_{OL} = 15.3$ mA	$\overline{HALT}$, $\overline{RESET}$		—	0.4	V
$I_{OL} = 64$ mA	CLKOUT		—	0.4	V
Same as 6833x Specification	Any pins not listed		—	0.4	V

HP 64782 — DC ELECTRICAL SPECIFICATIONS $(V_{DD}$ and $V_{DDSYN} = 5.0$ Vdc $\pm10\%$, $V_{SS} = 0$ Vdc)					
Characteristic	**Signal**	**Symbol**	**Min**	**Max**	**Unit**
Input Capacitance	$\overline{CSBOOT}$, $\overline{CS10\text{-}0}$, D15-0	C_{in}	—	50	pF
	A18-0, $\overline{DSACK[1:0]}$, DSZ1-0, R/W, AS, DS		—	30	pF
	Any pins not listed		—	20	pF
Load Capacitance		C_L	—	100	pF
Power Supply Current Drawn from Target System			—	250	mA

HP 64782 AC Timing Specifications

All specifications are the same as listed in the Motorola *M6833x User's Manual*, except for the following.

Num	Characteristic	16.78 MHz		20.97 MHz		Unit
		Min	Max	Min	Max	
F1	Frequency of Operation (32.768 kHz or 4.194304 MHz crystal, as appropriate)	0.13	16.78	0.13	20.97	MHz
1	Clock Period	59.6	—	47.7	—	ns
1A	ECLK Period	476	—	381	—	ns
1B	External Clock Input Period	59.6	—	47.7	—	ns
2,3	Clock Pulse Width	21	—	16	—	ns
2A,3A	ECLK Pulse Width	236	—	183	—	ns
2B,3B	External Clock Input High/Low Time	29.8	—	23.8	—	ns
4,5	Clock Rise and Fall Time	—	5	—	5	ns
5C	CLKOUT Buffer Delay	2	5	2	5	ns
6	Clock High to Address, FC, SIZE, RMC Valid	-5	27	-5	21	ns
7	Clock High to Address, FC, SIZE, RMC High Impedance	-5	57	-5	45	ns

Table title:

HP 64782 — AC TIMING SPECIFICATIONS
(V_{DD} and V_{DDSYN} = 5.0 Vdc $\pm$10%, V_{SS} = 0 Vdc)

HP 64782 — AC TIMING SPECIFICATIONS (V_{DD} and V_{DDSYN} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc)						
		16.78 MHz		**20.97 MHz**		
Num	**Characteristic**	**Min**	**Max**	**Min**	**Max**	**Unit**
8	Clock High to Address, FC, SIZE, $\overline{RMC}$ Invalid	-5	—	-5	—	ns
9	Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Asserted	-3	23	-3	21	ns
9A	$\overline{AS}$ to $\overline{DS}$ or $\overline{CS}$ Asserted (Read)	-15	15	-10	10	ns
11	Address, FC, SIZE, $\overline{RMC}$ Valid to $\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Asserted	15	—	10	—	ns
12	Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated	-3	27	-3	21	ns
13	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to Address, FC, SIZE Invalid (Address Hold)	15	—	10	—	ns
14	$\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Width Asserted	100	—	80	—	ns
14A	$\overline{DS}$, $\overline{CS}$ Width Asserted (Write)	45	—	36	—	ns
14B	$\overline{AS}$, $\overline{CS}$ (and $\overline{DS}$ Read) Width Asserted (Fast Write Cycle)	40	—	32	—	ns
15	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Width Negated	40	—	32	—	ns
16	Clock High to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$ High Impedance	—	57	—	45	ns
17	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to R/$\overline{W}$ Negated	15	—	10	—	ns
18	Clock High to R/$\overline{W}$ High	-5	27	-5	21	ns
20	Clock High to R/$\overline{W}$ Low	-5	27	-5	21	ns

		16.78 MHz		20.97 MHz		
HP 64782 — AC TIMING SPECIFICATIONS (V_{DD} and V_{DDSYN} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc)						
Num	**Characteristic**	**Min**	**Max**	**Min**	**Max**	**Unit**
21	R/$\overline{W}$ Asserted to $\overline{AS}$, $\overline{CS}$ Asserted	15	—	10	—	ns
22	R/$\overline{W}$ Low to $\overline{DS}$, $\overline{CS}$ Asserted (Write)	70	—	54	—	ns
23	Clock High to Data Out Valid	—	27	—	21	ns
24	Data Out Valid to Negating Edge of $\overline{AS}$, $\overline{CS}$ (Fast Write Cycle)	15	—	10	—	ns
25	$\overline{DS}$, $\overline{CS}$ Negated to Data Out Invalid (Data Out Hold)	15	—	10	—	ns
26	Data Out Valid to $\overline{DS}$, $\overline{CS}$ Asserted (Write)	15	—	10	—	ns
27	Data In Valid to Clock Low (Data Setup)	10	—	10	—	ns
27A	Late $\overline{BERR}$, $\overline{HALT}$ Asserted to Clock Low (Setup Time)	25	—	20	—	ns
28	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{DSACK[1:0]}$, $\overline{BERR}$, $\overline{HALT}$, $\overline{AVEC}$ Negated	0	80	0	60	ns
29	$\overline{DS}$, $\overline{CS}$ Negated to Data In Invalid (Data In Hold)	0	—	0	—	ns
29A	$\overline{DS}$, $\overline{CS}$ Negated to Data In High Impedance	—	55	—	48	ns
30	CLKOUT Low to Data In Invalid (Fast Cycle Hold)	13	—	8	—	ns
30A	CLKOUT Low to Data In High Impedance	—	88	—	70	ns

138

		\multicolumn{2}{c|}{**16.78 MHz**}		\multicolumn{2}{c|}{**20.97 MHz**}		
HP 64782 — AC TIMING SPECIFICATIONS (V_{DD} and V_{DDSYN} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc)							
Num	**Characteristic**	**Min**	**Max**	**Min**	**Max**	**Unit**
31	$\overline{DSACK[1:0]}$ Asserted to Data In Valid	—	50	—	46	ns
33	Clock Low to $\overline{BG}$ Asserted/Negated	—	27	—	21	ns
35	$\overline{BR}$ Asserted to $\overline{BG}$ Asserted ($\overline{RMC}$ Not Asserted)	1	—	1	—	t_{cyc}
37	$\overline{BGACK}$ Asserted to $\overline{BG}$ Negated	1	2	1	2	t_{cyc}
39	$\overline{BG}$ Width Negated	2	—	2	—	t_{cyc}
39A	$\overline{BG}$ Width Asserted	1	—	1	—	t_{cyc}
46	$R/\overline{W}$ Width Asserted (Write or Read)	150	—	115	—	ns
46A	$R/\overline{W}$ Width Asserted (Fast Write or Read Cycle)	90	—	70	—	ns
47A	Asynchronous Input Setup Time $\overline{BR}$, $\overline{BGACK}$, $\overline{DSACK[1:0]}$, $\overline{BERR}$, $\overline{AVEC}$, $\overline{HALT}$	10	—	10	—	ns
47B	Asynchronous Input Hold Time	13	—	10	—	ns
48	$\overline{DSACK[1:0]}$ Asserted to $\overline{BERR}$, $\overline{HALT}$ Asserted	—	30	—	30	ns
53	Data Out Hold from Clock High	-5	—	-5	—	ns
54	Clock High to Data Out High Impedance	—	26	—	21	ns
55	$R/\overline{W}$ Asserted to Data Bus Impedance Change	40	—	32	—	ns

Num	Characteristic	16.78 MHz		20.97 MHz		Unit
		Min	Max	Min	Max	
56	$\overline{\text{RESET}}$ Pulse Width (Reset Instruction)	512	—	512	—	t_{cyc}
70	Clock Low to Data Bus Driven (Show)	-5	27	-5	21	ns
71	Data Setup Time to Clock Low (Show)	17	—	12	—	ns
73	$\overline{\text{BKPT}}$ Input Setup Time	n/a	n/a	n/a	n/a	ns
74	$\overline{\text{BKPT}}$ Input Hold Time	n/a	n/a	n/a	n/a	ns
75	Mode Select setup Time	20	—	20	—	t_{cyc}
76	Mode Select Hold Time	0	—	0	—	ns
77	$\overline{\text{RESET}}$ Assertion Time	4	—	4	—	t_{cyc}
78	$\overline{\text{RESET}}$ Rise Time	—	10	—	10	t_{cyc}

HP 64782 — AC TIMING SPECIFICATIONS
(V_{DD} and V_{DDSYN} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc)

Physical

Emulator Dimensions

173 mm height x 325 mm width x 389 mm depth (6.8 in. x 12.8 in. x 15.3 in.)

Emulator Weight

Probe alone: 0.3 kg (10 oz).

Cable Length

Emulation Control Card to Probe, approximately 914 mm (36 inches).

Probe dimensions
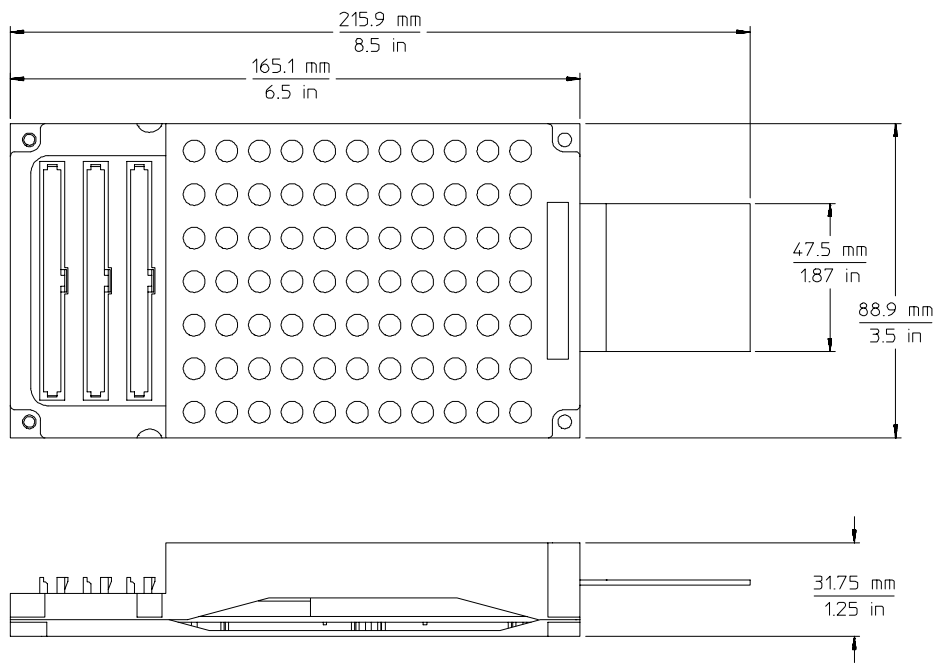
64782E01

Environmental

Temperature

Operating, 0° to +40° C (+32° to +104° F).
Nonoperating, -40° C to +60° C (-40° F to +140° F).

Altitude

Operating/nonoperating 4600 m (15 000 ft).

Relative Humidity

15% to 95%.

BNC, labeled TRIGGER IN/OUT

Output Drive

Logic high level with 50-ohm load >= 2.0 V.
Logic low level with 50-ohm load <= 0.4 V.

Input

74HCT132 with 135 ohms to ground in parallel.
Maximum input: 5 V above Vcc; 5 V below ground.

Communications

Host Port

25-pin female type "D" subminiature connector.

RS-232-C DCE or DTE to 38.4 kbaud.

RS-422 DCE only to 460.8 kbaud.

Auxiliary Port (64700A Only)

25-pin female type "D" subminiature connector.

RS-232-C DCE only to 19.2 kbaud.

CMB Port

9-pin female type "D" subminiature connector.

Glossary

Absolute Count

A count in the trace list count column that indicates the total count accumulated between the displayed state and the trigger state.

Absolute File

A file consisting of machine-readable instructions in which absolute addresses are used to store instructions, data, or both. These files are generated by the compiler/assembler/linker and are loaded into the emulator.

Access Breakpoint

A break from execution of your target program to execution of the emulation monitor when the emulator detects a read or write of an address or range of addresses.

Access Mode

Specifies the types of cycles used to access target system memory locations. For example a "byte" access mode tells the monitor program to use load/store byte instructions to access target memory.

Analyzer

An instrument that captures activity of signals synchronously with a clock signal. An emulation-bus analyzer captures emulator bus cycle information. An external analyzer captures activity on signals external to the emulator.

Analyzer Clock Speed

The bus cycle rate of the emulation processor. If the emulation processor is running at 21 MHz and the fastest bus cycle requires three clocks, then the analyzer clock speed (bus cycle rate) is $21/3 = 7$ MHz.

Arm Condition

A condition that reflects the state of a signal external to the analyzer. The arm condition can be used in branch or storage qualifiers. External signals can be from another analyzer or an instrument connected to the CMB or BNC.

Assembler

A program that translates symbolic instructions into object code.

Background

The emulator mode in which foreground operation is suspended so the emulation processor can be used for communication with the emulation controller. The background monitor does not occupy any processor address space.

Background Memory

Memory space reserved for the emulation processor when it is operating in the background mode. Background memory does not take up any of the microprocessor's address space.

Background Monitor

A monitor program that operates entirely in background memory. The background monitor can execute when target program execution is temporarily suspended. The background monitor does not occupy any of the address space that is available to your target program.

BNC Connector

A connector that provides a means for the emulator to drive/receive a trigger signal to/from an external device (such as a logic analyzer, oscilloscope, or HP 64000-UX system).

Breakpoint

A point at which emulator execution breaks from the target program and begins executing in the monitor. (See also Execution Breakpoint and Access Breakpoint.)

Command File

A file containing a sequence of commands to be executed.

Compatible Mode

The compatible mode of the deep analyzer configures the analyzer to provide the same memory depth as the 1K analyzer: 1024 states deep when the analyzer is not configured to make a count of states or time during a measurement, and 512 states deep when the analyzer is configured to make a count of states or time during a measurement. If the emulator interface you are using along with the deep analyzer requires that you use the compatible mode, the deep analyzer will still be able to provide one of its benefits for your measurement; you will be able to make your counts of states or time at full emulator clock speed.

Compiler

A program that translates high-level language source code into object code, or produces an assembly language program with subsequent translation into object code by an assembler. Compilers typically generate a program listing which may list errors displayed during the translation process.

Counter Overflow

When the counter reaches maximum count and begins a new count from zero. The counter of the deep analyzer simply counts continuously once a trace begins; it increments its count every 20 ns, and reaches maximum count in about 22.9 minutes (22 minutes and 54 seconds). The deep analyzer sets a flag in memory and stores it along with the first state that is captured after the counter overflow occurs (first state captured after the counter begins again at zero).

Configuration File

A file in which configuration information is stored. Typically, configuration files can be modified and re-loaded to configure instruments (such as an emulator) or programs (such as the PC Interface).

Coordinated Measurement

A synchronized measurement made between the emulator and analyzer, between emulation-bus analyzer and external analyzer, or between multiple emulators or analyzers. For example, a coordinated measurement is made when two or more HP 64700 emulators/analyzers start executing together, or break into background monitors at the same time.

Coordinated Measurement Bus (CMB)

The bus that is used for communication between multiple HP 64700 Series emulators/analyzers or between HP 64700 emulators/analyzers and an HP 64306 IMB/CMB Interface to allow coordinated measurements.

Cross Trigger

The situation in which the trigger condition of one analyzer is used to trigger another analyzer. Two signals internal to the HP 64700 can be connected through the BNC on the instrumentation card cage to allow cross-triggering between the emulation-bus analyzer and other analyzers.

DCE (Data Communications Equipment)

A specific RS-232C hardware interface configuration. Typically, DCE is a modem.

Deep Analyzer

In this manual, the term "deep analyzer" refers to the HP 64794 Emulation-Bus Analyzer with deep trace memory.

Display Mode

When displaying memory, this mode tells the emulator the size of the memory locations to display. When modifying memory, the display mode tells the emulator the size of the values to be written to memory.

Downloading

The process of transferring absolute files from a host computer into the emulator.

DTE (Data Terminal Equipment)

A specific RS-232C hardware interface configuration. Typically, DTE is a terminal or printer.

Embedded Microprocessor System

The microprocessor system which the emulator plugs into.

Emulation Bus Analyzer

A system component built into the HP 64700 that captures the emulation processor's address, data, and status information.

Emulation Memory

High-speed memory (RAM) in the emulator that can be used in place of target system memory.

Emulator

An instrument that performs just like the microprocessor it replaces, but at the same time, it gives you information about the operation of the processor. An emulator gives you control over target system execution and allows you to view or modify the contents of processor registers, target system memory, and I/O resources.

Emulator Probe

The assembly that connects the emulator to the target system microprocessor socket.

Escape Sequence (transparent mode)

A keyboard input consisting of a special sequence of characters, beginning with the escape character (1C hexadecimal). This sequence is used to access an emulator while in transparent mode. When using multiple emulators and transparent mode to access the different emulators, each one must be given a unique escape character.

Execution Breakpoint

A BKPT instruction placed in your software in RAM, replacing the normal instruction at the RAM address. Breakpoints for code in ROM
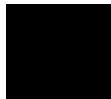
are stored in emulation hardware and jammed on the emulation bus during the fetch cycle. When the BKPT is executed, emulation immediately transfers from execution of your target program to execution of the emulation monitor.

Expression

The information that can fit into a single pattern or a single range (a pattern such as **addr=2105**, **data!=15**, or a range such as **addr=4012..401a**). A complex expression is made up of pattern, range, and arm labels joined together by various operators that define the specific condition. Each of the pattern and range labels must be previously assigned to specific simple expressions using the terminal interface commands: **tpat** and **trng**.

Foreground

The mode in which the emulator is executing the user program. In other words, the mode in which the emulator operates as the target microprocessor would.

Foreground Monitor

A monitor program that executes in the foreground address space. When the monitor exists in foreground, it is directly accessible by, and can interact with, your target program.

Guarded Memory

An address range that is to be inaccessible to the emulation processor. The emulator will generate a break and display an error message if an access to guarded memory occurs.

Handshaking

A process that involves receiving and/or sending control characters which indicate a device is ready to receive data, that data has been sent, and that data has been accepted.

Host Computer

A computer to which an HP 64700 Series emulator can be connected. A host computer may run interface programs which control the

emulator. Host computers may also be used to develop programs to be downloaded into the emulator.

Inverse Assembler

A program that translates absolute code into assembly language mnemonics.

Label

A set of one or more analyzer channels. Example, the label "addr" is used to identify the analyzer channels connected to the address bus of the emulation processor.

Linker

A program that combines relocatable object modules into an absolute file which can be loaded into the emulator and executed.

Logical Address Space

The addresses assigned to code during the process of compiling, assembling and linking to generate absolute files.

Macros

Custom made commands that represent a sequence of other commands. Entire sequences of commands defined in macros will be automatically executed when you enter the macro name. Macro nesting is permitted; this allows a macro definition to contain other macros.

Memory Mapper Term

A number assigned to a specific address range in the memory map. Term numbers are consecutive.

Memory Mapping

Defining ranges of the processor address space as emulation RAM or ROM, target RAM or ROM, or guarded memory.
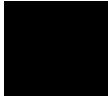
Monitor Program

A program executed by the emulation processor that allows the emulation system controller to access target system resources. For example, when you display target system memory locations, the monitor program executes microprocessor instructions that read the target memory locations and send their contents to the emulation controller.

Operating System

Software which controls the execution of computer programs and the flow of data to and from peripheral devices.

Overflow

See counter overflow.

Parity Setting

The configuration of the parity switches. Depending on the configuration of the parity output switch and the parity switch, a parity check bit is added to the end of data to make the sum of the total bits either even or odd. A parity check is performed after data has been transferred, and is accomplished by testing a unit of the data for either odd or even parity to determine whether an error has occurred in reading, writing, or transmitting the data.

Path

Also referred to as a directory (for example \users\projects).

Pass Through Mode

See Transparent Mode.

PC Interface

A program that runs on the HP Vectra and IMB PC/AT compatible computers. This is a friendly interface used to operate an HP 64700 Series emulator.

Performance Verification

A program that tests the emulator to determine whether the emulation and analysis hardware is functioning properly.

Physical Address Space

The address space in hardware memory and hardware I/O that is accessed by the microprocessor during normal program execution.

PQFP

A plastic quad flat pack is a type of integrated circuit.

P/O

An abbreviation for "part of." Used in illustrations to show that a part is shipped with other parts under a certain HP part number.

Prefetch

The ability of a microprocessor to fetch additional opcodes and operands before the current instruction is finished executing.

Prestore

The storage of states captured by the analyzer that precede states which are normally stored. If the normal storage qualifier specifies the entry address of a function or routine, prestore can be used to identify the callers of that function or routine.

Prestore Qualifier

A specification that must be met by a state before it can be saved in the analyzer prestore memory.

Primary Sequencer Branch

Occurs when the analyzer finds the primary branch state specified at a certain level and begins searching for the states specified at the primary branch's destination level.
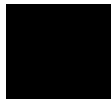
QFP

A quad flat pack is a type of integrated circuit.

Qualifier

A specification that must be met before an action can be taken by the analyzer. For example, a store qualifier is a specification that must be met by an incoming state before it can be stored in the trace memory. The "arm" condition can be used as an additional qualifier. For example, an external analyzer may be set up to supply a true signal to the rear panel BNC connector on the card cage when it detects a true condition in the target system. Then the analyzer can be set up to store qualify a certain kind of state, but only when the arm signal from the BNC is true.

Real-Time Execution

Continuous execution of the user program without interference from the emulator. (Such interference occurs when the emulator temporarily breaks into the monitor so that it can access register contents or target system memory or I/O.)

Relative Count

A count in the trace list count column that shows the count between the present displayed state and the state displayed immediately before it. Relative time count, for example, shows the elapsed time between the previous displayed state and the present state. Note that the count is between displayed states. If your trace list is inverse assembled and/or dequeued, several states may have been captured in memory between the present displayed state and the displayed state immediately before it.

Remote Configuration

The configuration in which an HP 64700 Series emulator is directly connected to a host computer via a single port. Commands are entered (typically from an interface program running on the host computer) and absolute code is downloaded into the emulator through that single port.

RS-232C

A standard serial interface used to connect computers and peripherals.

Secondary Sequencer Branch

Occurs when the analyzer finds the secondary branch state specified at a certain level before it found the primary branch state and begins searching for the states specified at the secondary branch's destination level.

Sequencer

The part of the analyzer that allows it to search for a certain sequence of states before triggering.

Single-step

The execution of one microprocessor instruction. Single-stepping the emulator allows you to view program execution one instruction at a time.

Software Breakpoint

Refer to execution breakpoint and access breakpoint in this glossary.

Software Performance Analyzer

An analyzer that measures execution of software modules, interaction between software modules, and usage of data points and I/O ports.

Standalone Configuration

The configuration in which a data terminal is used to control the HP 64700 Series emulator, and the emulator is not connected to a host computer.

stderr

An abbreviation for "standard error output." Standard error can be directed to various output devices connected to the HP 64700 ports.
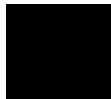
stdin

An abbreviation for "standard input." Standard input is typically defined as your computer keyboard.

stdout

An abbreviation for "standard output." Standard output can be directed to various output devices connected to the HP 64700 ports.

Step

See Single-step.

Store Qualifier

A specification that must be met by a state before it can be saved in the analyzer trace memory.

Synchronous Execution

The execution of multiple HP 64700 Series emulators/analyzers at the same time (i.e., multiple emulator start/stop).

Syntax

The order in which expressions are structured in command languages. Syntax rules determine which forms of command language syntax are grammatically acceptable.

Target Program

The program you are developing for your product. It is also called user program.

Target System

The circuitry where the emulator probe is connected (typically a microprocessor-based system under development).

Target System Memory

Storage that is present in the target system.

Terminal Interface

The command interface present inside the HP 64700 Series emulators that is used when the emulator is connected to a simple data terminal. This interface provides on-line help, command recall, macros, and other features which provide for easy command entry from a terminal.

TQFP

A thin quad flat pack is a type of integrated circuit.

Trace

A collection of states captured synchronously by the analyzer.

Transparent Configuration

The configuration in which the HP 64700 Series emulator is connected between a data terminal and a host computer. When the emulator is in the transparent (pass through) mode, the data terminal acts like a normal terminal connected to the computer. In this configuration, you can develop code on the host computer and download absolute code into the emulator for debugging and testing.

Transparent Mode

The emulator mode in which all characters received on one port will be copied to the other port. This mode allows a data terminal (connected to one emulator port) to access a host computer (connected to the other emulator port) through the emulator.

Trigger

The condition that identifies a reference state within an analyzer trace measurement. Trigger also refers to the analyzer signal that becomes active when the trigger condition is found.

Trigger signals called trig1 and trig2 are bidirectional signal lines that can be used to coordinate measurement activity between emulators and analyzers installed in the instrumentation card cage, and between instruments connected to the BNC on the rear panel of the card cage.

Note that there is a delay when you use a trigger for measurement coordination. For example, you may specify that the emulator break to

its monitor program when it receives trig1 from the analyzer. Several states may be executed in the emulator between the time the analyzer recognizes its trigger condition, generates trig1, delivers trig1 to the emulator, and the emulator responds to trig1 by breaking to its monitor program.

Uploading

The transfer of emulation or target system memory contents to a host computer.

Unlocked Exit

One of two methods used to leave the high level (Graphical or Softkey) Interface and return to the host computer operating system. An unlocked exit command allows you to exit the high level interface and re-enter later with the default configuration. (See also Locked Exit.) This is not available in the Terminal Interface.

User Program

Another name for your target program (the program you are developing for your product.

Viewport

See Window.

Wait States

Extra microprocessor clock cycles that increase the total time of a bus cycle. Wait states are typically used when slower memory is implemented.

Window

A specified rectangular area of virtual space shown on the display in which data can be observed.

1K Analyzer

The term "1K analyzer" refers to the HP 64704 Emulation-Bus
Analyzer with 1K trace memory.

!

When shown in the trace list count column of the terminal interface or
the PC interface, the exclamation mark "!" indicates counter overflow.

Index

DECLARATION OF CONFORMITY

according to ISO/IEC Guide 22 and EN 45014

Manufacturer's Name: Hewlett-Packard Company

Manufacturer's Address: Colorado Springs Division
1900 Garden of the Gods Road
Colorado Springs, CO 80907 USA

declares, that the product

Product Name: Microprocessor Emulator (HP 64700 Series)

Model Number(s):   HP 64782B/C/D/E/F/G/H/J

Product Option(s):   All

conforms to the following Product Specifications:

Safety: IEC 1010-1:1990+A1 / EN 61010-1:1993
UL 3111
CSA-C22.2 No. 1010.1:1993
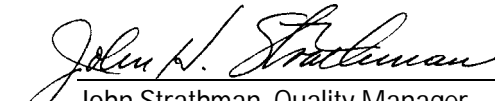
EMC:		
CISPR 11:1990 / EN 55011:1991		Group 1 Class A
IEC 801-2:1991 / EN 50082-1:1992		4 kV CD, 8 kV AD
IEC 801-3:1984 / EN 50082-1:1992		3 V/m, {1kHz 80% AM, 27-1000 MHz}
IEC 801-4:1988 / EN 50082-1:1992		0.5 kV Sig. Lines, 1 kV Power Lines

Supplementary Information:

The product herewith complies with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC and carries the CE marking accordingly.

This product was tested in a typical configuration with Hewlett-Packard test systems.

Colorado Springs,9/20/96

John Strathman, Quality Manager

Product Regulations

Safety	IEC 1010-1:1990+A1 / EN 61010-1:1993
	UL 3111
	CSA-C22.2 No. 1010.1:1993

EMC

CE
ISM 1-A

This Product meets the requirement of the European Communities (EC) EMC Directive 89/336/EEC.

Emissions	EN55011/CISPR 11 (ISM, Group 1, Class A equipment)		
Immunity	EN50082-1	Code[1]	Notes[2]
	IEC 801-2 (ESD) 4kV CD, 8kV AD	3	1,2,3
	IEC 801-3 (Rad.) 3 V/m	1	3
	IEC 801-4 (EFT) 0.5 kV, 1kV	1	3

[1] Performance Codes:
  1 PASS - Normal operation, no effect.
  2 PASS - Temporary degradation, self recoverable.
  3 PASS - Temporary degradation, operator intervention required.
  4 FAIL - Not recoverable, component damage.

[2] Notes:
  1 Electrostatic discharge (ESD) to the 64700B mainframe may cause degraduation in performance requiring operator intervention.
  2 The active probe assembly is sensitive to ESD events. Use standard ESD preventative practices to avoid component damage.
  3 The CMB and active probe power connectors were not subjected to immunity testing.

Sound Pressure Level	Less than 60 dBA

Certification and Warranty

Certification

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Bureau of Standards, to the extent allowed by the Bureau's calibration facility, and to the calibration facilities of other International Standards Organization members.

Warranty

This Hewlett-Packard system product is warranted against defects in materials and workmanship for a period of 90 days from date of installation. During the warranty period, HP will, at its option, either repair or replace products which prove to be defective.

Warranty service of this product will be performed at Buyer's facility at no charge within HP service travel areas. Outside HP service travel areas, warranty service will be performed at Buyer's facility only upon HP's prior agreement and Buyer shall pay HP's round trip travel expenses. In all other cases, products must be returned to a service facility designated by HP.

For products returned to HP for warranty service, Buyer shall prepay shipping charges to HP and HP shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to HP from another country. HP warrants that its software and firmware designated by HP for use with an instrument will execute its programming instructions when properly installed on that instrument. HP does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error free.

Limitation of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environment specifications for the product, or improper site preparation or maintenance.

No other warranty is expressed or implied. HP specifically disclaims the implied warranties of merchantability and fitness for a particular purpose.

Exclusive Remedies

The remedies provided herein are buyer's sole and exclusive remedies. HP shall not be liable for any direct, indirect, special, incidental, or consequential damages, whether based on contract, tort, or any other legal theory.

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales and Service Office.

Safety

Summary of Safe Procedures

The following general safety precautions must be observed during all phases of operation, service, and repair of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the instrument. Hewlett-Packard Company assumes no liability for the customer's failure to comply with these requirements.

Ground The Instrument

To minimize shock hazard, the instrument chassis and cabinet must be connected to an electrical ground. The instrument is equipped with a three-conductor ac power cable. The power cable must either be plugged into an approved three-contact electrical outlet. The power jack and mating plug of the power cable meet International Electrotechnical Commission (IEC) safety standards.

Do Not Operate In An Explosive Atmosphere

Do not operate the instrument in the presence of flammable gases or fumes. Operation of any electrical instrument in such an environment constitutes a definite safety hazard.

Keep Away From Live Circuits

Operating personnel must not remove instrument covers. Component replacement and internal adjustments must be made by qualified maintenance personnel. Do not replace components with the power cable connected. Under certain conditions, dangerous voltages may exist even with the power cable removed. To avoid injuries, always disconnect power and discharge circuits before touching them.

Designed to Meet Requirements of IEC Publication 348

This apparatus has been designed and tested in accordance with IEC Publication 348, safety requirements for electronic measuring apparatus, and has been supplied in a safe condition. The present instruction manual contains some information and warnings which have to be followed by the user to ensure safe operation and to retain the apparatus in safe condition.

Do Not Service Or Adjust Alone

Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

Do Not Substitute Parts Or Modify Instrument

Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification of the instrument. Return the instrument to a Hewlett-Packard Sales and Service Office for service and repair to ensure that safety features are maintained.

Dangerous Procedure Warnings

Warnings, such as the example below, precede potentially dangerous procedures throughout this manual. Instructions contained in the warnings must be followed.

Warning	Dangerous voltages, capable of causing death, are present in this instrument. Use extreme caution when handling, testing, and adjusting.

Safety Symbols Used In Manuals

The following is a list of general definitions of safety symbols used on equipment or in manuals:

Instruction manual symbol: the product is marked with this symbol when it is necessary for the user to refer to the instruction manual in order to protect against damage to the instrument.

Hot Surface. This symbol means the part or surface is hot and should not be touched.

Indicates dangerous voltage (terminals fed from the interior by voltage exceeding 1000 volts must be marked with this symbol).
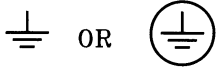
Protective conductor terminal. For protection against electrical shock in case of a fault. Used with field wiring terminals to indicate the terminal which must be connected to ground before operating the equipment.

Low-noise or noiseless, clean ground (earth) terminal. Used for a signal common, as well as providing protection against electrical shock in case of a fault. A terminal marked with this symbol must be connected to ground in the manner described in the installation (operating) manual before operating the equipment.
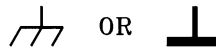
rame or chassis terminal. A connection to the frame (chassis) of the equipment which normally includes all exposed metal structures.
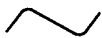
Alternating current (power line).

Direct current (power line).

Alternating or direct current (power line).

| CAUTION | The Caution sign denotes a hazard. It calls your attention to an operating procedure, practice, condition, or similar situation, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the product. |

| WARNING | The Warning sign denotes a hazard. It calls your attention to a procedure, practice, condition or the like, which, if not correctly performed, could result in injury or death to personnel. |