About this Manual

We've added this manual to the Agilent website in an effort to help you support your product. This manual is the best copy we could find; it may be incomplete or contain dated information. If we find a more recent copy in the future, we will add it to the Agilent website.

Support for Your Product

Agilent no longer sells or supports this product. Our service centers may be able to perform calibration if no repair parts are needed, but no other support from Agilent is available. You will find any other available product information on the Agilent Test & Measurement website, www.tm.agilent.com.

HP References in this Manual

This manual may contain references to HP or Hewlett-Packard. Please note that Hewlett-Packard's former test and measurement, semiconductor products and chemical analysis businesses are now part of Agilent Technologies. We have made no changes to this manual copy. In other documentation, to reduce potential confusion, the only change to product numbers and names has been in the company name prefix: where a product number/name was HP XXXX the current name/number is now Agilent XXXX. For example, model number HP8648A is now model number Agilent 8648A.

HEWLETT PACKARD

Installation/Service/Terminal Interface User's Guide

Motorola 6830x-Family Emulator/Analyzer HP 64798C/E/F

Notice

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

Hewlett-Packard Company
P.O. Box 2197
1900 Garden of the Gods Road
Colorado Springs, CO 80901-2197, U.S.A.

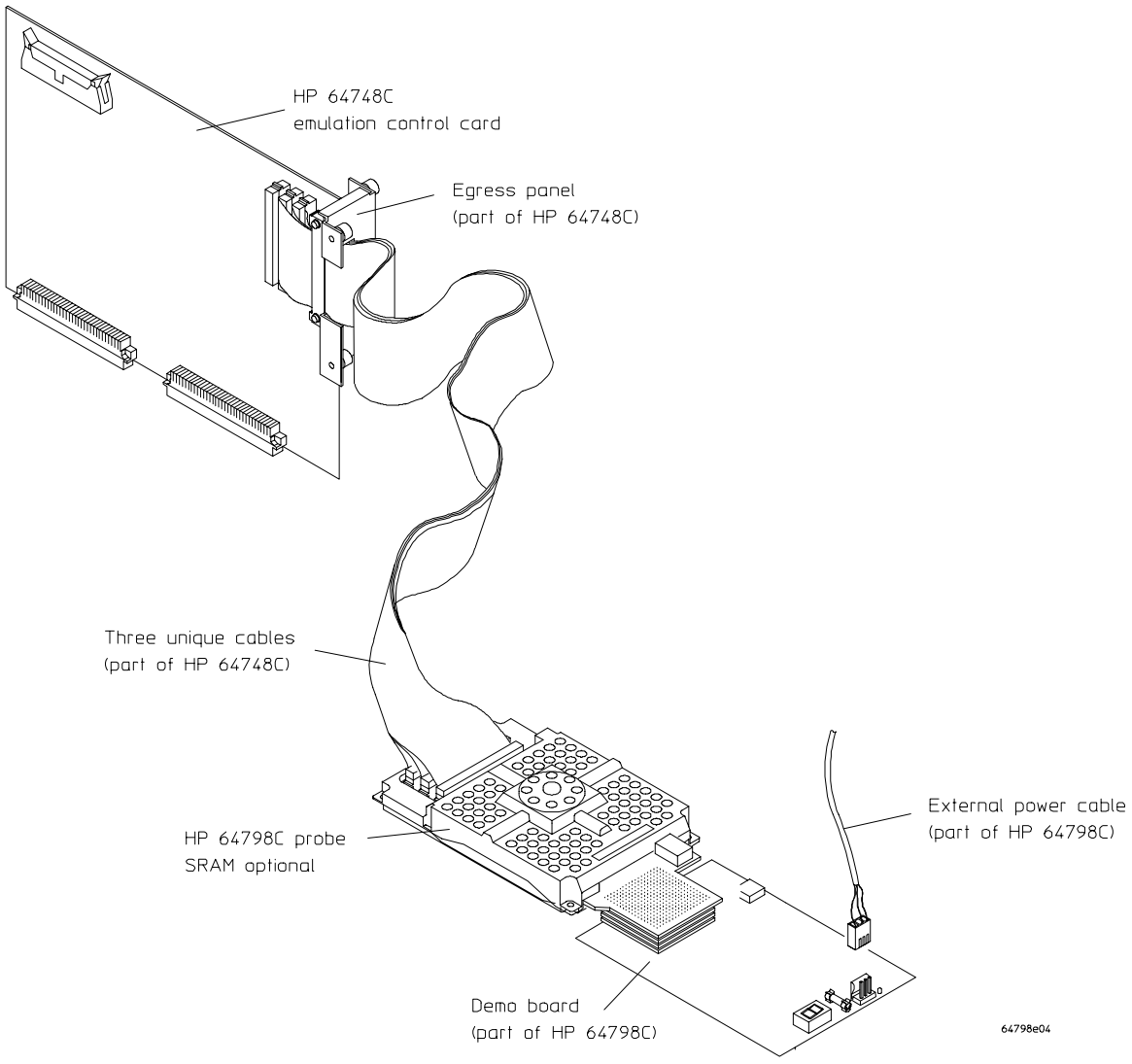
Printing History

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	64798-97000, July 1996
Edition 2	64798-97001, August 1996
Edition 3	64798-97002, November 1996
Edition 4	64798-97003, December 1996
Edition 5	**64798-97004, 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)
HP 64798C probe
SRAM optional
External power cable
(part of HP 64798C)
Demo board
(part of HP 64798C)
64798e04

The HP 64798C/E/F Emulator

Description

The Hewlett-Packard Model 64798C/E/F emulator supports the Motorola 6830x microprocessor family. The emulators and supported microprocessors include HP 64798C/MC68302, HP 64798E/MC68EN302, and HP 64798F/MC68LC302. In this manual, the HP emulators are referred to by Model Number 64798, and the Motorola microprocessors are referred to by MC6830x, except where specific information applies only to certain individual members of the emulator or microprocessor family.

The emulator probe has a PGA-style connector. It can be plugged directly into the PGA connector on the demo board. Using optional accessories, it can be plugged into a 144-pin TQFP, a 132-pin PQFP, or a 100-pin TQFP target system. The 68302 emulator can also be plugged into a PGA connector on the target system using an optional transition socket. Find the list of probing accessories in the parts list in Chapter 7 and select the probe assembly kit or transition socket you need to connect this emulator to your 6830x target system.

Note that the PGA-style connector on the emulator probe is NOT compatible with the MC68302 PGA pinout. If you wish to connect the emulator to an MC68302 PGA pinout on your target system, use the PGA transition socket listed in chapter 7.

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. A full 56 Kbytes of dual-port emulation memory is built into the emulator. Up to 8 megabytes of additional emulation memory may be installed on the probe. High performance download is achieved through the use of an optional LAN interface. An RS-232 port 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.

Features

HP 64798 Emulator

- Active probe emulator; supports the fastest processor speeds currently available from Motorola. At the time this manual was printed, the emulator had been tested to the following processor speeds:
  - 25 MHz for MC68302
  - 25 MHz for MC68EN302
  - 25 MHz for MC68LC302
- 5V operation (MC68302 and MC68EN302)
- 3.3V and 5V operation (MC68LC302)
- Supports the following clock speeds
  - 8 MHz to 25 MHz (MC68302 and MC68EN302)
  - 0 MHz to 25 MHz (MC68LC302)
- No wait states required
- Unlimited software breakpoints
- Symbolic support
- 36-inch cable and 219-mm (8.6-inch) x 109 mm (4.3-inch) probe
- Background monitor
- 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

- 80-channel emulation-bus analyzer
- Post-processed 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
- 56 Kbytes of dual-ported emulation memory built in
- Mapping resolution is 256 bytes
- For MC68302, MC68EN302, and MC68LC302, no wait states required

In This Book

This user's guide covers the Hewlett-Packard Model 64798 emulator for the MC6830x 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 PC, and connect the emulator probe into the demo board and into your target system.

Chapter 4 in Part 1 leads you through a detailed procedure designed to help you connect the emulator into your target system and obtain satisfactory operation. Even if you have extensive emulator experience, you should read Chapter 4 to become familiar with problems that are specific to this processor and emulator.

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

3 Connecting to the Demo Board

4 Connecting the Emulator to a Target System

Part 2 Service

5 Installing/Updating HP 64700 Firmware

6 Solving Problems

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 64798 emulator probe to the HP 64748C emulator control card.
3. Install cards in the HP 64700 card cage.
4. Install emulation memory modules on the emulator probe.
5. Installing optional clock modules.
6. Connect a power cord to the HP 64700 Card Cage.

What you need

Equipment supplied

The minimum system contains:

- HP 64798 MC6830x 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 80-channel Emulation-Bus Analyzer (1K analyzer) card.
- 80-Channel Analyzer Extender Ribbon Cable.
- 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).
- Probing accessories for connecting to a target system (see Chapter 7).

Equipment and tools needed

In order to install and use the MC6830x emulation system, you need:

- Probe assembly kit for connecting the emulator probe to a target system (not needed for connecting to the demo board). See Chapter 7.
- 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.

Installing a different emulator

If you already have an HP 64700-Series Card Cage and want to remove the existing emulator and insert an HP 64798 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.

Installing a different analyzer

If you already have an 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)

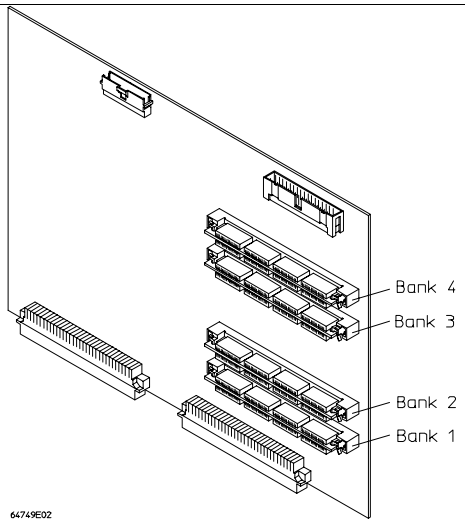
Observe antistatic precautions

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 the 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.

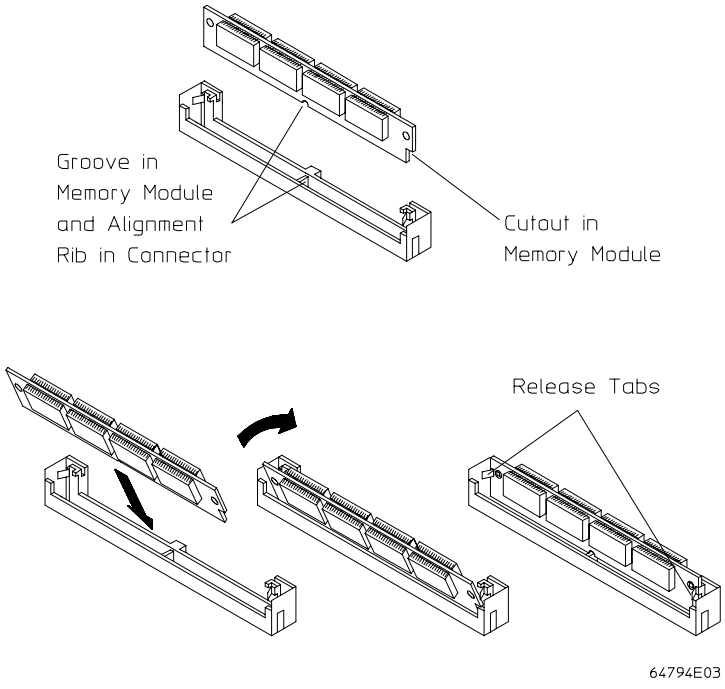
Align the cutout in the memory module with the projection in the connector.

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.



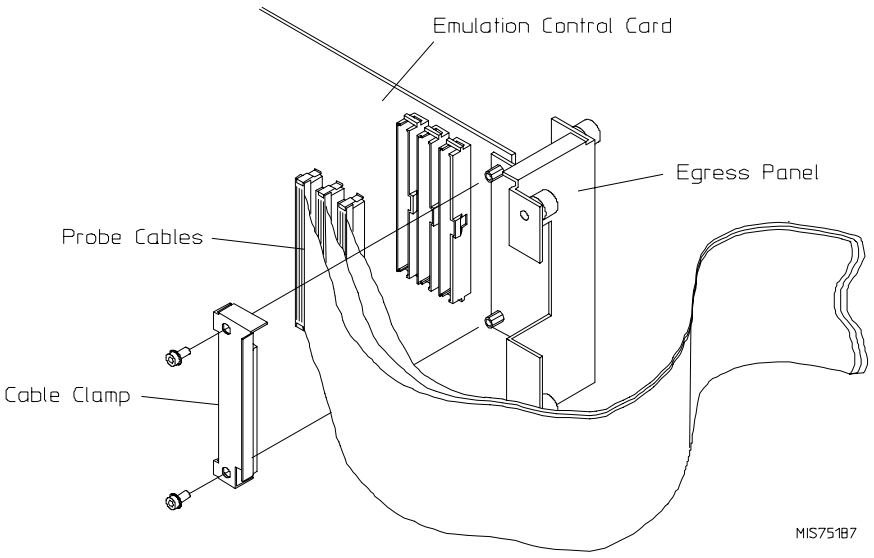
Step 2. Connect the emulator probe cables

Three ribbon cables connect the HP 64748C emulation control card to the HP 64798 emulator probe.

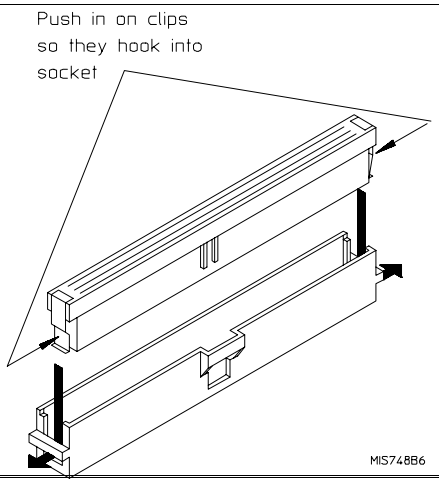
The shortest cable connects from J1 of the emulation control card to J3 of the emulator probe. The medium length cable connects from J2 of the emulation control card to J2 of the emulator probe. The longest cable connects from J3 of the emulation control card to J1 of the emulator probe.

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).

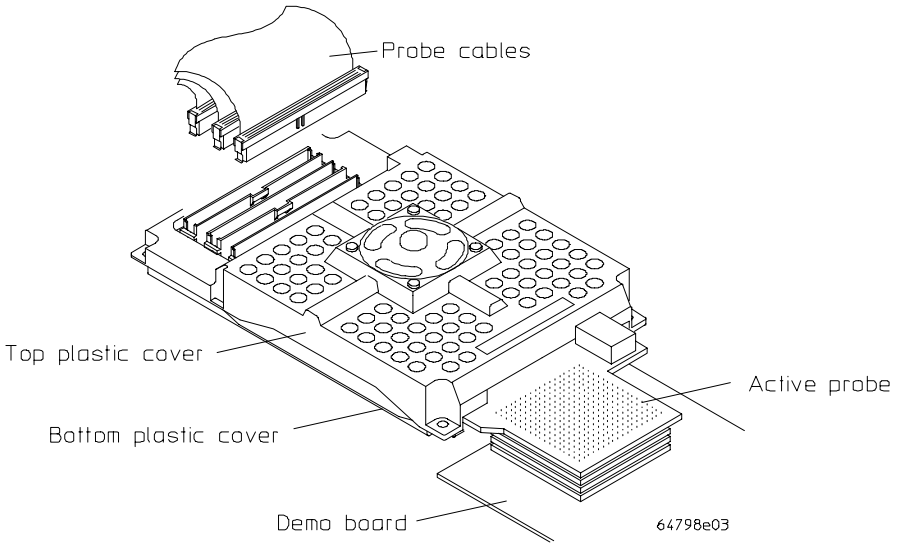
1 Connect the emulator probe cables to the emulation control card.



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.



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.

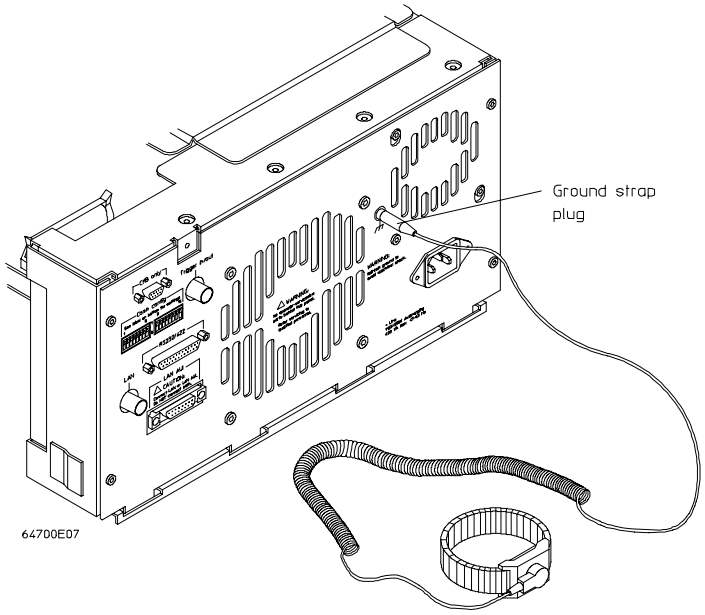


Step 3. Install boards into the HP 64700 Card Cage

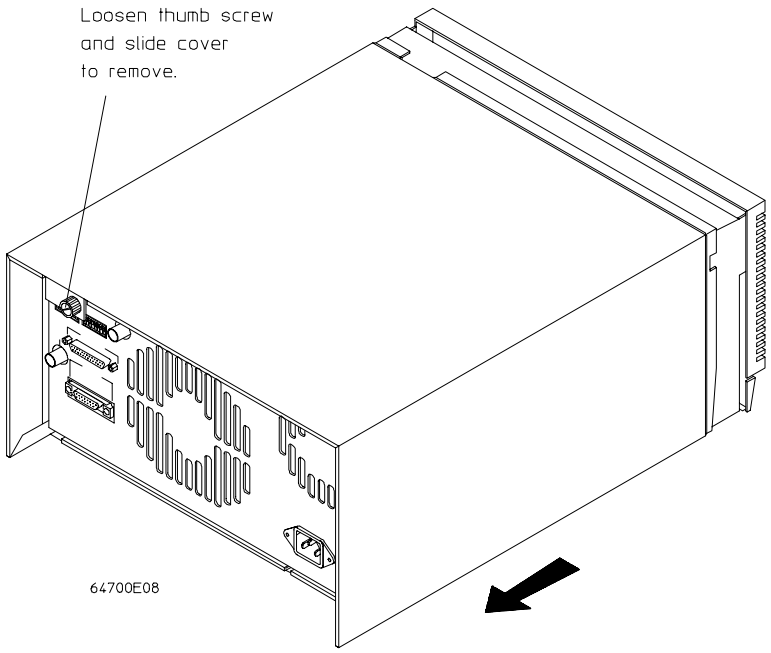
WARNING Electrical shock hazard. Before removing or installing parts in the HP 64700 Card Cage, make sure the card cage power is off and the power cord is disconnected.

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

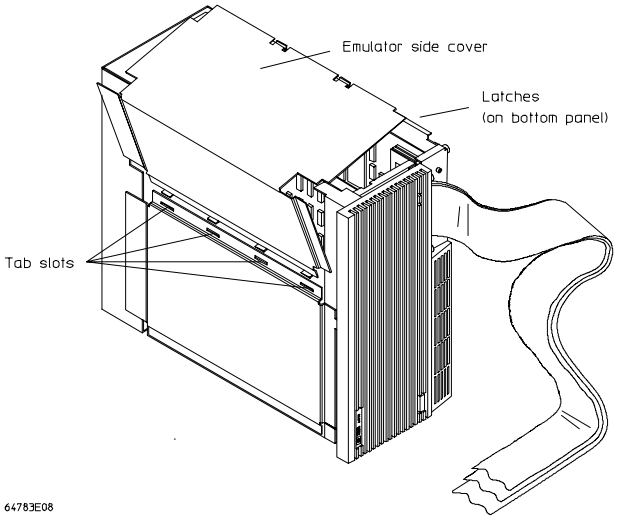
1 Use a ground strap when removing or installing boards into the HP 64700 Card Cage to reduce the risk of damage to the circuit cards from static discharge. A jack on the rear panel of the HP 64700 Card Cage is provided for this purpose.



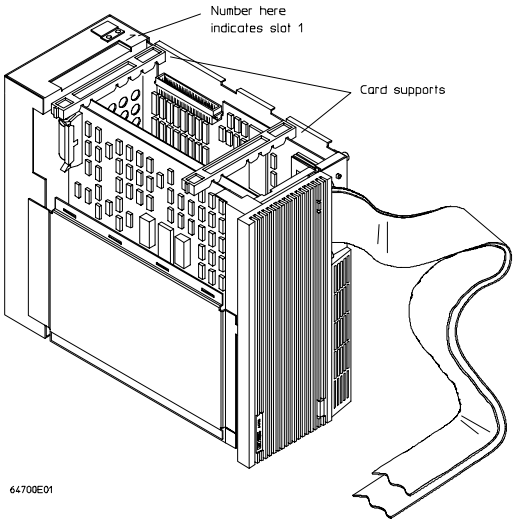
2 Turn the thumb screw and remove the top cover by sliding the cover toward the rear and up.



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

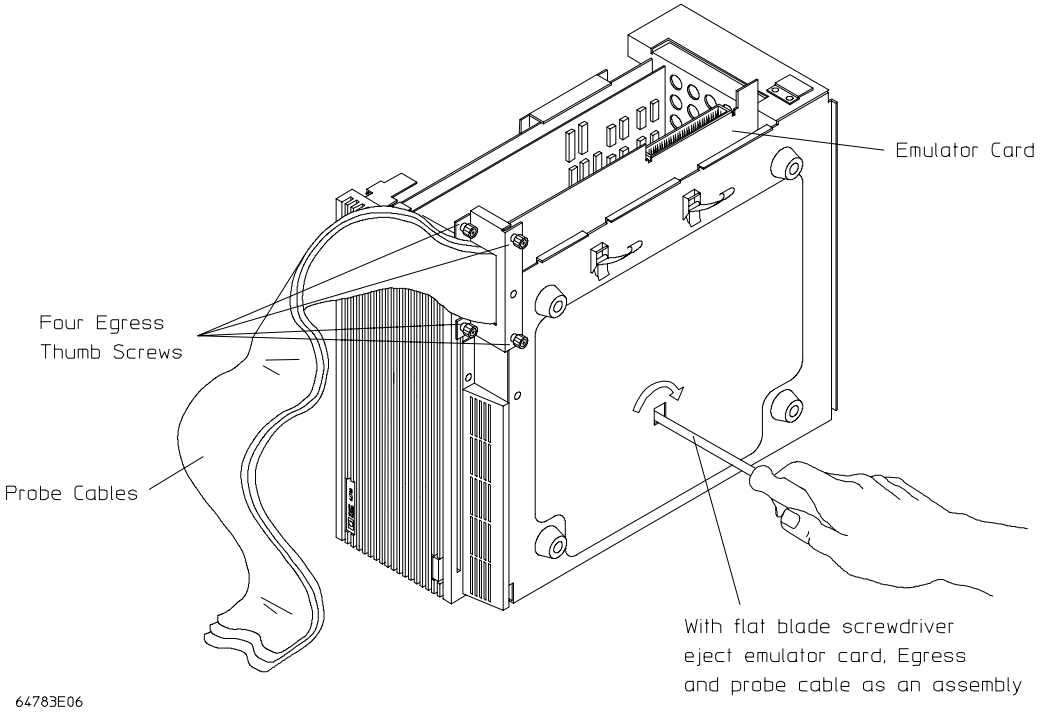


4 Remove the card supports.



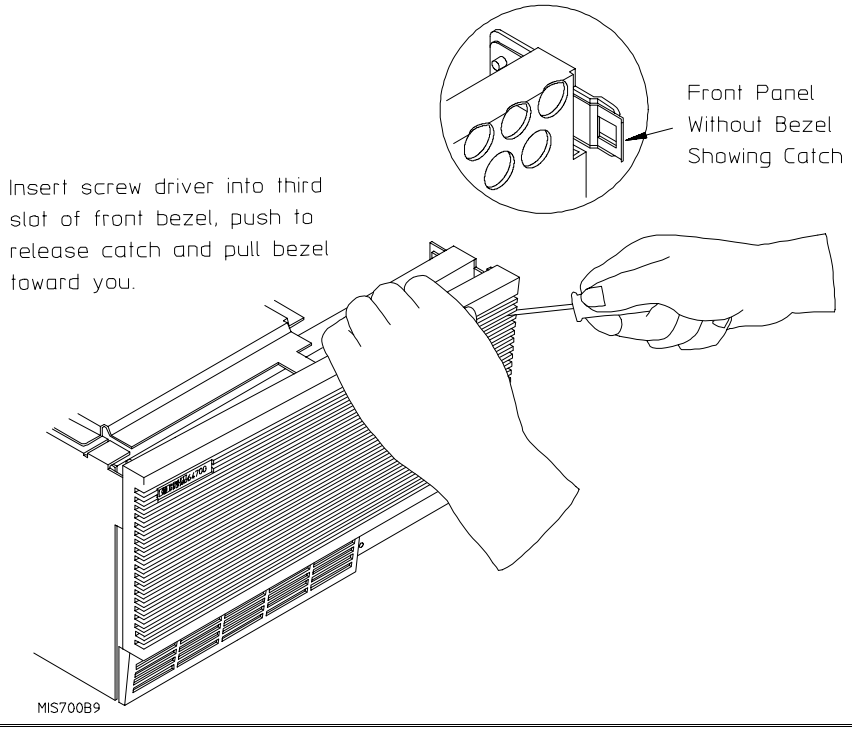
5 Completely loosen the four egress thumb screws.

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

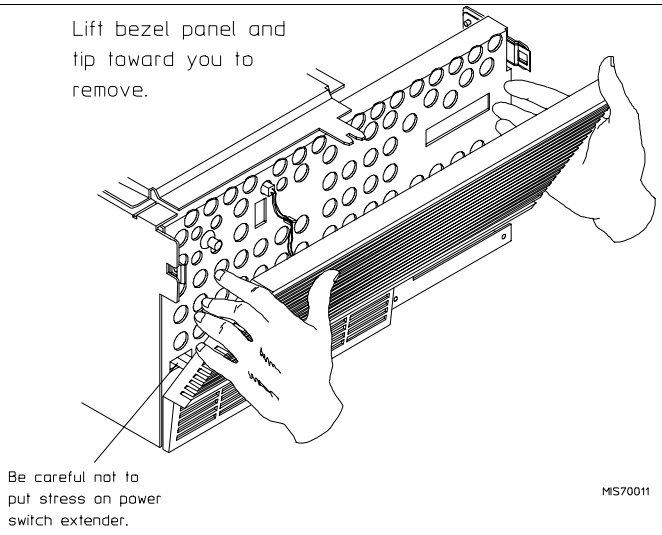


6 Insert a screw driver into the third slot of the right side of the front bezel, push to release catch, and pull the right side of the bezel about one-half inch away from the front of the HP 64700. Then, do the same thing on the left side of the bezel. When both sides are released, pull the bezel toward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

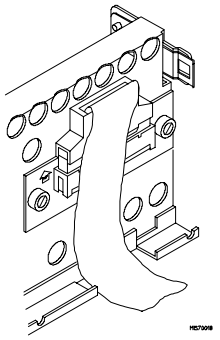
Be careful because the plastic ears are easily broken on the front bezel.



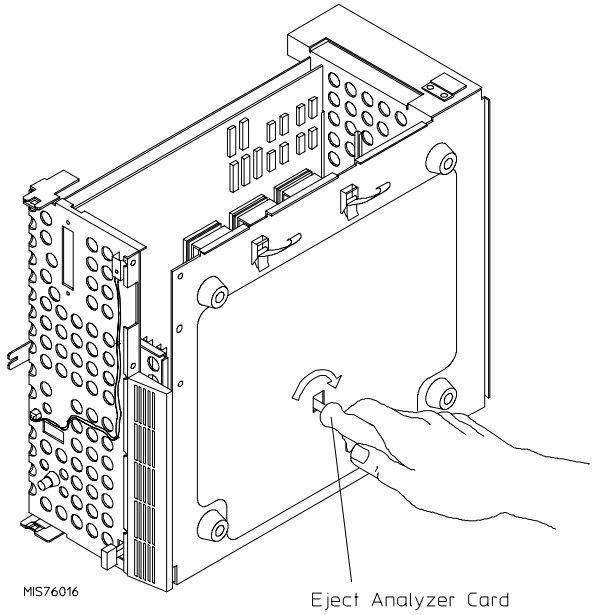
7 Lift the bezel panel to remove. Be careful not to put stress on the power switch extender.



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.

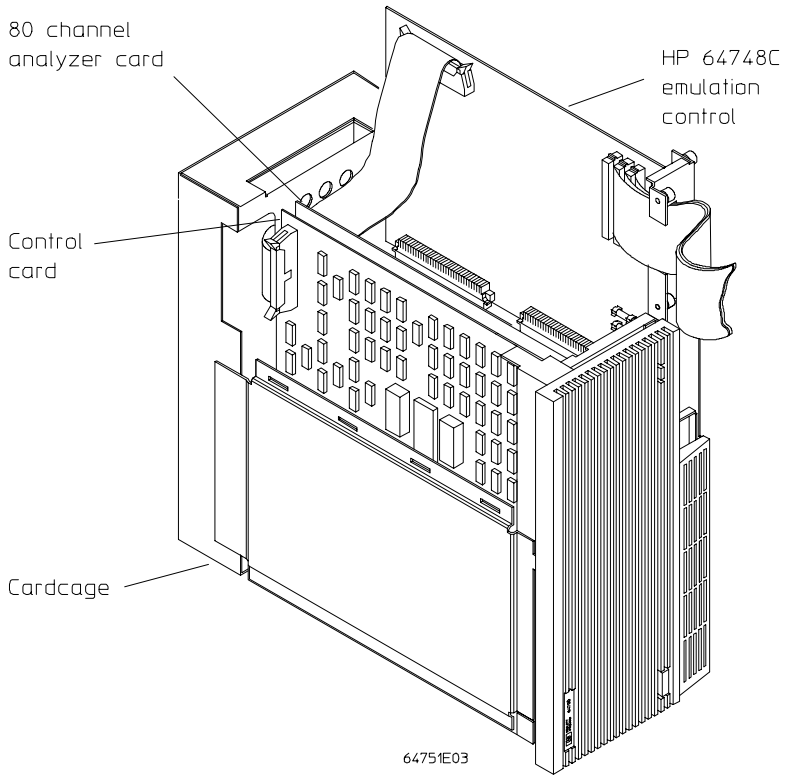


Do not remove the system control board. This board is used in all HP 64700 emulation and analysis systems.

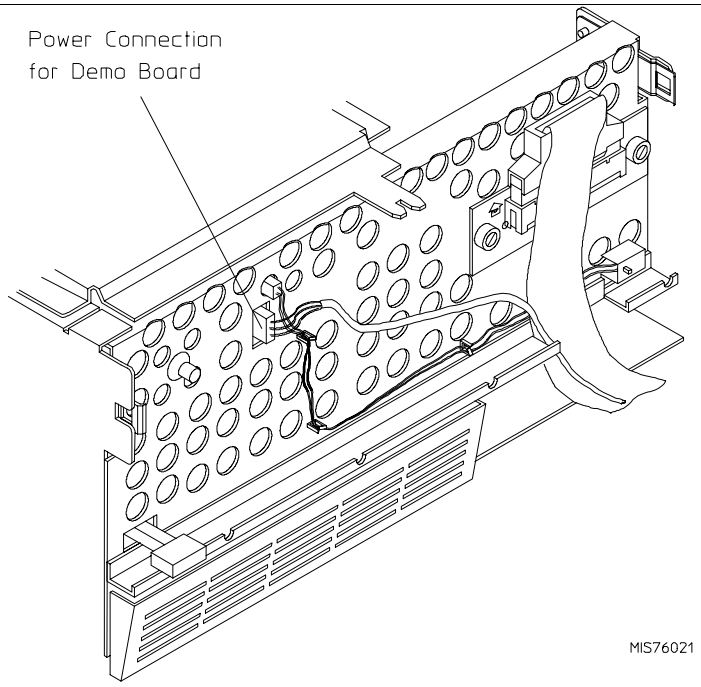
10 Install the analyzer and emulation control cards. The analyzer is installed in the slot next to the system control card. The emulation control card is installed in the second slot from the bottom of the card cage. The software performance analyzer card may occupy any slot between the emulation-bus 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.

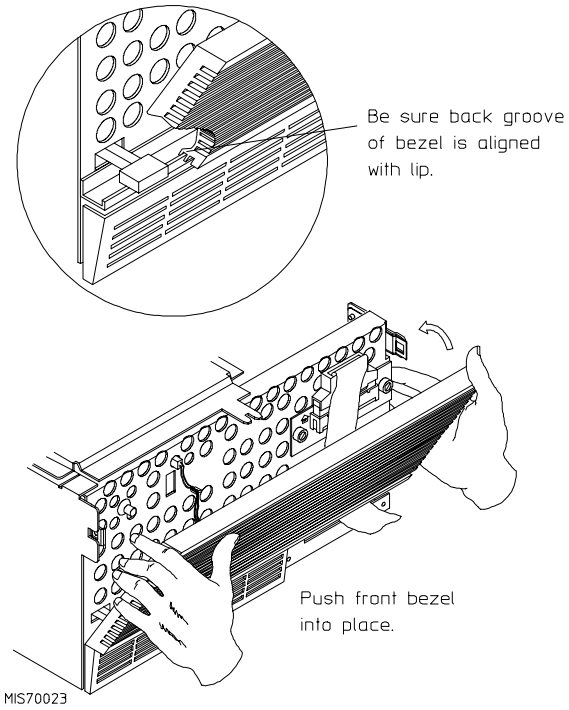
Attach the ribbon cable from the emulation control card to the analyzer card, and to the software performance analyzer, if installed. Tighten the thumbscrews that hold the emulation control card to the cardcage frame.



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



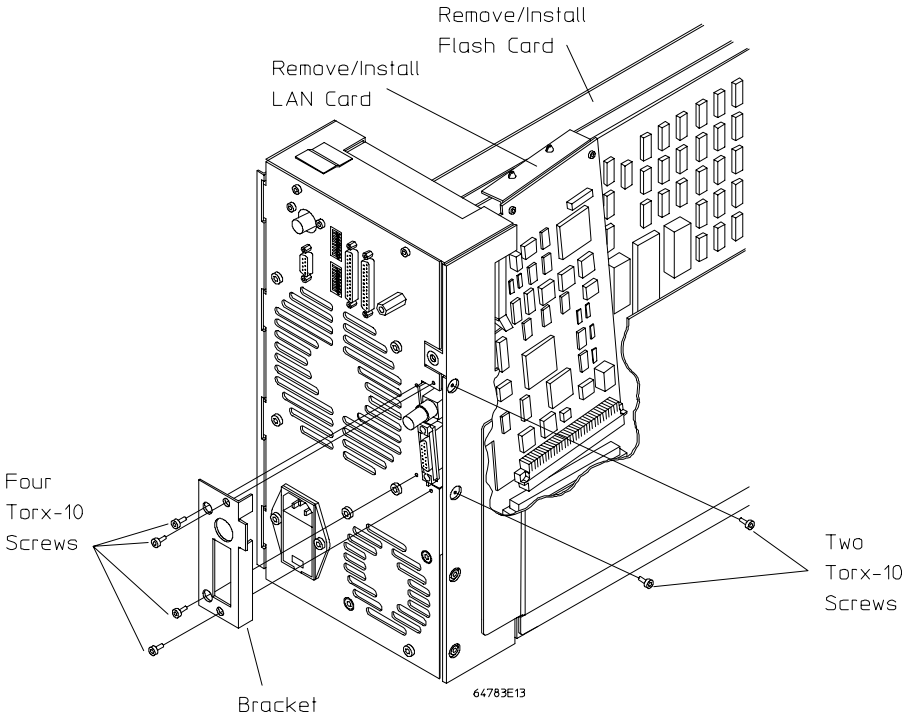
12 To reinstall the front bezel, be sure that the bottom rear groove of the front bezel is aligned with the lip as shown below.



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

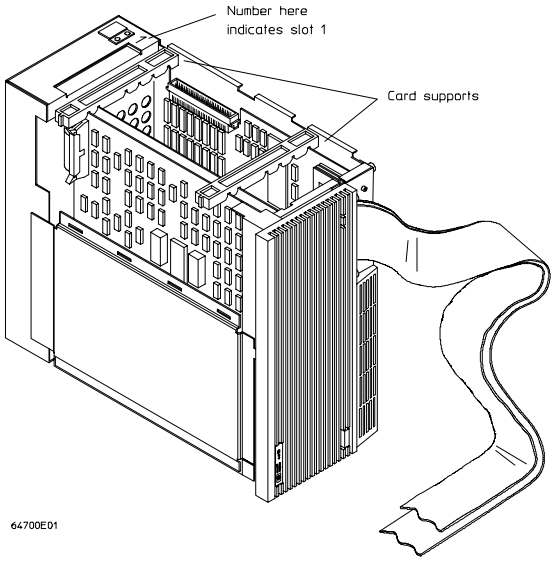
Before you install the LAN card, you must remove the rear panel cover plate that is replaced by the bracket shown in the figure below.

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.

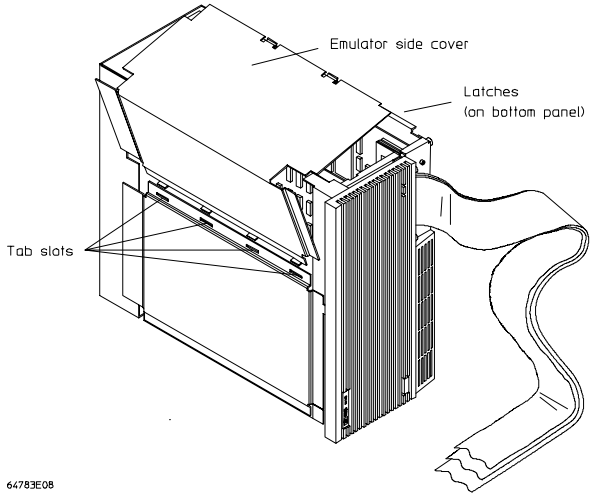


14 *This step applies only to the HP 64700A*. If you wish to install the flash card (used for updating firmware, see Chapter 5), refer to the diagram above. Install the flash card in any available slot between the 80-channel analyzer card and the HP 64748C control card in the cardcage. Insert the flash card in the plastic guides. Make sure the connectors are properly aligned. Then press the card all the way into the mother board sockets.

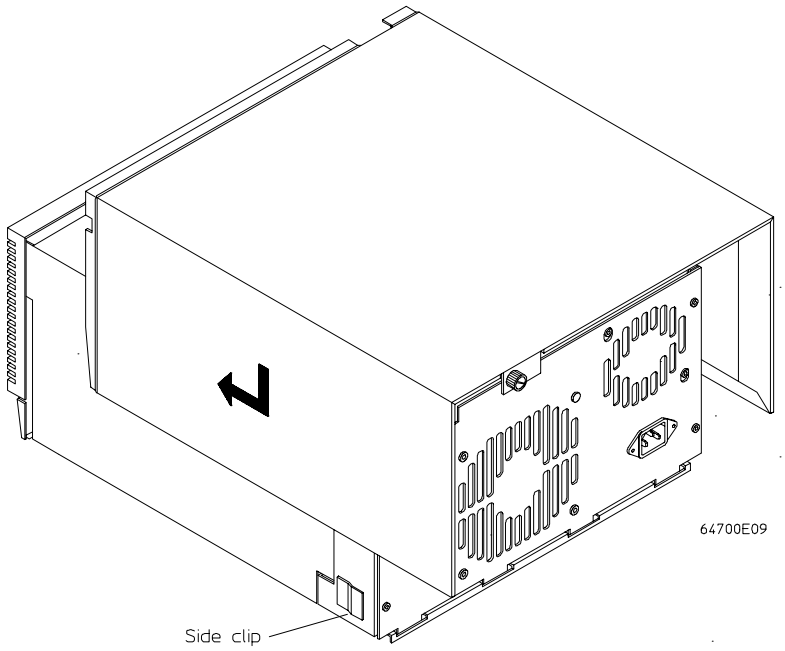
15 Install the card supports.



16 To install the side cover, insert the side cover into the tab slots and fasten the two latches.



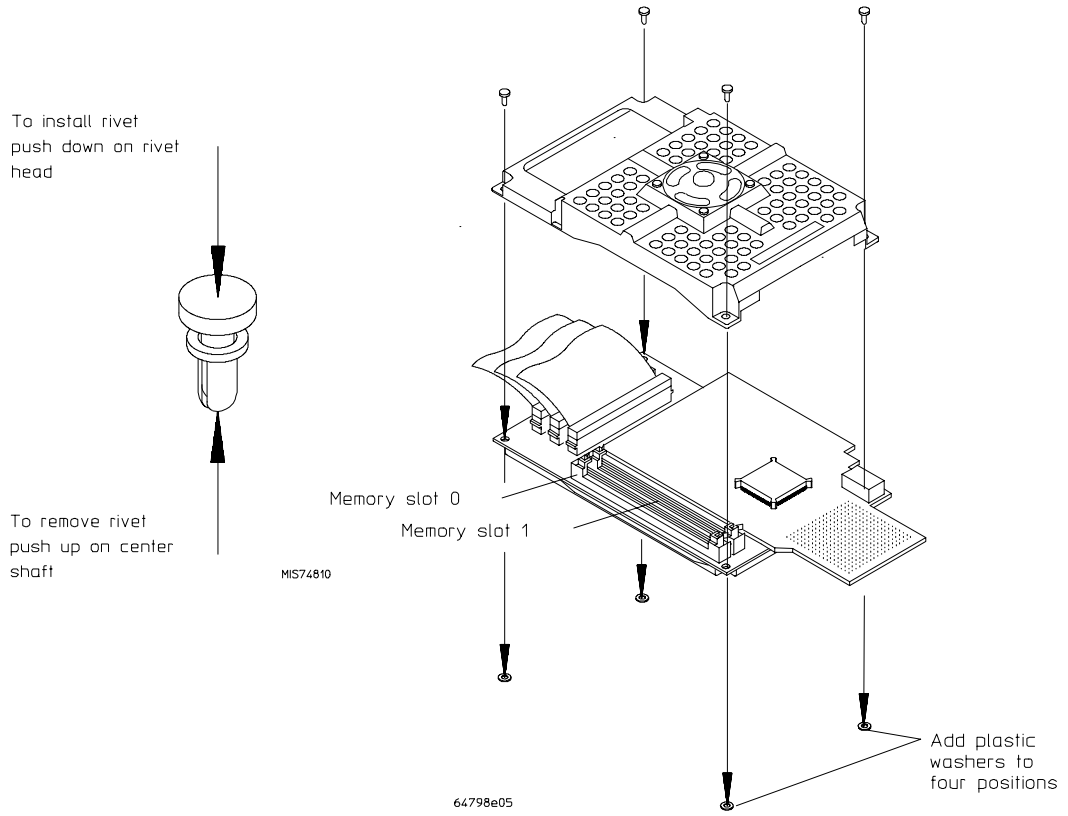
17 Install the top cover in reverse order of its removal, but make sure that the side panels of the top cover are attached to the side clips on the frame.



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 top cover. The bottom cover is never removed, except when you need to replace a defective active probe on the exchange program.



2 Emulation memory modules (SIMM) can be installed in either or both memory slots on the emulator probe. The emulator software will automatically arrange the most efficient use of SIMM hardware when mapping memory. See "How memory is allocated in the Emulation Probe," below.

Three types of SIMMs may be installed: 256 Kbyte (HP 64171A/64172A), 1 Mbyte (HP 64171B/64172B), and 4 Mbyte (HP 64173A).

If you need new rivets or washers to complete the installation procedure, refer to Chapter 7, "Parts List," for the part number of the Plastic Rivets Kit.

How memory is allocated in the Emulation Probe

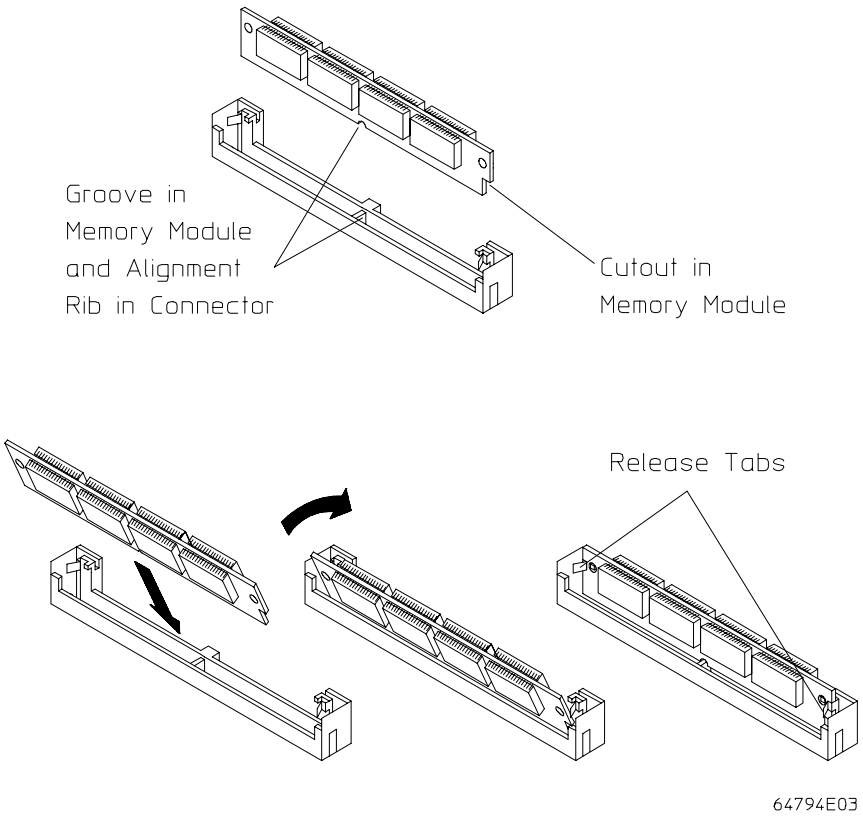
Number of emulation memory modules (SIMM) installed	Blocks of SIMM available for mapping (Notes 1, 2)	Blocks of built-in, dual-port memory available for mapping (Notes 3, 4)	Maximum number of memory blocks available for mapping
0	0	7	7
1	4	7 (Note 5)	8
2 (Notes 6, 7)	8	7 (Note 5)	8

Notes:
1. SIMM hardware is not dual-port, but built-in memory is dual-port.
2. Each SIMM is divided into four equal blocks by emulator software. Each block is 1/4 the size of the associated SIMM.
3. Built-in emulation memory contains seven 8-Kbyte, dual-port blocks.
4. 8-Kbyte blocks of built-in memory can be substituted for SIMM blocks using a command such as, "map 1000..1fff eram dp."
5. By default, the emulator maps memory to blocks of SIMM before mapping memory to the built-in, 8-Kbyte, dual-port blocks. To override the default, include "dp" in your map command. See Note 4.
6. Different size SIMMs can be installed in either memory slot.
7. The emulator automatically selects the most appropriate block size to contain each mapped address range.

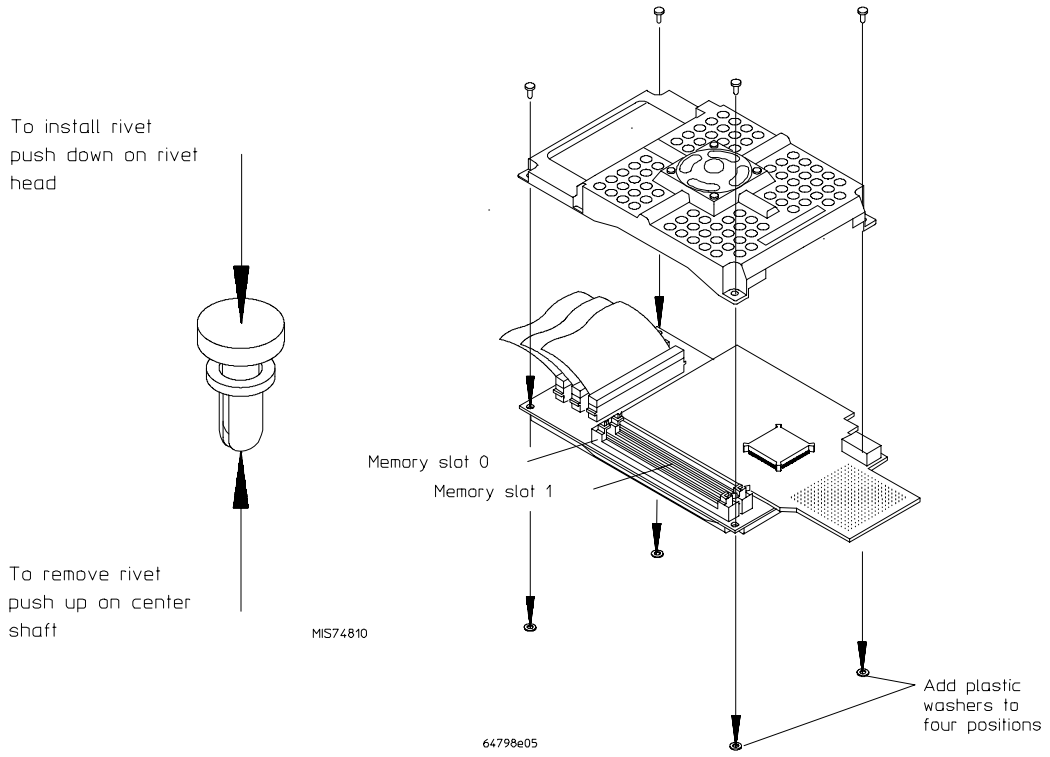
3 Install emulation memory modules on the emulator probe. There is a cutout at one end of the memory modules so they can only be installed the correct way.

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.



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



Step 5. Select a clock module

The MC6830x emulator can operate with either an external clock (supplied from the target system), or an internal clock (supplied from a module installed on the emulator probe clock socket). In addition, the MC6830x processors can operate from either an oscillator or a crystal. The table below will help you select the clock source that is appropriate for the microprocessor and clock mode being emulated.

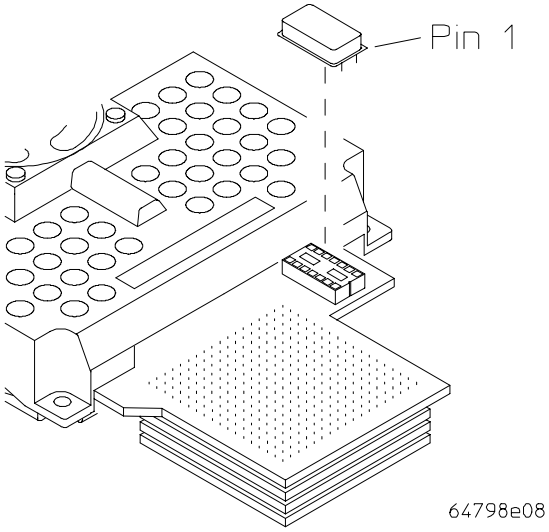
Microprocessor and Clock Mode	**Clock source for emulator**	
	Internal **cf clk = int**	**External** **cf clk = ext**
68302 and 68EN302 with oscillator	oscillator module: 20 MHz supplied by HP or 8-25 MHz supplied by user	Any oscillator or crystal module. This provides a working internal clock which must be detected during interface initialization.
68302 and 68EN302 with crystal	24 MHz (Module E), or User 1 (Module A), or User 2 (Module B)	Not supported.
68LC302 with oscillator	oscillator module: 20 MHz supplied by HP or 0-25 MHz supplied by user	Any oscillator or crystal module. This provides a working internal clock which must be detected during interface initialization.
68LC302 with crystal	32.768 kHz (Module C), or 4.194 MHz (Module D), or User 1 (Module A), or User 2 (Module B)	Not supported.

Refer to Chapter 4 for details of making the choices described in the table above. If you wish to develop your own clock source with the User 1 or User 2 modules, example schematics are shown in Chapter 4.

When using the internal clock, you can install either an oscillator or a crystal in the emulator probe clock socket. The diagram below shows how to install an internal clock module.

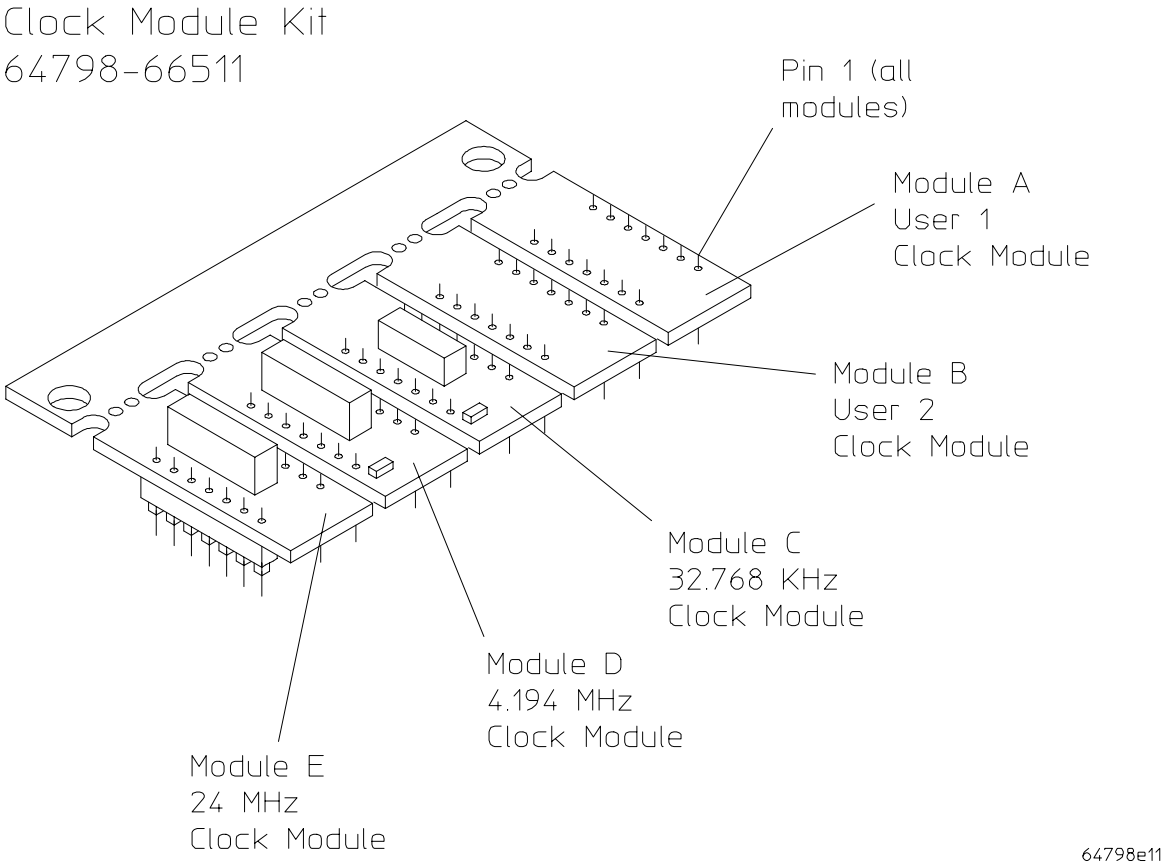
The emulator probe clock socket is a 14-pin DIP socket located on the top of the emulator probe. A standard 14-pin DIP type CMOS oscillator operating within specifications appropriate for your emulator (see table on previous page) may be installed in the clock socket. Pin 1 provides Oscillator Enable (active high) for oscillators that need it. A white dot beside the clock socket indicates pin 1.

The MC6830x emulator is shipped with a 20-MHz CMOS oscillator, packaged separately. You must install an internal clock module when running the emulator performance verification (pv).



Step 5. Select a clock module

Five clock modules are available on the HP Clock Module Kit, which is supplied with the MC6830x emulator. Additional clock module kits can be ordered from Hewlett-Packard for emulator support. To use one of these modules, break off the desired module from the clock module kit and install it in the emulator probe clock socket. Be careful to align pin 1 correctly.



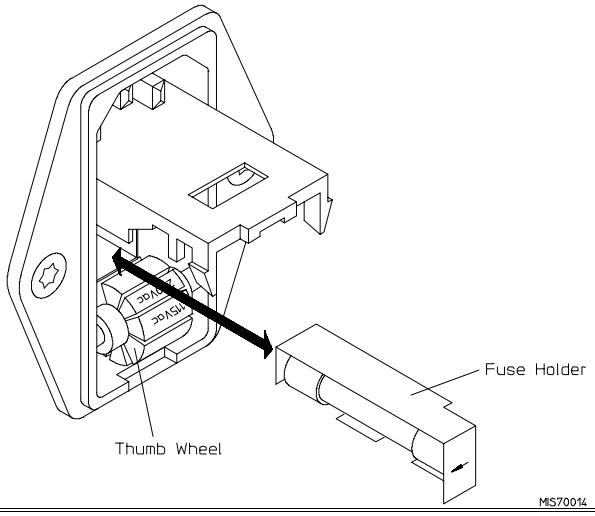
Step 6. Connect the power cord

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 cable appropriate for your country. You should verify that you have the correct power cable for installation.

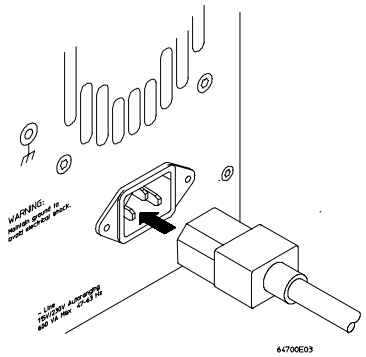
If the cable you received is not appropriate for your electrical power outlet type, contact your Hewlett-Packard sales and service office.



Step 6. Connect the power cord

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 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: Update emulator firmware

If you did not receive your HP 64700 Card Cage and HP 64798 Emulator together (perhaps you are installing your HP 64798 Emulator in a card cage that had a different emulator installed in it), update the emulator firmware according to the steps in Chapter 5, "Installing/Updating Emulator Firmware."

Step 5: 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 Debugger 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.

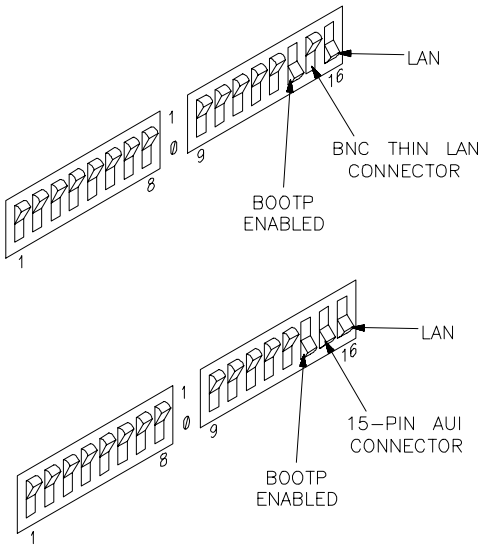
The ipconfig700 command sets the Internet Address (IP address), Gateway Address, and Subnet Mask on the card cage LAN interface. An Internet Address (also known as an IP address) must be configured for the card cage before a network interface connection can be made.

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 and the RS232/422 port on a 64700B 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 1s in the bits which correspond to the network and subnet portions of the Internet address and all 0s 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 OFF power to the emulator, and connect the emulator to your network. This connection can be made by using either the 15-Pin AUI connector or the BNC connector.

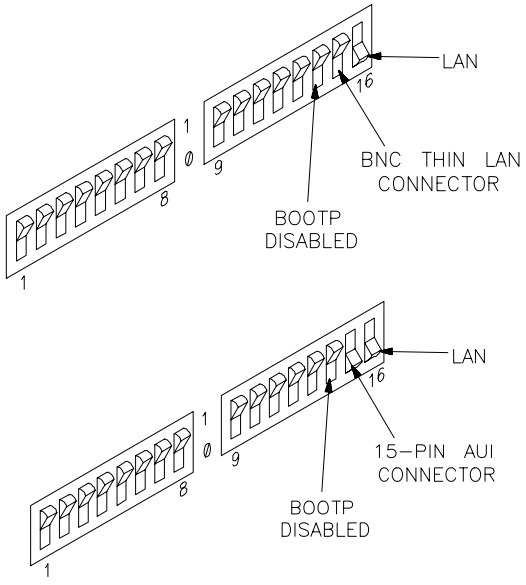
7 Set the rear panel dip switches to indicate the type of connection that is to be made:

Set switch 16 to one (1), indicating that a LAN connection is being made.

Set switch 15 to zero (0) if you are connecting up to the BNC connector, or set switch 15 to one (1) if a 15-pin AUI connection is made.

Set switch 14 to 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, wait a minute. It takes a minute before the emulator can be recognized on the network. Then try the telnet <internet address> command.

If telnet does not make the connection:

- ☐ Make sure you have connected the emulator to the proper power source and that the power light is lit.
- ☐ Make sure 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 cycled (turned OFF and then ON again).
- ☐ Make sure the Internet Address is set up correctly. You must use the RS-232/RS-422 port to verify 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 you see no terminal interface prompt (such as R>, M>, or U>):

- ☐ It's possible 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.

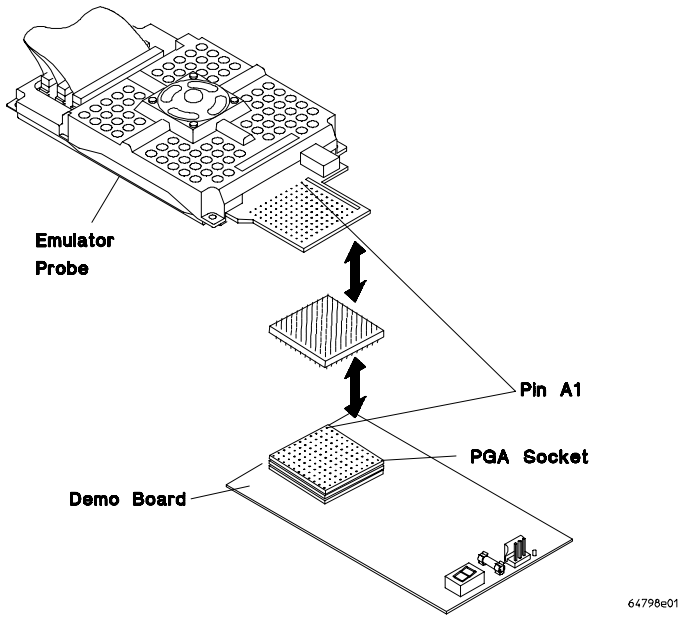
Installation

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.

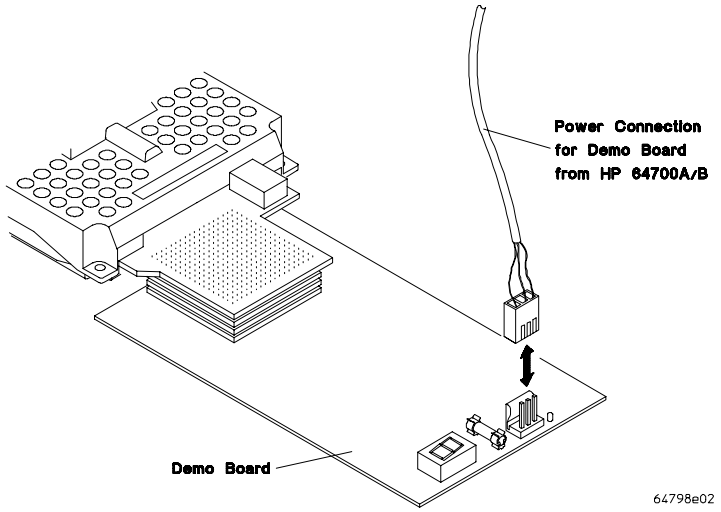
Step 1. Connect the emulator probe to the demo target system

1 With HP 64700 power OFF, connect the emulator probe to the demo target system. Notice the white line labeled 144-TQFP on the probe (white line with no label on MC68LC302 and MC68EN302). It will be above, and parallel with the white line on the demo target system when the emulator probe is connected properly.

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 MC6830x emulator into the demo board socket incorrectly. Be very careful.



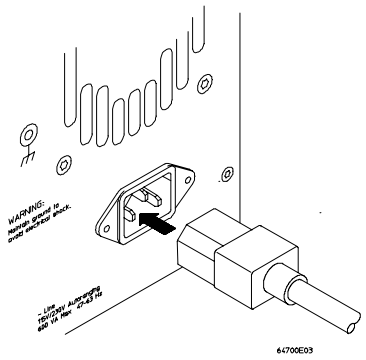
2 Connect the power supply cable from the emulator to the demo target system. The 3-wire cable has one power wire and two ground wires. When attaching the 3-wire cable to the demo target system, make sure the connector ***is aligned properly*** so that all three pins are connected.



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, "Solving Problems."

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.

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 microprocessor signals are electrically isolated from their counterparts on the target system connection. This is done for both electrical and functional reasons. The impact of these circuits is calculated and presented in the emulator specifications beginning on page 145 for the MC68302 and MC68EN302 and on page 189 for the MC68LC302.

Target system design

A target system that is designed around the worst-case specifications of an MC6830x-series 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 microprocessor 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 CLKO signal is always buffered to the target with a CDC392 clock driver. This improves drive capability to the target system and signal quality to the emulator. The ability to change clock drive strength is not supported by the emulator.

Various strobes and control signals ($\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, $\overline{R/W}$, $\overline{IACK7}$, $\overline{CS0}$-$\overline{CS3}$, and $\overline{FC0}$-$\overline{FC2}$ in the MC68302 and MC68EN302, and AS, OE, WEH, WEL, and $\overline{CS0}$-$\overline{CS3}$ in the MC68LC302) are normally gated using transmission gates but can optionally be buffered through 16V8 or 20V8 PALs. Buffering may improve signal quality with certain target systems but at the expense of additional delay and slightly lower V_{OH}. It is recommended that strobes and control signals NOT be buffered unless the particular target system clearly benefits from buffering.

The strobes $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, and $\overline{IACK7}$ (when not used as PB0) in MC68302 and MC68EN302, and AS, $\overline{OE}$, $\overline{WEH}$, and $\overline{WEL}$ in MC68LC302 are grouped together for buffering or un-buffering. R/W, the chip selects, and the function-code lines each can be buffered or unbuffered independent of the strobes. This buffering can be controlled through the configuration dialogs in the graphical interfaces (see the on-line help for details). In the built-in terminal interface, the **cf bufstrbs**, **cf bufrw**, **cf bufcs**, and **cf buffc** commands (and **cf bufwe** in the MC68LC302) control buffering. Different timing specifications are given depending on the selections chosen for these configurations.

Examine the AC specifications of the emulator to evaluate their differences from the microprocessor specifications. Because the emulator does not behave exactly like the microprocessor, you may need to examine signal quality and take appropriate steps to compensate for differences.

DC specifications

Examine the DC specifications of the emulator to evaluate their differences from the microprocessor specifications. The emulator adds input leakage current (I_{IN}) which is included in the I_{IL} and I_{IH} specifications, and additional capacitance which is included in the C_{IN} specifications. The C_{EL} specifications include how much additional capacitance the emulator places on outputs from the microprocessor which go to the target. This can be used in calculating output drive derating factors for all signals except CLKO, which is buffered before it is sent to the target.

When $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, R/$\overline{W}$, $\overline{IACK7}$, $\overline{CS0}$-$\overline{CS3}$, or $\overline{FC0}$-FC2 on the MC68302 and MC68EN302, or $\overline{AS}$, $\overline{OE}$, $\overline{WEH}$, $\overline{WEL}$, and $\overline{CS0}$-$\overline{CS3}$ on the MC68LC302 are configured to be buffered, output drive capability will meet or exceed that of the microprocessor itself except for V_{OH} which is slightly lower.

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.

Clock information

MC6830x processors can operate in either of the following two clock modes:

- Clocked from an oscillator with CMOS levels supplied to EXTAL.
- Clocked from a crystal connected to EXTAL and XTAL.

The emulator supports both of the above clock modes. For flexibility when supporting these clock modes, the emulator allows operation from either an external or internal clock source, as follows:

- External clocks are clocks supplied from an oscillator within the target system.
- Internal clocks are clocks supplied from either an oscillator or crystal plugged into the 14-pin clock socket on top of the emulation probe.

Crystal operation is only supported when using internal clocks because cable capacitance may make external crystal operation unreliable. If you wish to use crystal operation, the clock module kit (included with the emulator) provides three standard frequency crystal modules. If you wish to use a crystal frequency that is not provided in the clock module kit, you can build your own circuit on either User 1 (Module A) or User 2 (Module B) of the clock module kit. These two modules differ only in the land pattern for the surface mount crystal. Choose the module that best fits your crystal.

The following table shows you how to configure the emulator for the supported clock modes.

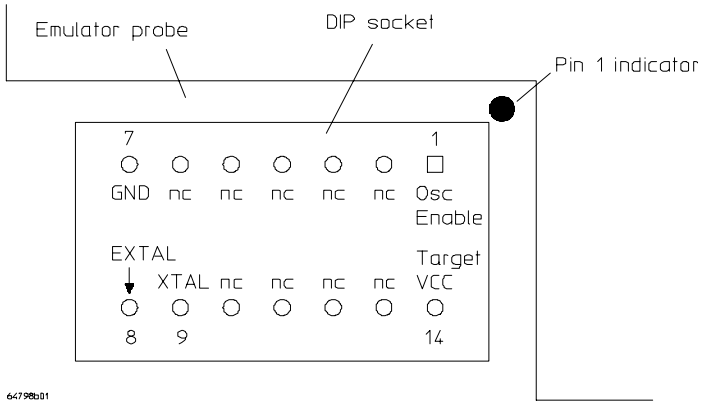
Microprocessor and Clock Mode	Clock source for emulator	
	Internal **cf clk = int**	**External** **cf clk = ext**
MC68302 and MC68EN302 with oscillator	oscillator module: 20 MHz supplied by HP or 8-25 MHz supplied by user	Any oscillator or crystal module. This provides a working internal clock which must be detected during interface initialization.
MC68302 and MC68EN302 with crystal	24 MHz (Module E), or User 1 (Module A), or User 2 (Module B)	Not supported.
MC68LC302 with oscillator	oscillator module: 20 MHz supplied by HP or 0-25 MHz supplied by user	Any oscillator or crystal module. This provides a working internal clock which must be detected during interface initialization.
MC68LC302 with crystal	32.768 kHz (Module C), or 4.194 MHz (Module D), or User 1 (Module A), or User 2 (Module B)	Not supported.

Note

To run performance verification on the emulator, the demo/test board must be attached to the emulator probe instead of the flexible adapter cable, the +5V power cable from the emulator frame to the demo/test board attached, and the emulator internal clock source must contain an oscillator or crystal circuit with an appropriate frequency. If the internal clock source is missing, performance verification will fail.

The clock socket on top of the emulation probe is a 14-pin DIP socket, accessible without removing the probe top cover. You can install a standard 14-pin DIP CMOS oscillator of any frequency that is approporiate for the version of MC6830x processor being emulated. The white dot beside the clock socket indicates pin 1. Pin 1 provides Oscillator Enable (active high) for oscillators that need it. A 20-MHz CMOS oscillator is included with the emulator for use as the internal clock source.

The diagram below shows the pinout of the clock socket on the MC68302 and MC68EN302 emulator probes. When using the 68LC302 emulator, pins 4, 6, 11, and 12 also have connections. Refer to the schematic on page 60 to see their usage.



If you are using an external clock source, you still must have an oscillator or crystal clock module installed. It must be detected during initialization of the graphical user interface. To run PV, you must install an appropriate oscillator or crystal in the clock socket. Performance verfication can only be run in the internal clock mode. A relay is used to switch between external and internal clock sources so no additional delay is added from the target EXTAL signal to the microprocessor. The XTAL signal from the microprocessor is not sent to the target system because external crystal operation is not supported.

The CLKO signal out of the microprocessor is the most important signal to the emulator because all system timing is derived from it. The CLKO signal from the microprocessor is always buffered to the target to isolate any target reflections on the target CLKO signal from the emulator. For proper operation, the CLKO signal must always be available to the emulator. It is possible to turn off CLKO from the microprocessor by setting both bits 14 and 15 of the CLKOCR register (address $0FA) to 1. If CLKO is turned off, the emulator will indicate a *Slow Clock* prompt; a reset must be done to recover from this condition. It is also possible to change

the drive strength to 2/3 or 1/3 by setting these two bits to values other than zero. In this case, the emulator may appear to work but performance will be impaired. Make certain your target program never sets bits 14 and 15 of the CLKOCR register (address $0FA) to anything but zero.

Selecting a clock module

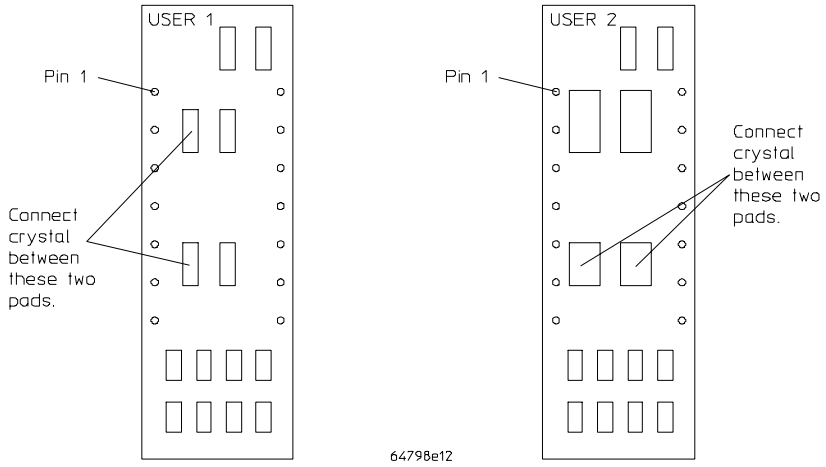
If you can, use the clock circuitry from your target system to clock the emulator. If you must use an internal clock, ensure you have installed the correct clock module in the emulator probe before turning on the emulator.

The emulator is shipped with a 20-MHz CMOS oscillator for use as an internal clock source. You may use your own CMOS oscillator or build up a DIP header containing your own crystal circuit (see the *Motorola MC68302 User's Guide* for details). When using the demo/test board that is supplied with the emulator, if you choose external clock source, the demo/test board will supply a 25-MHz CMOS level clock.

If you wish to install a new clock module, the following guidelines will help you select the one that is correct for your application.

- If the target system uses a CMOS oscillator and you will clock your emulator from that oscillator:
  - Configure emulator for external clock (cf clk=ext).
  - Any clock module can be plugged in on the emulator probe. A valid internal clock is required during initialization of a graphical user interface.
- If the target system uses a 32.768-kHz crystal (MC68LC302 only):
  - Configure emulator for internal clock (cf clk=int).
  - Plug in Module C from the clock module kit (32.768 kHz crystal).
  - MODCLK = 1, driven by emulator.
- If the target system uses a 4.194-MHz crystal (MC68LC302 only):
  - Configure emulator for internal clock (cf clk=int).
  - Plug in Module D from the clock module kit (4.194 MHz crystal).
  - MODCLK = 0, driven by emulator.
- If the target system uses a 24-MHz crystal (MC68302 or MC68EN302):
  - Configure emulator for internal clock (cf clk=int).
  - Plug in Module E, 24-MHz crystal module.

- If you need to design your own clock module:
  - Use Module A (User 1) or Module B (User 2). The modules are the same except for the land pattern for the crystal. Be careful to ensure that your crystal connects properly to the land pattern of the module.
  - Design your clock module using the module schematics on the next page, and the USER1/USER2 diagrams below to help you place your components.

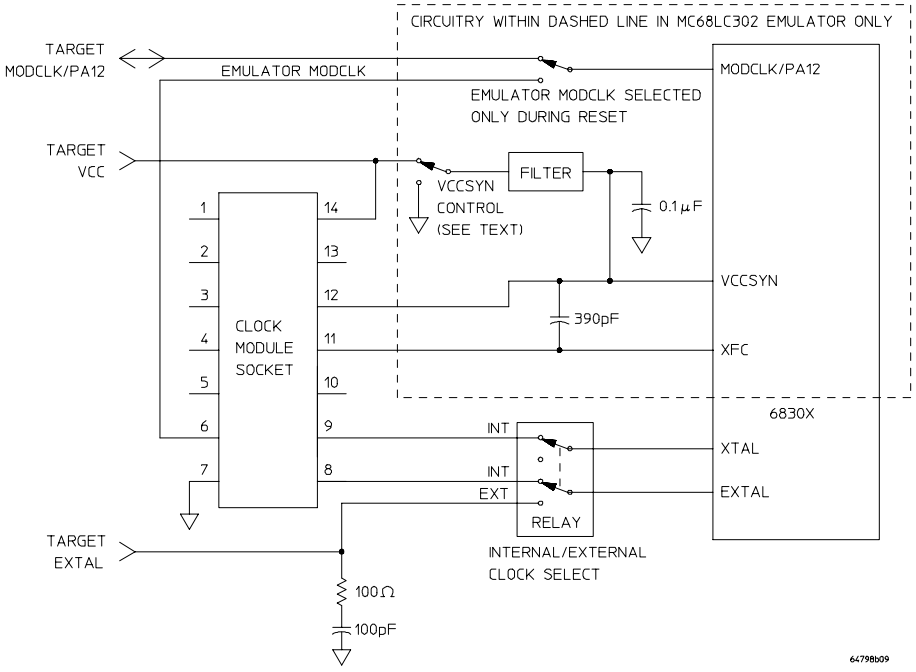


Notes when designing a clock module for the MC68LC302 emulator:

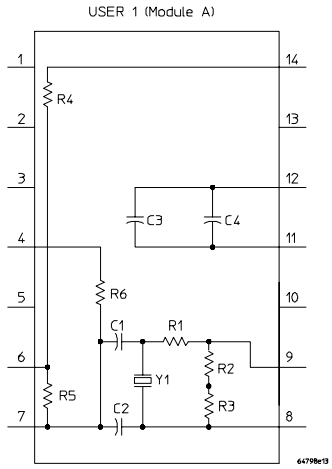
- To make the emulator pull MODCLK low during reset, load a 0 to 215-ohm resistor in R5 (R15 for User 2).
- To make the emulator pull MODCLK high during reset, load a 0 to 215-ohm resistor in R4 (R14 for User 2).
- To make the emulator drive VCCSYN high (to enable the PLL), load a 0 to 215-ohm resistor in R6 (R16 for User 2). If R6 (R16) is not loaded, VCCSYN will be connected to GND and the PLL will not be enabled.

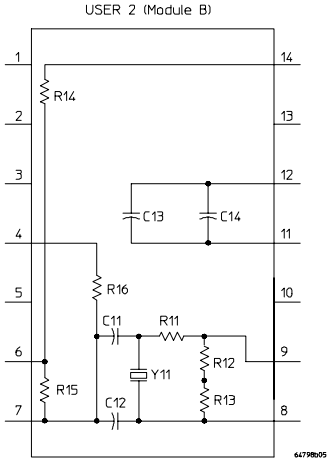
Clock module circuits

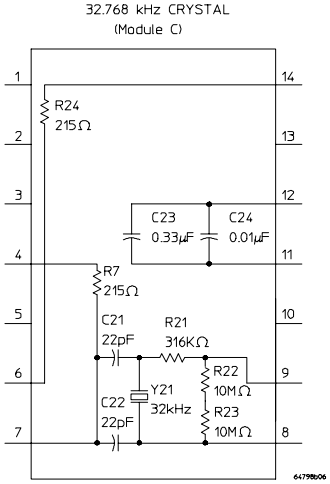
Emulation Probe circuitry associated with the installed clock module:

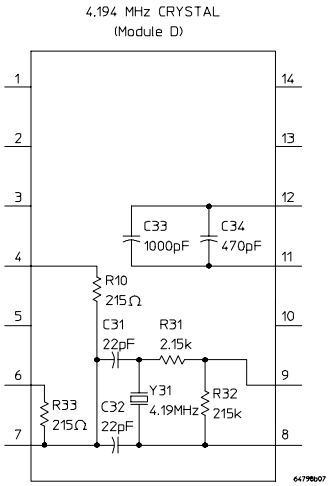


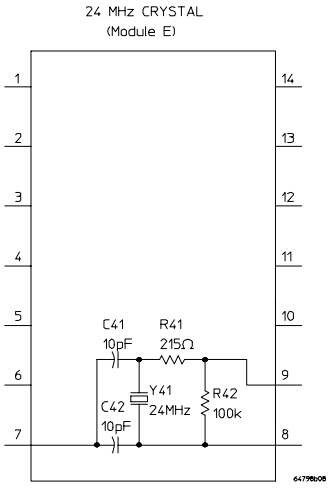
Individual clock modules:











Clock module circuit details

Resistor R6 in the clock module completes a circuit that allows the emulator to obtain target VCC through the VCCSYN control switch. This switch supplies VCCSYN to the microprocessor where it enables the PLL in the microprocessor. Load a 0 to 100-ohm resistor for R6. Without resistor R6, the VCCSYN input of the microprocessor will be connected to ground and the PLL will be disabled.

The filter is an inductor-capacitor network that provides fine filtering of Target VCC to the microprocessor.

During reset, the emulator drives MODCLK to the microprocessor, and the value on the MODCLK/PA12 pin from the target system is ignored. During this time, the value supplied from clock module pin 6 either pulls MODCLK high or low, depending on whether you installed R4 or R5 in the clock module. During all non-reset times, the target system controls the signal on the MODCLK/PA12 pin.

The schematics show that target VCC is unused. It provides operating power to oscillators when they are plugged into the clock socket.

For an external oscillator clock source, the emulator terminates the EXTAL signal with 100 ohms in series with 100 picofarads.

Testing your custom crystal clock circuit

To determine if your custom crystal clock circuit operates properly, first make sure the emulator power is turned off and the emulator is plugged into the demo/test board. Then install your custom crystal header into the clock socket on the emulator and power up the emulator. In the built-in terminal interface, the **cf clk** command selects either an internal or an external clock source. The default from power-up is internal. The emulator contains logic which can determine the frequency of internal or external clock sources by measuring the CLKO signal out of the microprocessor. Use the **info clock** terminal interface command to display the clock frequency and source. The frequency should be near what you expect for the crystal you are using. If the frequency is 0.000 MHz or the wrong value, the crystal circuit is not working properly. If the frequency is correct, run performance verification to ensure the emulator will function with this internal clock source.

Connecting the emulator to an MC68302 target system

This emulator supports connections to 132-pin PQFP and 144-pin TQFP package types for the MC68302. It also supports connections to the PGA pinout of older MC68302 microprocessors. Before connecting the emulator, a 132-pin PQFP or 144-pin TQFP "dummy part" (a mechanical sample with no internal connections) must be soldered onto the target system in place of the microprocessor. This is necessary because an MC68302 microprocessor has no facility to tri-state all of its signals. It is best to solder the dummy part onto the target system using automated surface mounting equipment to give more reliable probing. Hand soldering may result in solder wicking up the leads, which can prevent the probe adapter cable assembly from making good contact.

If connecting the emulator to a target system PGA connector, remove the microprocessor from the target system and install the transition socket (listed in Chapter 7).

The emulator probe pod is normally connected to the target system using one of the available probe assemblies with their flexible adapter cables (see the following pages for connection details). The specifications given in this manual (page 145, and page 189) include use of the appropriate adapter cable. The probe assembly kits, which include dummy part kits, are listed in Chapter 7, Parts List (page 131).

The keep-out area is shown in your probing hardware manual. This shows the space required on your target system to allow connection of the probing hardware.

See Also: The documentation included in the probe kits for detailed instructions.

The 144-pin and 132-pin adapter cables can be installed in one of four orientations (each at right angles to the others). This allows flexibility in attaching the probe when target system components interfere. Select the orientation (shown on the next pages) that best suits your target system, and note the position of Pin 1 on the microprocessor (dummy part) on your target board.

There are two labels with color coding and bar coding on the adapter cables; use them to ensure correct orientation when the probe adapter is connected to the emulator. Note the color or count the bars on the edge of the probe that is placed over the Pin 1 side of the target microprocessor chip. (For example, Pin 1 of the microprocessor chip may be along the side that is color coded yellow, or along the side that has three bars.) There is a corresponding edge on the PGA end of the

adapter cable; it has the same color code and bar code. For example, the PGA connector of the MC68302 emulator probe has a white bar and corresponding notation "144-TQFP" and a second white bar and corresponding notation "132-PQFP". These bars indicate the side where Pin 1 occurs on the emulator probe 13x13 PGA connector for a 144-pin TQFP target and for a 132-pin PQFP target. If you are making connections to an MC68302 dummy part, connect the "144-TQFP" side of the emulator probe 13x13 PGA connector into the PGA end of the 144-pin TQFP adapter cable, or connect the "132-PQFP" side of the emulator probe 13x13 PGA into the PGA end of the 132-pin PQFP adapter cable. Note that the 132-pin PQFP adapter cable must include a "transition board" which converts the 12x12 PGA at the end of the PQFP adapter cable to the 13x13 PGA used by the emulator probe. This "transition board" is included in the E3437A kit.

If you use the PGA transition socket to connect directly into the PGA connector on your target system, the position of pin 1 on your target system pinout governs PGA connector orientation.

Caution

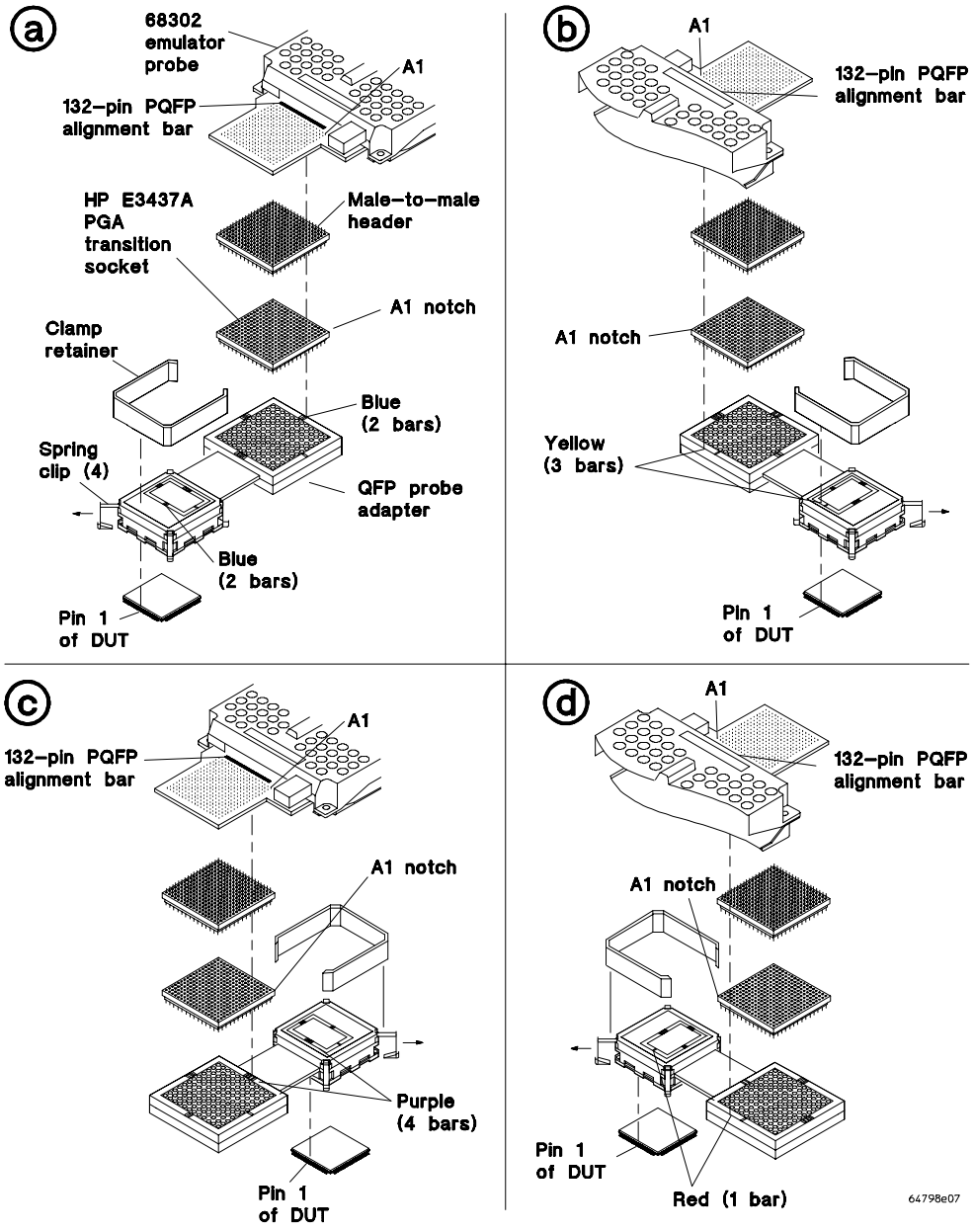
Equipment damage. The connections between the emulator probe, flexible probe adapter cable, and microprocessor (dummy part) on the target board are delicate and must be done with care. Refer to the Operating Note supplied with the flexible adapter cable for specific instructions when making the connection.

Connecting the emulator to an MC68EN302 target system

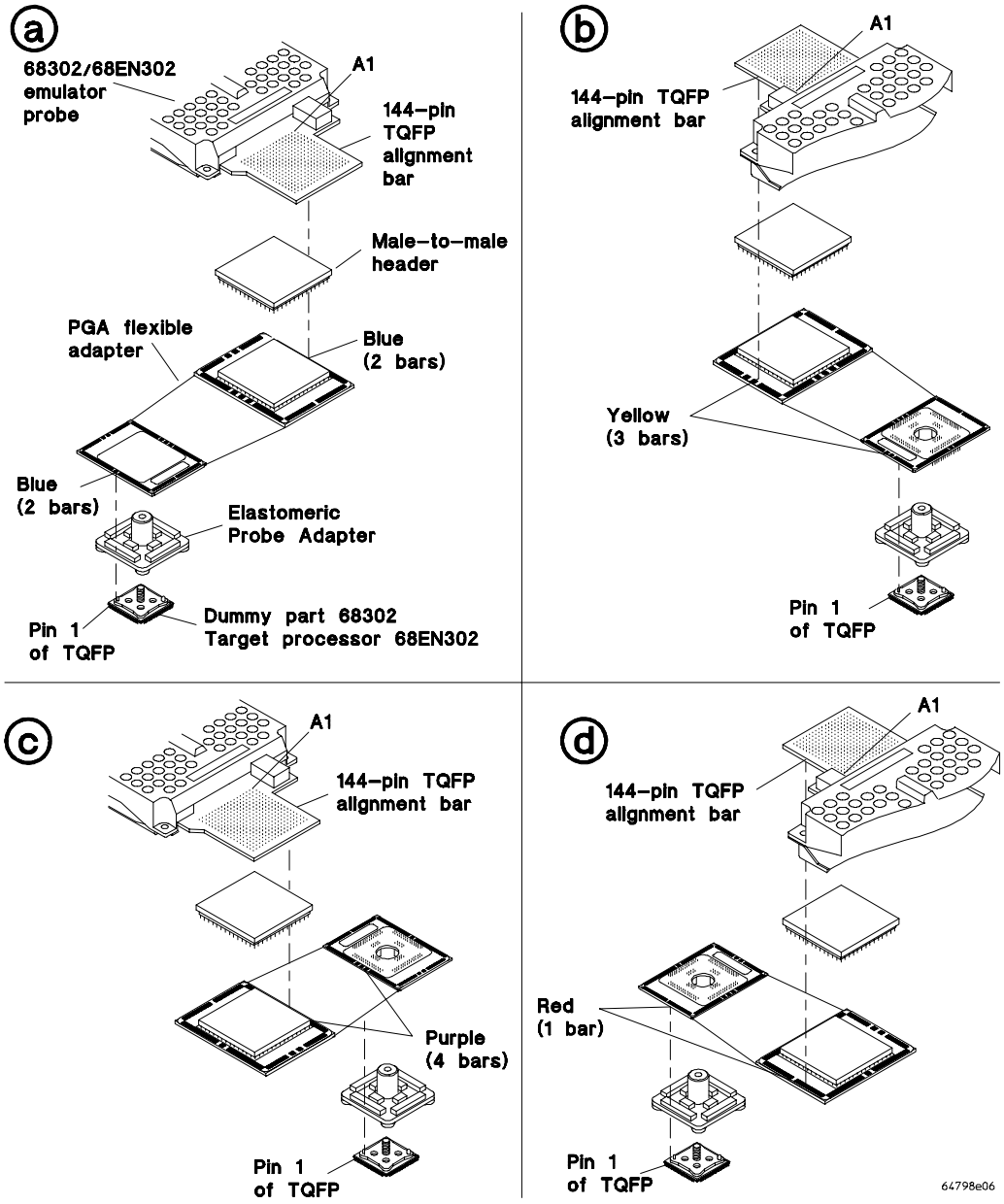
This emulator supports connections to a 144-pin TQFP package type. An MC68EN302 emulator connects directly to the target microprocessor, not to a dummy part. The MC68EN302 can tri-state its signals.

Connect the emulator probe pod to the target system using the 144-pin probe assembly shown on the following pages. The specifications given in this manual (page 145) include use of the adapter cable. The probe adapter and flexible adapter cable are listed in Chapter 7, Parts List.

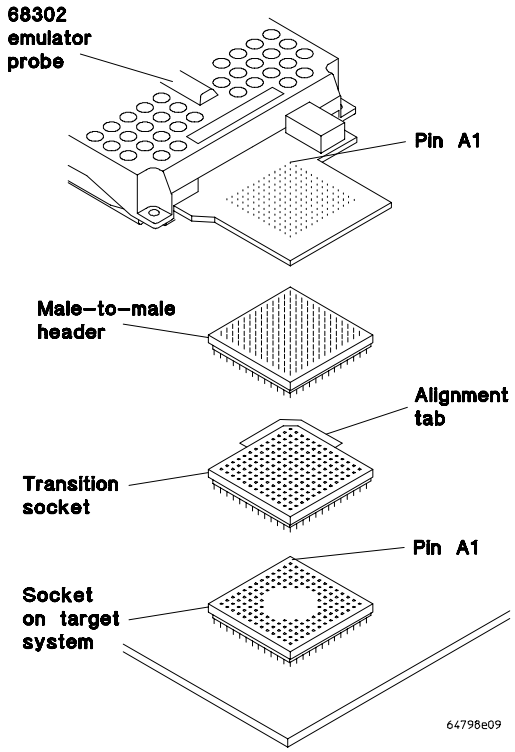
Observe the instructions and precautions for installing the 144-pin connector hardware described in the paragraph titled, "Connecting the emulator to an MC68302 target system," above.



E3437A Probing of 132-pin PQFP



E5336A/E5338A Probing of 144-pin TQFP

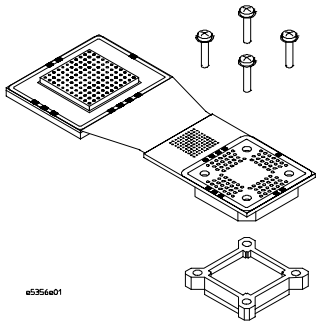


E5367A Probing of target system PGA

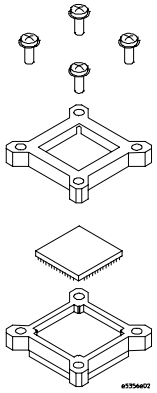
Connecting the emulator to an MC68LC302 target system

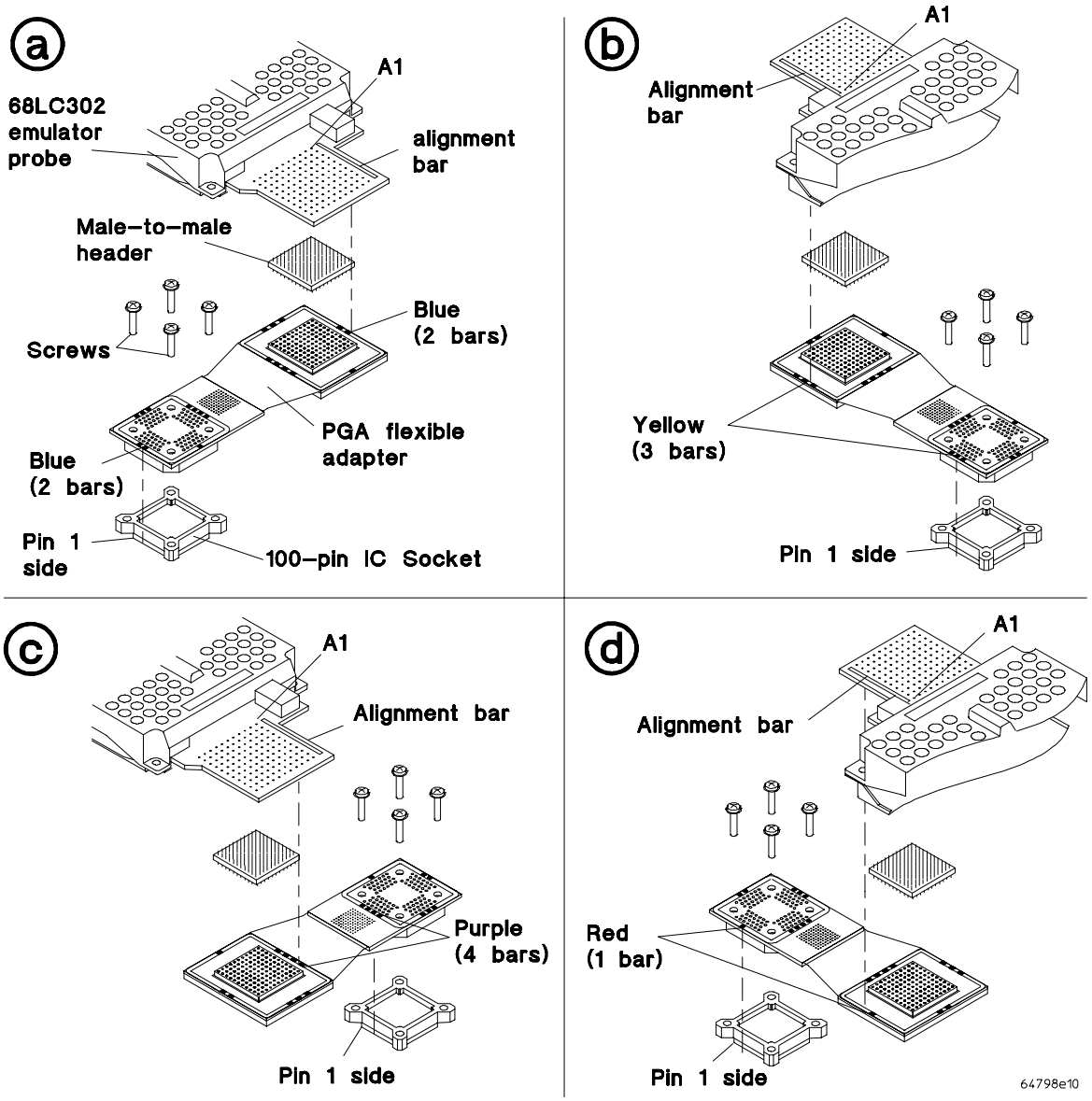
First, remove the MC68LC302 microprocessor from your target system. Solder the 100-pin IC socket in its place. The 100-pin IC socket is included in the HP E5356A probing kit. There is a triangle molded inside the IC socket. For ease of reference, you can install the IC socket so that the triangle points to the location of pin 1 on your target board.

To connect the HP 64798F emulator probe to your target board, see the illustration titled, "E5356A probing with MC68LC302 emulator." Select the orientation that allows the best probing connection to the target board. Attach the PGA flexible adapter to the IC socket with the four longer screws.



If you wish to install the MC68LC302 microprocessor on your target board, install it in the IC socket in place of the flexible cable. Hold it in place with the plastic cover and the four short screws. See below.





E5356A Probing with MC68LC302 emulator

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 115.

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 microprocessor 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. Do not attempt to make measurements at the microprocessor chip on the emulator by removing the cover. There may be logic circuitry between the microprocessor chip on the emulator and the target connector. Make measurements on the target system as near as possible to the target microprocessor or dummy part. Alternatively, measurements can be made on the PGA target connector of the emulator probe where it attaches to the flexible adapter cable. 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 13x13 PGA target connector on the MC68302 and MC68EN302, or the 11 x 11 PGA on the MC68LC302) is a good ground that is easy to connect to when probing the target connector. Refer to the pin-out for the PGA target connector on page 178 (for the MC68302), page 183 (for the MC68EN302), or page 217 (for the MC68LC302) to find specific signals on the target connector.

The emulator specifications listed in this manual include the use of the 144-pin TQFP, 132-pin PQFP, or the 100-pin TQFP flexible adapter cable. The flexible adapter cable capacitance has already been included in the emulator's Input Capacitance (C_{IN}) and Emulator Load Capacitance (C_{EL}) specifications.

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 clk=int (for internal clock crystal or oscillator)
   OR
   cf clk=ext (for external clock oscillator only)
   cf berr=en
   cf ti=en

4. Set up the emulation-bus analyzer to capture all MC6830x 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, "Resetting into the background 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 MC6830x often or for long periods of time, the emulator can display the "g>" prompt or alternate between "g>" and "U>".

All other prompts usually indicate a problem. For example, the "b>" prompt indicates that there currently are no bus cycles. This may occur if the MC6830x has entered a low power mode. Even the "g>" prompt 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 CLKO 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.

To capture alternate bus master cycles by the analyzer, first execute the command **cf trcdma=en**.

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.

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 DTACK internally for the current bus cycle. If the DTACK 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.

An MC6830x bus cycle starts with the assertion of AS, and possibly a chip select. The AS signal stays low throughout the cycle and is deasserted between cycles. The end of the cycle occurs when the processor samples DTACK or BERR. A typical system may sample AS on the rising clock edge and then generate a DTACK signal an integral number of clocks later. Wait states are added to a cycle by delaying when the DTACK 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 DTACK 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 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 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 DTACK is present for the duration required to terminate a cycle. If DTACK is not asserted at all, it could be an indication that the target system missed the assertion of 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.

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 BG in response to a target system bus request BR. The processor does not grant the bus until it is idle so bus ownership changes as soon as the BG signal is asserted. The alternate master asserts BGACK to claim the bus before deasserting BR.

If the bus is granted by the processor but is not being returned, check the bus arbitration signals BR, BG, and BGACK. If the bus is never released, the alternate bus master may be stuck in the middle of a cycle. Check the cycle strobes AS, DTACK, and BERR. These strobes do not have to be asserted during alternate master accesses, but if AS is shown to the processor, it will generate memory control signals and may even provide DTACK 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  68302 Mnemonic            stat,H   count,R
  -----   ------  ------------------------ ------ ----------
      0   000000  $0000      supr prgm rd   f79d   ----------
      1   000002  $0000      supr prgm rd   f79d   0.20uS
      2   000004  $00BA      supr prgm rd   f79d   0.20uS
      3   000006  $DADD      supr prgm rd   f79d   0.20uS
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 -d -17..0

```
h>tl -d -17..0
   Line   addr,H  68302 Mnemonic
  -----   ------  ---------------------------------
    -17   001000   NOP
    -16   001002   Illegal Instruction: $4AFC     <- Illegal instruction
    -15   001004   NOP
    -14   001ffe   $1002     supr data wr         <- illegal instr stack
    -13   001ffa   $2700     supr data wr
    -12   001ffc   $0000     supr data wr
    -11   000010   $0011     supr data rd         <- odd vector
    -10   000012   $1111     supr data rd
     -9   001ff8   $0010     supr data wr         <- address error stack
     -8   001ff4   $2700     supr data wr
     -7   001ff6   $0000     supr data wr
     -6   001ff2   $4AFC     supr data wr
     -5   001ff0   $1111     supr data wr
     -4   001fec   $4AFE     supr data wr
     -3   001fee   $0011     supr data wr
     -2   00000c   $0033     supr data rd         <- odd vector
     -1   00000e   $3333     supr data rd
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 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 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.

As a last resort, experiment with buffering the strobes and control signals from the emulator. By default, AS, UDS, LDS, R/W, IACK7, CS0-CS0, and FC0-FC2 are unbuffered. To buffer AS, UDS, LDS, and IACK7, for example, use command **cf bufstrbs=en**. To buffer all strobes and control signals, use the following commands:

MC68302/MC68EN302	**MC68LC302**
cf bufstrbs=en	**cf bufstrbs=en**
cf bufrw=en	**cf bufcs=en**
cf bufcs=en	**cf bufwe=en**
cf buffc=en	

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 command:

cf clk=int

4 Map memory with the following command:

map 0..0fff eram dp

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

5 Power on the target system.

This is needed at this point because the internal clock must be present to access dual-port memory and the internal clock is powered from the target system.

6 Check the prompt by pressing <RETURN>.

mo -aw -dw
m 0=0,0f00,0,100
m 100=60fe,4e71

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

7 Set up a trace to capture all microprocessor cycles, as follows:

```
tck -u
tg any
tsto any
tp c
t
```

8 Power off the target system.

9 Execute **r rst**.

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

10 Power on the target system.

11 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, "Resetting into the background monitor." If you suspect problems, return to "Verifying System Operation" in the previous paragraphs. Keep in mind that 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 microprocessor.

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 several 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.

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

5 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

6 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 several 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 background monitor, by default, does not show microprocessor cycles to your target system. Therefore, the background monitor is transparent to your 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 commands:

   cf berr=en
   cf ti=en
   cf rv=2000,1000

   The **cf rv=2000,1000** command tells the emulator to use 2000H as the initial SSP and 1000H as the initial PC. If the command **cf rv=auto** had been used, the emulator would attempt to read the reset vector from target memory using foreground cycles to get the initial values for the SSP and the PC. You can change the values used in the **cf rv** command to the initial values you desire or simply change the SSP and PC after entering the background monitor.

3. Set up a trace to capture all MC6830x cycles, including background monitor cycles, by entering the following commands:

   tck -ub
   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 -dw
m 01adc=1234

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 01adc
  0001adc  ffff
```

If the target memory does not respond to a bus cycle and the MC6830x hardware watchdog timer causes a bus error, or the target system asserts BERR, the following error messages will be reported:

```
M>m badad0=1234
!ERROR 155! Target memory access; bus error
!ERROR 700! Target memory access failed
```

If your target memory does not respond to a bus cycle and the MC6830x hardware watchdog timer is turned off, the monitor will force termination of the cycle and report the following error messages:

```
M>m 0badad0=1234
!STATUS 164! Emulator terminated hung bus cycle: 0badad0@sd write
!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. Determine how you want the emulator to obtain the initial SSP and PC.

   The easiest way is to let the emulator read the reset vector directly from your target memory by using the **cf rv=auto** command. This command tells the emulator to read the target reset vector whenever the background monitor is entered from reset. Alternatively, you can use the **cf rv** command to manually specify initial SSP and PC values the emulator should use when the background monitor is entered from reset. Use this approach if your target reset vector is not always accessible by the emulator. A third approach is to simply set the SSP and PC after entering the background monitor.

2. Reset into the background monitor.

   rst -m

3. Load a program, if necessary.

4. Initialize the stack pointer (SSP) and program counter (PC) if not done earlier using the **cf rv** command.

   reg ssp=<initial supervisor stack pointer>
   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.

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

tg addr=<target program starting address>
t

6 Run the program with the command:

r

7 Verify 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. Use the following command:

b

The emulator uses an interrupt level-7 to break into the background monitor. One word of stacking occurs to the supervisor stack in target memory; from then on, the interrupt-acknowledge cycle and subsequent stack operations are hidden from the target. Target program execution stops and background monitor cycles are executed. The background monitor may access the target memory by performing foreground cycles during its operation.

While the emulator is executing in the background monitor, no target interrupts are serviced; the interrupt level-7 signal(s) from the target system are blocked. Since level-7 interrupts are shared between the emulator and the target, the emulator will remember any "dedicated edge mode" level-7 target interrupt that occurs while in the monitor and apply it to the MC6830x when exiting the background monitor. While in the background monitor if the interrupt signals are configured for "normal" mode, $\overline{\text{IPL0}}$-$\overline{\text{IPL2}}$ are blocked; if configured for "dedicated" mode, only

$\overline{\text{IPL2}}$ ($\overline{\text{IRQ7}}$) is blocked. Note that command **cf ti=dis** turns off target interrupts, causing IPL0-IPL2 to always be blocked.

Note that the **cf ti=dis** command only blocks external interrupts. Interrupts from on-chip counters, timers, etc., are not blocked. While in the background monitor, however, the interrupt mask is set at level 7 in the status register.

If the target is asserting level-7 on the interrupt line(s), the emulator will be unable to break into the background monitor. The emulator will display the following messages:

```
U>b
!STATUS 156! Target may be asserting INTR 7
!ERROR 608! Unable to break
```

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.

Software breakpoint entry into the background monitor

The background monitor can also be entered via a software breakpoint. If breakpoints are enabled, the emulator will respond to one of the sixteen TRAP instructions. To specify which TRAP instruction should be used by the emulator, use the following command:

cf swtp=<trap number>

Software breakpoints are enabled by the following command.

bc -e bp

Set breakpoints only on the initial word of an instruction; 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 TRAP 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 the specified TRAP instruction is executed, the emulator transitions into the background monitor.

If your program uses the same TRAP instruction as specified for software breakpoints, your program may be inadvertently stopped and the background monitor entered with the following message:

!ASYNC_STAT 605! Undefined software breakpoint: 0001006@sp

If this occurs and you are not using software breakpoints, disable software breakpoints using the **bc -d bp** command. If you are using software breakpoints, change which TRAP instruction the emulator should use for software breakpoints; use the **cf swtp** command.

Stepping with the background monitor

The last feature of the background monitor that needs to be evaluated is the single-stepping facility. The emulator asserts an interrupt level-7 at just the right time after transitioning to foreground cycles (running user code) to allow exactly one instruction to be fetched and executed before returning to the background monitor.

b
tck -u
tg any
tsto any
t
s

A typical trace of a single step is shown below. This trace shows only foreground cycles, including foreground cycles generated by monitor activity before and after the single step itself. Note the time count. It shows "long" times between foreground cycles indicating monitor operations that set up the supervisor stack before executing an RTE instruction to transition to user code, and monitor reads after the stepping is complete.

```
M>TL -d 0
  Line  addr,H  68302 Mnemonic             count,R
 -----  ------ -------------------------- -------
     0  001ffe  $1002      supr data wr        ---  <- stack setup
     1  001ffc  $0000      supr data wr    6.520 uS     by monitor
     2  001ffa  $2700      supr data wr    6.480 uS
     3  001ffa  $2700      supr data rd    28.60 uS <- RTE executed
     4  001ffc  $0000      supr data rd    0.200 uS     to leave mon.
     5  001ffe  $1002      supr data rd    0.200 uS
     6  001002 NOP                         0.520 uS <- executed code
     7  001004 NOP                         0.480 uS <- prefetch code
     8  001006 BRA.B     $00001006         0.520 uS <- prefetch code
     9  001ffe  $1004      supr data wr    0.480 uS <- entering mon.
    10  001ffe  $1004      supr data rd    43.12 uS <- monitor reads
    11  001002 NOP                         24.37 ms
    12  001002 NOP                         5.263 ms
M>
```

To see both foreground and background (monitor) cycles in a trace, you can do the following:

b
tck -ub
tg addr=<address _to_single_step>
tsto any
tp c
t
s 1 <address_to_single_step>
th
tl -d -10..10

The resulting trace will look like the following:

```
M>tl -d -10..10
  Line   addr,H   68302 Mnemonic
 -----   -------  ------------------------------
   -10   001886   BRA.W    $0000181C        <- background monitor cycle
    -9   000458    $0000 mon supr data  rd
    -8   00045a    $1FFA mon supr data  rd
    -7   001888    $FF94 mon supr prgm  rd
    -6   00181c   NOP
    -5   00181e   RTE                      <- last monitor instruction
    -4   001820   NOP                      <- background prefetch cyc.
    -3   001ffa    $2700     supr data  rd  <- first foreground cycle
    -2   001ffc    $0000     supr data  rd
    -1   001ffe    $1004     supr data  rd
     0   001004   NOP                      <- stepped instruction
     1   001006   BRA.B    $00001006       <- foreground prefetch cyc.
     2   001008   NOP
     3   001ffe    $1006     supr data  wr  <- foreground stack write
     4   fffffe    $0017 mon cpu space  rd IAC  <- background int-ack.
     5   001ffa    $2700 mon supr data  wr  <- background stack writes
     6   001ffc    $0000 mon supr data  wr
     7   00005c    $0000 mon supr data  rd  <- background vector fetch
     8   00005e    $0A8A mon supr data  rd
     9   000a8a   MOVE.W   #$0002,$00000412 <- background monitor exec.
    10   000a8c    $0002 mon supr prgm  rd
M>
```

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

Driving background cycles to the target

Some target systems have "keep-alive" circuitry that looks for bus activity and will reinitialize the system if there is no bus activity. To deal with this, you can choose for the background monitor cycles to be visible to your target system by using the command **cf dbc=en**. The background monitor cycles will appear in address range 0H through 1FFFH and will always be read cycles; all microprocessor write cycles will be turned into read cycles to the target. The R/W signal to the target is forced high to change a write into a read cycle, however no attempt is made to alter UDS and LDS, which will still have write-cycle timing. Any data read from the target is ignored. If your target modifies hardware or data upon read cycles between addresses 0H through 1FFFH, do not drive background cycles to your target.

Freezing microprocessor peripherals during background cycles

By default, the emulator will assert $\overline{\text{FRZ}}$ to the microprocessor while in the background monitor. Note that the timers, and in particular the watchdog timer, must also be programmed in the System Control Register (SCR) to suspend counting when FRZ is asserted. FRZ is not available to the target system for the MC68EN302. It is also not available for the MC68LC302 because of the 100-pin TQFP package limitation. It is available to the emulation processor because the emulation processor is in a PGA package.

If you do not want to freeze microprocessor peripherals during background cycles, use command **cf bkgfrz=dis**. Note, however, that any interrupts generated by peripherals will be ignored while in the background monitor. Target assertions of FRZ will always be seen by the MC68302 microprocessor (but not by the MC68LC302 or MC68EN302 microprocessors).

Emulation memory

The last feature of the emulator that you need to integrate is emulation memory. Emulation memory is intended to overlay ROM memory in the target system. Emulation memory allows you to quickly load changes to target programs into a system. The emulator includes 56K of built-in, fully dual-ported, emulation memory. This dual-port memory is organized into seven blocks with 8K bytes each.

Expanding emulation memory

Emulation memory can be expanded by installing optional SIMM-type, non-dual-ported, emulation memory. Either one or two SIMMs can be installed in the emulator probe; if only one SIMM is installed, it may be placed in either SIMM socket. Three different sizes of SIMMs are supported: 256 Kbytes, 1 Mbyte, and 4 Mbytes. Each SIMM is divided into four equal-sized blocks of memory by the emulator. You may install different sized SIMMs in the two SIMM sockets. The emulator will automatically optimize allocation of the different SIMM block sizes when mapping emulation memory.

Mapping emulation memory

When SIMMs are used to expand emulation memory, the seven blocks of dual-port memory are still available, however you may have to specify the **dp** attribute to use them. Without the **dp** attribute, the emulator will allocate emulation memory from the SIMMs first. If only one SIMM is installed, dual-port memory will be automatically allocated after the four blocks of SIMM memory have been allocated. The following two lines show how to allocate one block of dual-port memory before using all of the SIMM memory blocks.

map 1000..1fff eram dp :use a block of dual-port memory for this address
map 2000..0fff eram

The maximum number of blocks of emulation memory that can be mapped is eight with one or two SIMMs installed, and seven with no SIMMs installed. Even though there may be eleven or fifteen blocks of memory available on your system (seven dual-port plus four or eight SIMM), only eight can be mapped at one time. If two SIMMs are installed, mapping 8 Kbytes of dual-ported emulation memory will remove one block (64 Kbytes, 256 Kbytes, or 1 Mbyte, depending on SIMM size) of SIMM memory from the emulation memory available. Mapping 56 Kbytes

as dual-ported will allocate seven blocks of built-in emulation memory and leave only one block of SIMM memory that can still be mapped. With one SIMM installed, you can map up to 32 Kbytes (4 blocks) of dual-ported emulation memory without using any SIMM memory.

Accessing emulation memory

To modify or display emulation memory, the emulator will use the most transparent method available for the type of emulation memory being accessed. For built-in, dual-ported memory, those accesses are completely transparent to the microprocessor; in fact, the microprocessor can even be reset. All that is required is for AS to be occasionally negated (high) and CLKO be active. To access SIMM emulation memory, the emulator will use the background monitor in much the same fashion as when accessing target memory.

Selecting 8-bit or 16-bit emulation memory

Emulation memory can appear to an MC6830x target system as either 8-bit or 16-bit memory. For the MC68302 and MC68LC302, the width of the emulation memory is automatically set by the target system BUSW signal. The MC68EN302 can dynamically vary the bus width, depending on the programming of its four chip selects. To enable emulation memory to respond with the correct bus width for MC68EN302, you must program the emulation copy of the chip select extension register (emcserx). For more information, refer to the following section titled, "Function of Emulation Copy of Processor Registers."

Function of Emulation Copy of Processor Registers

The values in the SIM register group are maintained inside the 6830x processor. The 6830x emulator also maintains an emulation copy of these registers. This emulation copy of the SIM registers (called the EMSIM set) have corresponding registers for each register in the SIM. For example, the EMSIM set has an emmobar register which is an emulation copy of the mobar register in the SIM.

The registers in the EMSIM set are part of the emulation configuration file. Registers in the EMSIM set should be programmed with the "after initialization code" or destination values of the SIM registers. The default values of these registers are the same as the reset values of the SIM registers.

There are two purposes for the emulation copy of the SIM registers: configuring the emulator, and user convenience.

Configuring the emulator

The emulation copy of the SIM registers allows the emulator to be configured or programmed to operate with the target 6830x processor after it has been configured in a unique way by your initialization code. For example, some processor pins have multiple functions that can be selected by user programming of the SIM registers. Some processor pins might be programmed to carry either general purpose I/O or control signals. Also, in the case of the MC68EN302, the chip selects can be programmed for either 8-bit or 16-bit operation independently. The emulator needs to know the programming of multiple function pins in order to respond to them or drive them correctly. The emulator gets much of the information it needs from the EMSIM register set.

The MC68EN302 has a unique need for the EMSIM register set. Emulation or overly memory will respond or emulate either 8-bit wide or 16-bit wide memory, depending upon the programming of bit EN8 in the four emcserx registers. In order for emulation memory to operate properly for the MC68EN302, the EMSIM registers must be programmed correctly and not left at the default values.

User Convenience

If the emulator is currently in the reset state and you wish to download a program to target memory or emulation memory, the 6830x processor must transition from the reset state to the monitor to perform this transfer. After being released from reset, the SIM registers hold the reset values, which are probably not the correct values to commumicate with memory. This is particularly true if chip select pins are used to interface with memory. To overcome the need to set up all of the SIM registers manually, or to run some initialization code before performing this download, the emulator will load the EMSIM values into the SIM of the target 6830x processor when the monitor is entered from reset. This prepares the 6830x to properly communicate with memory. Note that the only time that the EMSIM registers are automatically loaded into the SIM of the target 6830x is when the

monitor is entered directly from reset. If the emulator is commanded to run your target program from reset, the EMSIM registers are not loaded.

Completing emulation memory cycles with target DTACK

Memory cycles are complete when the microprocessor detects the assertion of either the DTACK signal or the BERR signal. By default, the emulator supplies a "0 wait" DTACK to the microprocessor for all background cycles and foreground cycles to emulation memory; the target must supply DTACK for all other cycles. If a microprocessor chip select is programmed to supply DTACK, then the processor will supply the DTACK internally instead of the emulator or target system supplying the DTACK.

After reset, microprocessor chip-select #0 will supply a "6 wait state" DTACK for read or write cycles to addresses 0H through 1FFFH with function code "supervisor program". This means that the background monitor will use the "6 wait state" DTACK supplied by the microprocessor when fetching its opcodes, but use the "0 wait state" DTACK supplied by the emulator for all other cycles. The same will apply to a target program executing in supervisor state, out of emulation memory, between addresses 0H and 1FFFH.

If you prefer that certain emulation memory cycles be completed by a target DTACK, use the **dti** (target DTACK interlock) attribute when mapping that range of emulation memory. Individual blocks of emulation memory can be mapped with or without the **dti** attribute. Use command **cf dtack=maplock** to make sure that the emulator DTACK configuration is set to enable interlocking target DTACK with emulation memory; otherwise the **dti** attribute will be ignored.

```
cf dtack=maplock
map 1000..1fff eram dp
map 2000..2fff eram dti,dp
map 3000..0ffff eram
map 10000..1ffff eram dti
map other tram
```

The default emulator DTACK configuration is **cf dtack=maplock** and the **dti** attribute of the map command controls whether a given block of emulation memory is interlocked with the target DTACK signal or not. If you want all

processor cycles, including both emulation memory and target memory cycles, which are not completed by an internal microprocessor chip-select generated DTACK, to be completed by the emulator, use the command **cf dtack=0wait** or **cf dtack=1wait**.

If you want all processor cycles, including emulation memory, target memory, and emulator background monitor cycles to be completed by the target system, use command **cf dtack=target**. However, if you select **cf dtack=target** you MUST either drive background monitor cycles to the target using the **cf dbc=en** command or have programmed a microprocessor chip select to supply DTACK for the address range 0H through 1FFFH with no function code. If the target is to supply DTACK for the background monitor, the target must be able to see the background monitor cycles (as described in this paragraph) or it will fail to operate.

Part 2

Service

5

Installing/Updating HP 64700 Firmware

Installing/Updating HP 64700 Firmware

You need to update the emulator firmware if:

- You ordered the emulator probe and the HP 64748C emulation control card 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 emulator.
- You received a firmware update disk from HP.

If you ordered the emulator probe and the HP 64748C emulation control card together, you can ignore this chapter. The control card already contains the correct firmware for the emulator probe.

This chapter shows you how to install or update HP 64700 firmware.

Note

If you are using an HP 64700A, it must contain the optional Flash EPROM memory card before you can install or update HP 64700 system firmware. Flash EPROM memory is standard in the HP 64700B card cage.

The firmware, and the program that downloads it into the HP 64700, are included with the debugger hardware on flexible disks labeled HP64700 EMUL/ANLY FIRMWARE.

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 MC6830x Emulator Graphical User Interface User's Guide.

Updating the firmware using a PC

The firmware, and the program that downloads it into the control card, are included with the MC6830x emulator probe on MS-DOS format floppy disks labeled:

- 64700 SW UTIL.
- 6830X EMULATION FIRMWARE.

The steps to install or update the HP 64700 Emulator/Analyzer Firmware are:

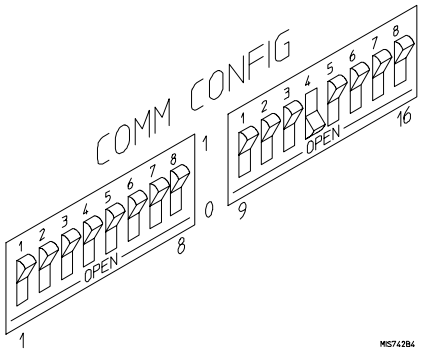
- Step 1. Connect the HP 64700 card cage to your PC.
- Step 2. Install the firmware update utility and the emulator firmware.
- Step 3. Run PROGFLASH to update HP 64700 firmware.
- Step 4. Verify emulator performance.

The above steps are described in detail on the following pages.

Step 1. Connect the HP 64700 card cage to your PC

1 Set the COMM CONFIG switches for RS-232C communication. To do this, locate the DIP switches on the HP 64700 rear panel, and set them as shown below.

You may wish to make a note of the COMM CONFIG switch settings before you change them. That way, you can restore the switch settings when you finish updating the firmware.



Switches 12 and 13 are set to 1 and 0, respectively. This sets the RTS/CTS hardware handshake, which is needed to make sure all characters are processed.

Switches 1, 2, and 3 are set to 0. This sets the baud rate to 9600.

Switch 16 is the LAN/RS-232 switch.

The switch settings are read during the HP 64700 power up routine.

2 Connect an RS-232C modem cable from the PC to the HP 64700 (for example, an HP 24542M 9-pin to 25-pin cable or an HP 13242N 25-pin to 25-pin cable).

You can also use an RS-232C printer cable, but if you do, you must set COMM CONFIG switch 4 to 1.

Insert the 25-pin male connector of the cable into PORT A on the HP 64700 rear panel. See which COM port the cable is connected to on your PC. Some PCs label COM ports as Serial ports (for example, COM1 may be labeled Serial A).

3 Turn ON power to the HP 64700.

The power switch is located on the lower left-hand corner of the front panel. The power lamp at the lower right-hand corner of the front panel will light.

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 64798 firmware require about 300 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 64798 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 to 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-C, and CTRL-Break.

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 C:\HP64700\TABLES\64700TAB file contains entries to establish communications connections for COM1 and COM2. The content of this file is:

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

7 You can edit this file to identify your emulator, if desired:

```
EMUL_COM1 m68302  COM1 OFF 9600 NONE ON 1 8
EMUL_COM2 unknown COM2 OFF 9600 NONE ON 1 8
```

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

```
EMUL_COM3 m68302  COM3 OFF 9600 NONE ON 1 8
EMUL_COM4 unknown COM4 OFF 9600 NONE ON 1 8
```

Software installation is now complete. 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:

```
PROGFLAS
```

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

```
PROGFLAS [-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 (example EMUL_COM1).

The PRODUCT option names the product whose firmware is to be updated (example 64798).

If you enter the PROGFLAS command without options, it begins an interactive session. You can abort the interactive PROGFLAS session by pressing CTRL-C.

PROGFLAS 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.

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

Example

To install or update the HP 64798 emulator firmware in the HP 64700 card cage that is connected to the COM1 port:

```
C> PROGFLAS <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           m68302
          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 64798

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

To update the HP 64798 MC68302 emulator firmware, enter "1".

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

To enable status messages, enter "y".

Step 3: Run "progflash" to update emulator firmware

```
Config file path is /hp64700/update/64798.cfg
System firmware revision required = A.04.00
ROM identifier address = 2FFFF0H
Required hardware identifier = 1FFFH,1FFCH
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
Checking System firmware revision...
Rebooting HP64700...
Downloading flash programming code: /hp64700/lib/npf.X
Checking Hardware id code...
Downloading ROM code: /hp64700/update/64798.X
    Code start 280000H (should equal control ROM start)
    Code size 30C6CH (must be less than control ROM size)
Finishing up...
Rebooting HP64700...
C>
```

You could perform the same update as in the previous example with the following command:

```
C> PROGFLAS -V EMUL_COM1 64798 <RETURN>
```

If the "progflash" routine won't work

For PROGFLAS to work, the selected communications port must have its Interrupt Request Line set to the default value. To set the value of the Interrupt Request Line:

1. Choose Control Panel in the Main window.
2. Choose Ports in the Control Panel window.
3. Choose the COMx port you are using, and click Settings...
4. Click Advanced... in the Settings for COMx dialog box.
5. Select the default value for the Interrupt Request Line in the Advanced Settings for COMx dialog box. The default settings are:

COM1 and COM3	IRQ4
COM2 and COM4	IRQ3

Step 4. Verify emulator performance

- Perform the procedure in the paragraph titled, "To verify the performance of the emulator" in Chapter 6 of this manual.

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 Make sure a clock module or oscillator is installed in the socket on the emulator probe. The emulator must have an internal clock source in order to pass performance verification (see page 55).

6 Turn on power to the HP 64700 Card Cage.

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

8 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. Refer to the list of replacable parts in Chapter 7. Contact your local Hewlett-Packard representative. Refer to the list of Sales Offices in the Support Services book supplied with this manual.

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 B Series Emulation System
    Version:   A.05.05 25Jul94
    Location:  Flash
    System RAM:1 Mbyte

  HP64798C Motorola 68302 Emulator
    Version:   A.00.00 26Jun96
    Control:   HP64748C  Emulation Control Board
    Speed:     25 MHz
    Memory:    1336 Kbytes
      Bank 0:  HP64171A (35ns) or HP64172A (20ns) 256 Kbyte Memory Module
      Bank 1:  HP64171B (35ns) or HP64172B (20ns) 1 Mbyte Memory Module

  HP64740 Emulation Analyzer with External State/Timing Analyzer
    Version:   A.02.03 20Dec93

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

What is pv doing to the emulator?

The performance verification procedures provide a thorough check of the functionality of all of the products installed in the HP 64700 Card Cage. The following is an example test suite for an HP 64700 Card Cage containing an HP 64798 Emulator. To see the exact Test Suite for the HP 64798 Emulator and associated modules in your HP 64700 Card Cage, enter the command: pv -l.

```
R>pv -l
  Tests available in Emulator Subsystem:
    Test # 1 (ABG Type Map)
    Test # 2 (Host Side - ABG 68k RAM)
    Test # 3 (CPU Side - ABG 68k RAM)
    Test # 4 (Emulator Reset)
    Test # 5 (RCP In-Circuit Verification)
    Test # 6 (Host Side - RCP DP Data Bus)
    Test # 7 (Host Side - RCP DP Addr Bus)
    Test # 8 (Tgt CPU Side - RCP DP Data Bus)
    Test # 9 (Break to Monitor)
    Test #10 (Tgt CPU Side - RCP DP Addr Bus)
    Test #11 (Target Chip Selects)
    Test #12 (Target Reset and Halt)
    Test #13 (Target Data Bus)
    Test #14 (Target Address Bus)
    Test #15 (Target Run, *DTACK, *BERR)
    Test #16 (Demo Board SRAM Cell Integrity)
    Test #17 (Dual Port Access)
    Test #18 (Dual Port Cell Integrity)
    Test #19 (RCP SIMM Data Bus)
    Test #20 (RCP SIMM Address Bus)
    Test #21 (RCP SIMM Cell Integrity)
    Test #22 (Background Monitor Breaks)
    Test #23 (Analysis Trace)
    Test #24 (Analysis Break)
    Test #25 (CMB)
    Test #26 (Target DMA / Bus Arbitration)
    Test #27 (Target Interrupt Rqst/Ack)
    Test #28 (Misc Target Control Signals)
    Test #29 (Target I/O Ports A and B)
    Test #30 (Target Serial Ports)
    Test #31 (Demo Board LED)
R>
```

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 the PV test includes a loopback test that performs read and write tests to the demo board. The demo board also stimulates inputs to the emulator.

A FAIL message indicates that one or more of the tested functions is NOT working. In this event, call your local Hewlett-Packard field representative.

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 in the HP 64700B card cage 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 using the Flash EPROM in the HP 64700B card cage.

If you are using an HP 64700A card cage, you can obtain an optional LAN card and an optional Flash EPROM card from your local Hewlett-Packard representative. The optional Flash EPROM card can be inserted in an available slot in the HP 64700A card cage to override old versions of firmware in products with conventional EPROMs. The host controller in the HP 64700A card cage is already programmed to look for a Flash EPROM card if one is installed. To obtain an optional LAN card and optional Flash EPROM card, refer to the list of HP Sales Offices in the Support Services book supplied with this manual.

The software for the HP 64700 MC6830x Graphical User Interface for UNIX workstations is supplied on a tape. To install the product software, follow the instructions shown in the MC6830x Emulator Graphical Interface User's Guide.

The software for the HP 64700 MC6830x Real-Time C Debugger for PCs is shipped on two flexible disks that also contain the latest versions of firmware for the host controller card, analyzer card, LAN card, and a program called *progflash*.

When you load all your new versions of software onto your host computer, you are ready to load the new version of firmware from your host computer to the assemblies in the HP 64700 Card Cage. To load the new firmware, follow the procedures described in Chapter 5, "Installing/Updating HP 64700 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 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
  68302--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 MC6830x 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 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.

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 procedure, "To verify performance of the emulator," 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 have RS-232 connection problems

☐ Windows 3.1 only allows two active RS-232 connections at a time. To be warned when you violate this restriction, choose Always Warn in the Device Contention group box under 386 Enhanced in the Control Panel.

☐ Use the "Terminal" program (usually found in the Accessories windows program group) to set up the "Communications..." settings as follows:

```
Baud Rate" 9600
Data Bits: 8
Parity: None
Flow Control: Hardware
Stop Bits: 1
```

When you are connected, press the Enter key. You should get a prompt back. If nothing echos back, check the switch settings on the back of the emulator:

Switches 1 through 3 set the baud rate, as follows:

```
S1   S2   S3
0    0    0    9600
0    0    1    19200
0    1    0    2400
```

Switches 12 and 13 must be set to 1 and 0, respectively. This sets the RTS/CTS hardware handshake which is needed to make sure all characters are processed.

All other switches should be in the "0" position, especially switch 16 on the HP 64700 (which selects LAN/Serial interface).

Remember that if you change any of the switch positions, you must turn OFF power to the HP 64700 and turn it ON again before the changes will take effect.

☐ If the switches are in the correct position and you still do not get a prompt when you press return, check the following:

- Turn OFF power to the HP 64700 and then turn it ON again. Press return to see if you get a prompt.

- Check to make sure the RS-232 cable is connected to the correct port on your PC, and that the cable is appropriate for connecting the PC to a DCE device. If the cable is intended to connect the PC to a DTE device, set switch 4 to 1 (which makes the emulator a DTE device). Turn OFF power to the HP 64700. Then turn power ON, and try again.
- Check to make sure your RS-232 cable has the RTS, CTS, DSR, DCD, and DTR pins supported. If your PC RS-232 connection is a 9-pin male connection, HP cable number 24542M will work (set switch 4 to 0 if you use this cable). If your PC has a 25-pin RS232 connector, HPO cable number 13242N will work (set switch 4 to 0).

☐ **Always turn ON the HP 64700 before attempting to connect via RS-232. When using certain RS-232 cards, connecting to an RS-232 port where the HP 64700 is turned OFF (or not connected) will halt the PC. The only way to restore operation is to reboot the PC.**

☐ **If the PC fails to connect to the HP 64700, or connects but soon reports over-run errors or times out, check the following:**

- The serial interface in your PC may be insufficiently buffered. Use serial interface cards that include 16550AF UARTs (or better). To see which UART is used in your serial interface card:

  1. Enter the DOS command, "MSD".
  2. Within the window that appears, click on "COM Ports...".
  3. Within the COM Ports window, look at the entry beside "UART Chip Used."

- Turn off all unnecessary TSRs and other applications to allow the processor to more frequently service the serial interface.

If you see "Unable to break" when trying to initialize the graphical user interface

- ☐ Check to make sure you have an oscillator or crystal module installed in the probe clock socket. A working internal clock which must be detected during initialization of the graphical user interface even if you intend to use an external clock after initialization.

If emulation memory behavior is erratic

- ☐ Ensure that you have answered the configuration question correctly for the memory modules in use. The configuration question establishes the required number of wait states to be used within the emulator.
- ☐ Check that emulation memory is not configured as DRAM.

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 are not supported.

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 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 disassembler which operand state belongs with the named instruction state.

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 instrument requires cleaning

☐ If this instrument requires cleaning, disconnect it from all power sources and clean it with a mild detergent and water. Make sure the instrument is completely dry before reconnecting it to a power source.

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.
- SIMM 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 64798C Probe and Demo Board for MC68302		
64700 SW Utilities	64700-17534	
68302 Emulator Firmware	64798-17501	
MC68302 Probe Board Assy for HP 64798C	64798-62101	64798-69101
(Order the following parts separately:)		
Top Plastic Cover	64798-04101	
Bottom Plastic Cover	64783-04102	
Plastic Rivets Kit (rivets and washers)	64748-68700	
PGA Pin Protector, plastic	5181-0206	
Male/male header, 13x13 double header	5042-1724	
Clock oscillator, xtal, 20 MHz	1813-0885	
External +5V Power Cable	5181-0201	
Clock Module Kit (five break-off modules)	64798-66511	
Demo Board for MC68302 emul probe	64798-66506	
(Order the following parts separately:)		
Fuse 3A 250V	2110-0837	
Probing Accessories (not included with emulator, order separately):		
Kit, 132-pin PQFP Probe Assembly (includes)	E3437A	
Transition Board	E3437-63201	
132-pin PQFP Flexible Adapter Cable	5081-7736	
132-pin PQFP Dummy Part Kit	64748-87608	
144-pin Elastomeric Adapter	E5336A	
144-pin TQFP Flexible Adapter Cable	E5338A	
144-pin TQFP Dummy Part Kit	E3435-87601	
Transition socket, MC68302 PGA (includes)	E5367A	
Plastic alignment tab	E5367-94301	

Main Assembly **Component Part**	**New**	**Exchange**
HP 64798E Probe and Demo Board for MC68EN302		
64700 SW Utilities	64700-17534	
68EN302 Emulator Firmware	64798-17501	
MC68EN302 Probe Board Assy for HP 64798F	64798-62104	64798-69104
(Order the following parts separately:)		
Top Plastic Cover	64798-04101	
Bottom Plastic Cover	64783-04102	
Plastic Rivets Kit (rivets and washers)	64748-68700	
PGA Pin Protector, plastic	5181-0206	
Male/male header, 13x13 double header	5042-1724	
Clock oscillator, xtal, 20 MHz	1813-0885	
External +5V Power Cable	5181-0201	
Clock Module Kit (five break-off modules)	64798-66511	
Demo Board for MC68EN302 emul probe	64798-66513	
(Order the following parts separately:)		
Fuse 3A 250V	2110-0837	
Probing Accessories (not included with emulator, order separately):		
144-pin Elastomeric Adapter	E5336A	
144-pin TQFP Flexible Adapter Cable	E5338A	

Main Assembly **Component Part**	**New**	**Exchange**
HP 64798F Probe and Demo Board for MC68LC302		
64700 SW Utilities	64700-17534	
68LC302 Emulator Firmware	64798-17501	
MC68LC302 Probe Board Assy for HP 64798F	64798-62103	64798-69103
(Order the following parts separately:)		
Top Plastic Cover	64798-04101	
Bottom Plastic Cover	64783-04102	
Plastic Rivets Kit (rivets and washers)	64748-68700	
PGA Pin Protector, plastic	5181-0206	
Male/male header, 11x11 double header	1200-1996	
Clock oscillator, xtal, 20 MHz	1813-0885	
External +5V Power Cable	5181-0201	
Clock Module Kit (five break-off modules)	64798-66511	
Demo Board for MC68LC302 emul probe	64798-66508	
(Order the following parts separately:)		
Fuse 3A 250V	2110-0837	
Probing Accessories (not included with emulator, order separately):		
Probing Kit, 100-pin TQFP	E5356A	

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 (without external cable or egress panel)	64748-66515	64748-69515
Wrist strap	9300-1405	
HP 64171A 256-Kbyte, 35-nS SIMM Module	64171A	64171-69503
HP 64171B 1-Mbyte, 35-nS SIMM Module	64171B	64171-69502
HP 64172A 256-Kbyte, 20-nS SIMM Module	64172A	64172-69501
HP 64172B 1-Mbyte, 20-nS SIMM Module	64172B	64172-69502
HP 64173A 4-Mbyte, 25-nS SIMM 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	

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 (page 143). 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 online 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:

```
C:> 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:

```
C:> 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. They will be similar to the following:

```
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**

Graphical Interfaces

HP provides several interfaces for this emulator:

- 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.

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 (for example, register contents, memory contents, and high-level source code) in separate windows on the display at the same time.
- Simplify emulator configuration by clearly displaying configuration options and their relationship to one another.
- Automate frequent tasks through menu selections, command files, user-defined action keys on a workstation, and user-defined buttons in the button window of a PC.
- Work with compilers and assemblers to shorten the edit-compile-execute cycle.

9

MC68302/MC68EN302 Specifications and Characteristics

Processor compatibility

The HP 64798C Emulator supports MC68302 microprocessors operating at 25 MHz at 5 volts for the 132 PQFP package and the 144 TQFP package using optional accessories.

The HP 64798E Emulator supports MC68EN302 microprocessors operating at 25 MHz at 5 volts for the 144 TQFP package using optional accessories.

Electrical

Clock speed

The maximum clock speed of the MC68302 and MC68EN302 emulators is 25 MHz with no wait states required for emulation or target memory. The minimum clock speed required is 8 MHz. These clock speeds are supported when using any of the following emulator SIMM types:

- 64171A 256-Kbyte 35-ns SIMM
- 64171B 1-Mbyte 35-ns SIMM
- 64172A 256-Kbyte 20-ns SIMM
- 64172B 1-Mbyte 20-ns SIMM
- 64173A 4-Mbyte 25-ns SIMM

HP 64798C/E maximum ratings

Characteristic	Symbol	Value	Unit
Supply Voltage	V_{CC}	–0.3 to +5.5	V
Input Voltage	V_{in}	–0.3 to +5.5	V
Maximum Operating Ambient Temperature	T_A	40	deg C
Minimum Operating Ambient Temperature	T_A	0	deg C
Storage Temperature Range	T_{stg}	–40 to +70	deg C

HP 64798C/E DC electrical characteristics

Characteristic	Signal	Symbol	64798C/E Emulator		Unit
			Min	Max	
Input High Voltage	All Except EXTAL	V_{IH}	2.0	V_{DD}	V
Input Low Voltage	All Except EXTAL	V_{IL}	V_{SS} -0.3	0.8	V
Input High Voltage	EXTAL	V_{CIH}	4.0	V_{DD}	V
Input Low Voltage	EXTAL	V_{CIL}	V_{SS} -0.3	0.6	V
Output High Voltage (I_{OH} = 400 uA)	$\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, $\overline{IACK7}$, R/$\overline{W}$, $\overline{CS0}$-$\overline{CS3}$, FC0-FC2 (Unbuffered)	V_{OH}	V_{DD} -1.0	-	V
	$\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, $\overline{IACK7}$, R/$\overline{W}$, $\overline{CS0}$-$\overline{CS3}$, FC0-FC2 (Buffered)		3.5	-	
	CLKO		VDD -1.0	-	
	All Other Outputs		VDD -1.0	-	
Output Low Voltage (I_{OL} = 3.2 mA)	A1-A23, PB0-PB11, FC0-FC2, $\overline{CS0}$-$\overline{CS3}$, IAC, $\overline{AVEC}$, $\overline{BG}$, RCLK1, RCLK2, TCLK1, TCLK2, TCLK3, $\overline{RTS1}$, $\overline{RTS2}$, $\overline{RTS3}$, SDS2, PA12, RXD2, RXD3, $\overline{CTS2}$, $\overline{CD2}$, $\overline{CD3}$, $\overline{DREQ}$, BRG1	V_{OL}	-	0.5	V
	CLKO		-	0.4	

Characteristic	Signal	Symbol	64798C/E Emulator		Unit
			Min	Max	
Output Low Voltage (I_{OL} = 5.3 mA) Note 5	D0-D15, $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, R/$\overline{W}$, $\overline{BERR}$, $\overline{BGACK}$, BCLR, DTACK, DACK, RMC, $\overline{RESET}$	V_{OL}	-	0.5	V
Output Low Voltage (I_{OL} = 7.0 mA)	TXD1, TXD2, TXD3	V_{OL}	-	0.5	V
Output Low Voltage (I_{OL} = 8.9 mA)	$\overline{DONE}$, $\overline{HALT}$	V_{OL}	-	0.5	V
Input Low Current (V_{IL} = 0V) Notes 4 and 5	A1	I_{IL}	-	0.4	mA
	A2, A3, $\overline{IPL0}$-$\overline{IPL2}$, BUSW, DISCPU, $\overline{FRZ}$, $\overline{BGACK}$		-	0.2	
	A4-A23, D0-D15		-	70	uA
	FC0-FC2, PB0 ($\overline{IACK7}$)		-	0.5	mA
	$\overline{AS}$		-	1.0	
	$\overline{UDS}$, $\overline{LDS}$, R/$\overline{W}$		-	0.6	
	$\overline{RESET}$, $\overline{HALT}$, $\overline{BERR}$, $\overline{DTACK}$		-	0.7	
	All Other Signals		-	20	uA
Input High Current (V_{IH} = V_{DD}) Note 4	A1, FC0-FC2, $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, R/$\overline{W}$, PB0 ($\overline{IACK7}$)	I_{IH}	-	60	uA
	A2, A3, $\overline{RESET}$, $\overline{HALT}$, $\overline{BERR}$, $\overline{DTACK}$, $\overline{BGACK}$		-	30	
	All Other Signals		-	20	

Characteristic	Signal	Symbol	64798C/E Emulator		Unit
			Min	Max	
Input Capacitance (With 144-pin TQFP or 132-pin PQFP adaptor cable)	A1-A3	C_{IN}	-	70	pF
	A4-A23		-	60	
	D0-D15, FC0-FC2		-	90	
	EXTAL Note 1		-	50	
	$\overline{\text{RESET}}$, $\overline{\text{HALT}}$, $\overline{\text{BERR}}$, $\overline{\text{AS}}$ Note 1		-	105	
	$\overline{\text{UDS}}$, $\overline{\text{LDS}}$, R/$\overline{\text{W}}$		-	75	
	$\overline{\text{IPL0}}$-$\overline{\text{IPL2}}$, $\overline{\text{BGACK}}$		-	55	
	BUSW, $\overline{\text{DISCPU}}$, $\overline{\text{FRZ}}$, $\overline{\text{DTACK}}$, PB0 ($\overline{\text{IACK7}}$) Note 5		-	65	
	PB1-PB11		-	45	
	$\overline{\text{PA0}}$-$\overline{\text{PA15}}$, $\overline{\text{RXD1}}$, $\overline{\text{RCLK1}}$, TCLK1, $\overline{\text{CD1}}$, $\overline{\text{CTS1}}$, $\overline{\text{CD3}}$, $\overline{\text{CTS3}}$, $\overline{\text{BR}}$, $\overline{\text{AVEC}}$		-	40	

Characteristic	Signal	Symbol	64798C/E Emulator		Unit
			Min	Max	
Load Capacitance (Additional capacitance added to processor outputs by emulator with 144-pin TQFP or 132-pin PQFP adaptor cable)	A1-A3	C_{EL}	-	60	pF
	A4-A23		-	50	
	D0-D15, FC0-FC2 Note 3		-	80	
	$\overline{CS0}$-$\overline{CS3}$ Note 3		-	60	
	$\overline{RESET}$, $\overline{HALT}$, $\overline{BERR}$, $\overline{AS}$ Notes 1, 3		-	95	
	$\overline{UDS}$, $\overline{LDS}$, R/$\overline{W}$ Note 3		-	65	
	$\overline{BG}$, $\overline{BGACK}$		-	45	
	$\overline{DTACK}$, PB0 ($\overline{IACK7}$) Note 3		-	55	
	PB1-PB11, IAC, $\overline{RMC}$ Note 5		-	35	
	PA0-PA15, TXD1, RCLK1, TCLK1, $\overline{RTS1}$, $\overline{RTS3}$, BRG1, SPCLK, $\overline{BCLR}$		-	30	
	CLKO (at processor) Note 2		-	50	
Power		V_{DD}	4.5	5.5	V
Common		V_{SS}	0	0	V
Power Supply Current Drawn from Target System		I_{DD}	-	200	mA

Notes

1 EXTAL and $\overline{AS}$ are additionally terminated with 100 ohms in series with 100 pF on the emulator.
2 CLKO is buffered before it is sent to the target.
3 FC0-FC2, $\overline{CS0}$-$\overline{CS2}$, $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, R/$\overline{W}$, and $\overline{IACK7}$ unbuffered.
4 The Three-State Leakage Current (I_{TSI}) specification for those signals that can be outputs is the same as specified for Input Current at the given input voltage.
5 The $\overline{RMC}$, IAC, $\overline{FRZ}$, and DISCPU pins are not available on the MC68EN302.

HP 64798C/E AC Electrical Specifications

For the following tables *Buffered* and *Unbuffered* refer to whether or not the various processor control signals are buffered by the emulator or not. There are four configuration items that control buffering for the following four groups of signals:

- Buffer Strobes — $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, $\overline{IACK7}$
- Buffer Read/Write — R/$\overline{W}$
- Buffer Chip Selects — $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$
- Buffer Function Codes — FC0, FC1, FC2

The *All Unbuffered* column shows the emulator timing when all four configuration items are set to "unbuffered". The *All Buffered* column shows the emulator timing when all four configuration items are set to "buffered".

If some, but not all four configuration items are set to buffered, determine for each of the two signals in the timing relationship if it is buffered or unbuffered. If both are unbuffered, then use the *All Unbuffered* column, if both are buffered use the *All Buffered* column. If the first signal is buffered but the second is unbuffered, use the *Buffered / Unbuffered* column, and likewise if the first signal is unbuffered and the second signal is buffered use the *Unbuffered / Buffered* column. Footnotes help to make it clear which signal is buffered and which is unbuffered.

If neither or only one of the two signals in a given timing relationship can be configured for buffering/unbuffering, then the *Buffered / Unbuffered* and *Unbuffered / Buffered* columns are not used. In the case where just one signal can be buffered, use the *All Unbuffered* column if the signal is unbuffered, and the *All Buffered* column if the signal is buffered.

The timing specifications for some of the peripherals on the processor are not included here since they are unaffected by the emulator logic.

The subscript numbers in the *Characteristic* column refer to the notes in the Motorola data book.

Clock Timing

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
	Frequency of Operation	8	25	8	25	8	25					MHz
1	Clock Period (EXTAL)	40	125	40	125	40	125					ns
2,3	Clock Pulse Width (EXTAL)	16	62.5	19	62.5	19	62.5					ns
4,5	Clock Rise and Fall Times (EXTAL)	-	4	-	4	-	4					ns
5a	EXTAL to CLKO Delay 1,2	2	7	3.5 Note	12 Note	3.5 Note	12 Note					ns

Note

CLKO skew from the rising and falling edges of EXTAL will not differ from each other by more than 2.5 ns, if the EXTAL rise time equals the EXTAL fall time.

IMP Bus Master Cycles

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
6	Clock High to Address Valid	0	30	-5	28.5	-5	28.5					ns
	Clock High to FC Valid	0	30	-5	29	-3	38.5					
7	Clock High to Address, Data Bus High Impedance (Maximum)	-	33	-	31.5	-	31.5					ns
8	Clock High to Address Invalid	0	-	-5	-	-5	-					ns
	Clock High to FC Invalid	0	-	-5	-	-3	-					
9	Clock High to $\overline{AS}$, $\overline{DS}$ Asserted [1]	3	20	-2	19	-1	26					ns
11	Address Valid to $\overline{AS}$, $\overline{DS}$ Asserted [2]	10	-	10	-	11	-					ns
	FC Valid to $\overline{AS}$, $\overline{DS}$ Asserted [2]	10	-	10	-	4	-	3_b	-	11_a	-	
12	Clock Low to $\overline{AS}$, $\overline{DS}$ Negated [1]	-	20	-	19	-	26					ns

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
13	$\overline{AS}$, $\overline{DS}$ Negated to Address Invalid 2	10	-	10	-	2.5	-					ns
	$\overline{AS}$, $\overline{DS}$ Negated to FC Invalid 2	10	-	10	-	4.5	-	2.5a	-	11.5b	-	
14	$\overline{AS}$ (and $\overline{DS}$ Read) Width Asserted 2	80	-	80	-	77	-					ns
14A	$\overline{DS}$ Width Asserted (Write) 2	40	-	40	-	37	-					ns
15	$\overline{AS}$, $\overline{DS}$ Width Negated 2	40	-	40	-	37	-					ns
16	Clock High to Control Bus High Impedance	-	33	-	33	-	35 Note 2					ns
17	$\overline{AS}$, $\overline{DS}$ Negated to R/$\overline{W}$ Invalid 2	10	-	10	-	3.5	-	2.5c	-	10.5d	-	ns
18	Clock High to R/$\overline{W}$ High 1	-	20	-	19	-	26					ns
20	Clock High to R/$\overline{W}$ Low 1	-	20	-	19	-	26					ns
20A	$\overline{AS}$ Asserted to R/$\overline{W}$ Low (Write) 2,6	-	7	-	7	-	13.5	-	6.5c	-	14.5d	ns
21	Address Valid to R/$\overline{W}$ Low (Write) 2	10	-	10	-	11						ns
	FC Valid to R/$\overline{W}$ Low (Write) 2	10	-	10	-	1	-	0e	-	10.5f	-	

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
22	R/$\overline{W}$ Low to $\overline{DS}$ Asserted (Write) 2	20	-	20	-	16.5	-	15.5[d]	-	21[c]	-	ns
23	Clock Low to Data-Out Valid	-	20	-	19	-	19					ns
25	$\overline{AS}$, $\overline{DS}$ Negated to Data-Out Invalid (Write) 2	10	-	10	-	2.5	-					ns
26	Data-Out Invalid to $\overline{DS}$ Asserted (Write) 2	10	-	10	-	10.5	-					ns
27	Data-In Valid to Clock Low (Setup Time on Read) 5	5	-	10	-	10	-					ns
28	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{DTACK}$ Negated (Asynchronous Hold) 2	0	75	0	75	-1	67.5					ns
29	$\overline{AS}$, $\overline{DS}$ Negated to Data-In Invalid (Hold Time on Read)	0 Note 1	-	0	-	-1	-					ns
30	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{BERR}$ Negated	0	-	0	-	-1	-					ns
31	$\overline{DTACK}$ Asserted to Data-In Valid (Setup Time) 2,5	-	33	-	33	-	33					ns

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
32	$\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ Input Transition Time	-	150	-	150	-	150					ns
33	Clock High to $\overline{\text{BG}}$ Asserted	-	20	-	19	-	19					ns
34	Clock High to $\overline{\text{BG}}$ Negated	-	20	-	19	-	19					ns
35	$\overline{\text{BR}}$ Asserted to $\overline{\text{BG}}$ Asserted 11	2.5	4.5	2.5	4.5	2.5	4.5					clks
36	$\overline{\text{BR}}$ Negated to $\overline{\text{BG}}$ Negated 7	1.5	2.5	1.5	2.5	1.5	2.5					clks
37	$\overline{\text{BGACK}}$ Asserted to $\overline{\text{BG}}$ Negated	2.5	4.5	2.5	4.5	2.5	4.5					clks
37A	$\overline{\text{BGACK}}$ Asserted to $\overline{\text{BR}}$ Negated 8	10	1.5	10	1.5	10	1.5					ns clks
38	$\overline{\text{BG}}$ Asserted to Control, Address, Data Bus High Impedance (AS Negated)	-	33	-	33	-	35 Note 2					ns
39	$\overline{\text{BG}}$ Width Negated	1.5	-	1.5	-	1.5	-					clks
40	$\overline{\text{BGACK}}$ Asserted to Address Valid	15	-	15	-	15	-					ns
41	$\overline{\text{BGACK}}$ Asserted to $\overline{\text{AS}}$ Asserted	20	-	20	-	20	-					ns

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
44	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{AVEC}$ Negated	0	33	0	33	-1	25.5					ns
46	$\overline{BGACK}$ Width Low	1.5	-	1.5	-	1.5	-					clks
47	Asynchronous Input (Except $\overline{IPL0}$-$\overline{IPL2}$) Setup Time to Clock 5	7	-	12	-	12	-					ns
	Asynchronous Input ($\overline{IPL0}$-$\overline{IPL2}$) Setup Time to Clock 5	7	-	30	-	30	-					
48	$\overline{BERR}$ Asserted to $\overline{DTACK}$ Asserted 2,3	7	-	7	-	7	-					ns
53	Clock High to Data-Out Invalid (Hold Time on Write)	0	-	-5	-	-5	-					ns
55	R/$\overline{W}$ Asserted to Data Bus Impedance Change	0	-	0	-	-7.5	-					ns
56	$\overline{HALT}$/$\overline{RESET}$ Pulse Width 4	10	-	10	-	10	-					clks
57	$\overline{BGACK}$ Negated to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$ Driven	1.5	-	1.5 Note 3	-	0.5	-					clks

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
57A	$\overline{\text{BGACK}}$ Negated to FC Driven	1	-	1	-	0.5	-					clks
58	$\overline{\text{BR}}$ Negated to $\overline{\text{AS}}$, DS, R/W Driven 7	1.5	-	1.5 Note 3	-	0.5	-					clks
58A	$\overline{\text{BR}}$ Negated to FC Driven 7	1	-	1	-	0.5	-					clks
60	Clock High to $\overline{\text{BCLR}}$ Asserted	-	20	-	18.5	-	18.5					ns
61	Clock High to $\overline{\text{BCLR}}$ High Impedance 10	-	20	-	18.5	-	18.5					ns
62	Clock Low (S0 Falling Edge During Read) to $\overline{\text{RMC}}$ Asserted Note 4	-	20	-	18.5	-	18.5					ns
63	Clock High (During Write) to $\overline{\text{RMC}}$ Negated Note 4	-	20	-	18.5	-	18.5					ns
64	$\overline{\text{RMC}}$ Negated to BG Asserted 6	-	20	-	20	-	20					ns

Notes

1 Motorola data sheet does not indicate a value for 25 MHz but shows a value of 0 for 16.67 MHz and 20 MHz.

2 Actual value depends on the 1/2-clock period. The value given is based on a 1/2-clock period of 20 ns. To calculate this value, add 15 ns to the 1/2-clock period.

3 If bus arbitration occurs while in the background monitor, this value is 0.5 clks.
4 RMC pin not available on MC68EN302.

Notes for *Buffered / Unbuffered* and *Unbuffered / Buffered* columns:

a $\overline{AS}$, $\overline{DS}$, $\overline{IACK7}$ buffered; FC unbuffered.
b FC buffered; $\overline{AS}$, $\overline{DS}$, $\overline{IACK7}$ unbuffered.
c $\overline{AS}$, $\overline{DS}$, $\overline{IACK7}$ buffered; R/$\overline{W}$ unbuffered.
d R/$\overline{W}$ buffered; $\overline{AS}$, $\overline{DS}$, $\overline{IACK7}$ unbuffered.
e FC bufferd; R/$\overline{W}$ unbuffered.
f R/$\overline{W}$ buffered; FC unbuffered.

DMA

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
80	$\overline{REQ}$ Asynchronous Setup Time to Clock 1	10	-	15	-	15	-					ns
81	$\overline{REQ}$ Width Low 2	2	-	2	-	2	-					clks
82	$\overline{REQ}$ Low to $\overline{BR}$ Low 3,4	-	2	-	2	-	2					clks
83	Clock High to $\overline{BR}$ Low 3,4	-	20	-	18.5	-	18.5					ns
84	Clock High to $\overline{BR}$ High Impedance 3,4	-	20	-	18.5	-	18.5					ns
85	$\overline{BGACK}$ Low to $\overline{BR}$ High Impedance 3,4	20	-	20	-	20	-					ns
86	Clock High to $\overline{BGACK}$ Low	-	20	-	18.5	-	18.5					ns
87	$\overline{AS}$ and $\overline{BGACK}$ High (The Last One) to $\overline{BGACK}$ Low (When $\overline{BG}$ is Asserted)	1.5	2.5 +20	1.5	2.5 +20	1.5	2.5 +20					ns clks
88	$\overline{BG}$ Low to $\overline{BGACK}$ Low (No Other Bus Master) 3,4	1.5	2.5 +20	1.5	2.5 +20	1.5	2.5 +20					ns clks

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
89	$\overline{BR}$ High Impedance to $\overline{BG}$ High 3,4	0	-	0	-	0	-					ns
90	Clock on which $\overline{BGACK}$ Low to Clock on which $\overline{AS}$ Low	2	2	2	2	2	2					clks
91	Clock High to $\overline{BGACK}$ High	-	20	-	18.5	-	18.5					ns
92	Clock Low to $\overline{BGACK}$ High Impedance	-	10	-	8.5	-	8.5					ns
93	Clock High to $\overline{DACK}$ Low	-	20	-	18.5	-	18.5					ns
94	Clock Low to $\overline{DACK}$ High	-	20	-	18.5	-	18.5					ns
95	Clock High to $\overline{DONE}$ Low (Ouptut)	-	20	-	18.5	-	18.5					ns
96	Clock Low to $\overline{DONE}$ High Impedance	-	20	-	18.5	-	18.5					ns
97	$\overline{DONE}$ Input Low to Clock High (Asynchronous Setup)	10	-	15	-	15	-					ns

External Master Internal Asynchronous Read/Write Cycles

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
100	R/$\overline{W}$ Valid to $\overline{DS}$ Low	0	-	0	-	0	-					ns
101	$\overline{DS}$ Low to Data-In Valid	-	20	-	20	-	20					ns
102	$\overline{DTACK}$ Low to Data-In Hold Time	0	-	0	-	0	-					ns
103	R/$\overline{W}$ Valid to $\overline{DS}$ Low	0	-	0	-	0	-					ns
104	$\overline{DTACK}$ Low to $\overline{AS}$, $\overline{DS}$ High	0	-	0	-	0	-					ns
105	$\overline{DS}$ High to $\overline{DTACK}$ High	-	30	-	30	-	30					ns
106	$\overline{DS}$ Inactive to $\overline{AS}$ Inactive	0	-	0	-	0	-					ns
107	$\overline{DS}$ High to R/$\overline{W}$ High	0	-	0	-	0	-					ns
108	$\overline{DS}$ High to Data High Impedance	-	30	-	30	-	30					ns
108A	$\overline{DS}$ High to Data-Out Hold Time	0	-	0	-	0	-					ns

External Master Internal Asynchronous Read/Write Cycles

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
109A	$\overline{\text{Data-Out}}$ Valid to $\overline{\text{DTACK}}$ Low	10	-	10	-	10	-					ns

External Master Internal Synchronous Read/Write Cycles

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
110	Address Valid to $\overline{AS}$ Low	10	-	10	-	10	-					ns
111	$\overline{AS}$ Low to Clock High	20	-	25	-	25	-					ns
112	Clock Low to $\overline{AS}$ High	-	30	-	25	-	25					ns
113	$\overline{AS}$ High to Address Hold Time (Write)	0	-	0	-	0	-					ns
114	$\overline{AS}$ Inactive Time	1	-	1	-	1	-					clks
115	$\overline{DS}$ Low to Clock High 2	27	-	32	-	32	-					ns
116	Clock Low to $\overline{DS}$ High	-	30	-	25	-	25					ns
117	R/$\overline{W}$ Valid to Clock High 2	20	-	25	-	25	-					ns
118	Clock High to R/$\overline{W}$ High	-	30	-	25	-	25					ns
119	$\overline{AS}$ Low to IAC High Note 1	-	27	-	27	-	27					ns
120	$\overline{AS}$ High to IAC Low Note 1	-	27	-	27	-	27					ns

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
121	$\overline{AS}$ Low to $\overline{DTACK}$ Low (0 Wait States)	-	30	-	30	-	30					ns
122	Clock Low to $\overline{DTACK}$ Low (1 Wait State)	-	20	-	18.5	-	18.5					ns
123	$\overline{AS}$ High to $\overline{DTACK}$ High	-	30	-	30	-	30					ns
124	$\overline{DTACK}$ High to $\overline{DTACK}$ High Impedance	-	10	-	10	-	10					ns
125	Clock High to Data-Out Valid	-	20	-	18.5	-	18.5					ns
126	$\overline{AS}$ High to Data High Impedance	-	30	-	30	-	30					ns
127	$\overline{AS}$ High to Data-Out Hold	0	-	0	-	0	-					ns
128	$\overline{AS}$ High to Address Hold Time (Read)	0	-	0	-	0	-					ns
129	$\overline{DS}$ Inactive Time	1	-	1	-	1	-					clks
130	Data-In Valid to Clock Low	20	-	25	-	25	-					ns
131	Clock Low to Data-In Hold Time	10	-	8.5	-	8.5	-					ns

Notes:

1. $\overline{IAC}$ pin not available on MC68EN302.

Internal Master Internal Read/Write Cycles

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
140	Clock High to IAC High Note 1	-	27	-	25.5	-	25.5					ns
141	Click Low to IAC Low Note 1	-	27	-	25.5	-	25.5					ns
142	Clock high to DTACK Low	-	30	-	28.5	-	28.5					ns
143	Clock Low to DTACK High	-	27	-	25.5	-	25.5					ns
144	Clock High to Data-Out Valid	-	20	-	19	-	19					ns
145	$\overline{AS}$ High to Data-Out Hold Time	0	-	0	-	-7.5	-					ns

Notes:

1. $\overline{IAC}$ pin not available on MC68EN302.

Chip Select Timing Internal Master

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
150	Clock High to $\overline{CS}$, $\overline{IACK7}$ Low 2	0	27	-5	26	-4	33					ns
	Clock High to $\overline{IACK1}$, $\overline{IACK6}$ Low 2	0	27	-5	25.5	-5	25.5					
151	Clock Low to $\overline{CS}$, $\overline{IACK7}$ High 2	0	27	-5	26	-4	33					ns
	Clock Low to $\overline{IACK1}$, $\overline{IACK6}$ High 2	0	27	-5	25.5	-5	25.5					
152	$\overline{CS}$ Width Negated	40	-	40	-	37	-					ns
153	Clock High to $\overline{DTACK}$ Low (0 Wait States)	-	30	-	28.5	-	28.5					ns
154	Clock High to $\overline{DTACK}$ Low (1-6 Wait States)	-	20	-	18.5	-	18.5					ns
155	Clock Low to $\overline{DTACK}$ High	-	27	-	25.5	-	25.5					ns
156	Clock High to $\overline{BERR}$ Low 1	-	27	-	25.5	-	25.5					ns

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
157	Clock Low to $\overline{\text{BERR}}$ High Impedance [1]	-	27	-	25.5	-	25.5					ns
158	$\overline{\text{DTACK}}$ High to DTACK High Impedance	-	10	-	10	-	10					ns
171	Clock Low (end of S6) to Data-In Invalid (Hold Time on Read)	5	-	3.5	-	3.5	-					ns
172	$\overline{\text{CS}}$ Negated to Data-Out Invalid (Write)	7	-	7	-	-0.5	-					ns
173	Address Valid to $\overline{\text{CS}}$ Asserted	15	-	15	-	16	-					ns
	FC Valid to $\overline{\text{CS}}$ Asserted	15	-	15	-	6	-	5[h]	-	16[g]	-	
174	$\overline{\text{CS}}$ Negated to Address Invalid	12	-	12	-	4.5	-					ns
	$\overline{\text{CS}}$ Negated to FC Invalid	12	-	12	-	6.5	-	4.5[g]	-	13.5[h]	-	
175	$\overline{\text{CS}}$ Low Time (0 Wait States)	80	-	80	-	77.5	-					ns
176	$\overline{\text{CS}}$ Negated to R/$\overline{\text{W}}$ Invalid	7	-	7	-	0.5	-	-0.5[i]	-	7.5[j]	-	ns

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
177	$\overline{CS}$ Negated to R/$\overline{W}$ Low (Write)	-	8	-	8	-	14.5	-	7.5[i]	-	15.5[j]	ns
178	$\overline{CS}$ Negated to Data-In Invalid (Hold Time on Read)	0	-	0	-	-7.5	-					ns

Notes for *Buffered / Unbuffered* and *Unbuffered / Buffered* columns:

[g] $\overline{CS}$ buffered; $\underline{FC}$ unbuffered..
[h] $\underline{FC}$ buffered; $\underline{CS}$ unbuffered.
[i] $\overline{CS}$ buffered; R/$\overline{W}$ unbuffered.
[j] R/$\overline{W}$ buffered; $\overline{CS}$ unbuffered.

Chip Select Timing External Master

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
154	Clock High to $\overline{\text{DTACK}}$ Low (1-6 Wait States)	-	20	-	18.5	-	18.5					ns
160	$\overline{\text{AS}}$ Low to $\overline{\text{CS}}$ Low	-	20	-	20	-	20					ns
161	$\overline{\text{AS}}$ High to $\overline{\text{CS}}$ High	-	20	-	20	-	20					ns
162	Address Valid to $\overline{\text{AS}}$ Low	10	-	10	-	10	-					ns
163	R/$\overline{\text{W}}$ Valid to $\overline{\text{AS}}$ Low [1]	10	-	10	-	10	-					ns
164	$\overline{\text{AS}}$ Negated to Address Hold Time	0	-	0	-	0	-					ns
165	$\overline{\text{AS}}$ Low to $\overline{\text{DTACK}}$ Low (0 Wait States)	-	30	-	30	-	30					ns
167	$\overline{\text{AS}}$ High to $\overline{\text{DTACK}}$ High	-	20	-	20	-	20					ns
168	$\overline{\text{AS}}$ Low to $\overline{\text{BERR}}$ Low [2]	-	20	-	20	-	20					ns
169	$\overline{\text{AS}}$ High to $\overline{\text{BERR}}$ High Impedance [2,3]	-	20	-	20	-	20					ns

Parallel I/O

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
180	Input Data Setup Time to Clock Low	14	-	19	-	19	-					ns
181	Clock Low to Input Data Hold Time	19	-	17.5	-	17.5	-					ns
182	Clock High to Data-Out Valid (CPU Writes Data, Control, or Direction)	-	24	-	22.5	-	22.5					ns

Interrupts

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
190	Interrupt Pulse Width Low IRQ (Edge Triggered Mode)	34	-	34	-	34	-					ns
191	Minimum Time Between Active Edges	3		3		3						clks

Timers

Num	Characteristic	Motorola 25 MHz 68302/ 68EN302		64798C/E Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
200	Timer Input Capture Pulse Width	34	-	34	-	34	-					ns
201	TIN Clock Low Pulse Width	34	-	34	-	34	-					ns
202	TIN Clock High Pulse Width and Input Capture High Pulse Width	2	-	2	-	2	-					clks
203	TIN Clock Cycle Time	3		3	-	3	-					clks
204	Clock High to TOUT Valid	-	24	-	22.5	-	22.5					ns
205	$\overline{\text{FRZ}}$ Input Setup Time to Clock High Note 1	14	-	29	-	29	-					ns
206	Clock High to $\overline{\text{FRZ}}$ Input Hold Time Note 1	7	-	4.5	-	4.5	-					ns

Notes:

1. $\overline{\text{FRZ}}$ pin not available on MC68EN302.

Physical MC68302/MC68EN302

Emulator Dimensions

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

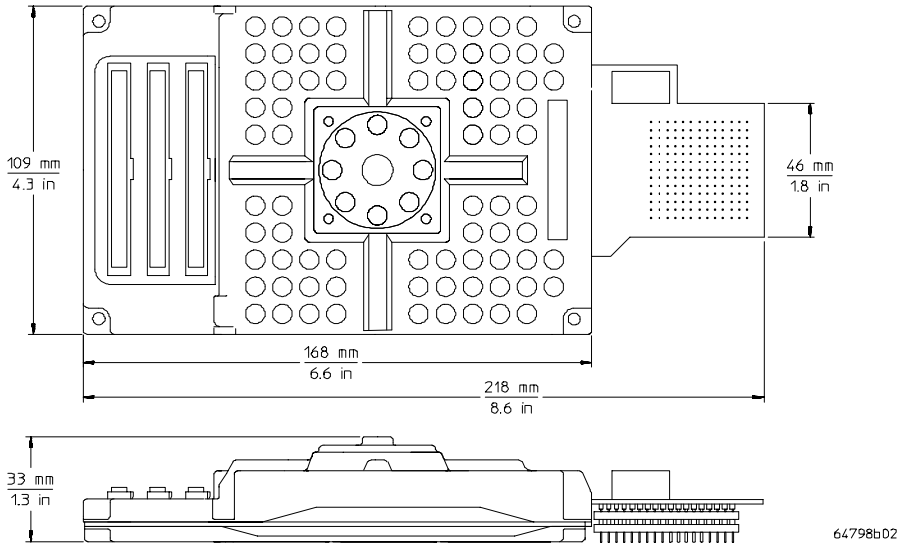
Emulator Weight

Probe alone: 0.3 kg (10 oz).

Cable Length

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

Probe dimensions



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

Input

74HCT132 with 135 ohms to ground in parallel.

Maximum input:

5V above Vcc; 5V 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.

Pinout for 169-pin PGA on HP 68302 Emulator Probe			
68302 Microprocessor Signal Name	**Emulator PGA Pin Number**	**68302 TQFP Pin Number**	**68302 PQFP Pin Number**
A1	H3	126	1
A2	G1	125	2
A3	J1	124	3
A4	G3	122	5
A5	G2	121	6
A6	H5	120	7
A7	H4	119	8
A8	H2	118	9
A9	H1	117	10
A10	J3	116	11
A11	J2	115	12
A12	K2	113	14
A13	K1	112	15
A14	L2	111	16
A15	L1	110	17
A16	M3	106	19
A17	N3	105	20
A18	L4	104	21
A19	M4	103	22
A20	K5	101	24
A21	M5	100	25
A22	N5	99	26
A23	K6	98	27
FC0	G4	127	132
FC1	F5	129	130
FC2	E3	130	129
$\overline{CS0}/\overline{IOUT2}$	G5	131	128
$\overline{CS1}$	F1	132	127
$\overline{CS2}$	E1	135	125
$\overline{CS3}$	E2	136	124

Pinout for 169-pin PGA on HP 68302 Emulator Probe			
68302 Microprocessor Signal Name	**Emulator PGA Pin Number**	**68302 TQFP Pin Number**	**68302 PQFP Pin Number**
D0	M11	75	48
D1	N10	76	47
D2	M10	77	46
D3	L10	78	45
D4	L9	80	43
D5	N8	81	42
D6	M8	82	41
D7	K8	83	40
D8	M7	85	38
D9	L7	86	37
D10	K9	87	36
D11	N9	88	35
D12	L8	90	33
D13	K7	91	32
D14	M6	92	31
D15	J6	93	30
PA0/RXD2	L12	70	53
PA1/TXD2	L13	69	54
PA2/RCLK2	K11	68	55
PA3/TCLK2	K12	67	56
PA4/CTS2	J12	64	58
PA5/RTS2	J13	63	59
PA6/CD2	H10	62	60
PA7/BRG2/SDS2	H11	61	61
PA8/RXD3	G9	59	63
PA9/TXD3	J11	58	64
PA10/RCLK3	H9	57	65
PA11/TCLK3	H12	56	66
PA12/BRG3	G13	53	68
PA13/DREQ	E13	52	69
PA14/DACK	E10	51	70
PA15/DONE	G11	50	71

Pinout for 169-pin PGA on HP 68302 Emulator Probe			
68302 Microprocessor Signal Name	**Emulator PGA Pin Number**	**68302 TQFP Pin Number**	**68302 PQFP Pin Number**
PB0/$\overline{\text{IACK7}}$	A6	9	108
PB1/$\overline{\text{IACK6}}$	C5	8	109
PB2/IACK1	B5	7	110
PB3/$\overline{\text{TIN1}}$	C4	6	111
PB4/TOUT1	A4	4	113
PB5/$\overline{\text{TIN2}}$	B3	3	114
PB6/$\overline{\text{TOUT2}}$	A3	2	115
PB7/WDOG	B1	143	117
PB8	C2	142	118
PB9	C1	141	119
PB10	D3	140	120
PB11	D2	139	121
RXD1/L1RXD	M13	71	52
TXD1/L1TXD	D12	41	80
RCLK1/L1CLK	C12	39	82
TCL$\overline{\text{K1}}$/L1SY0/SDS1	D13	40	81
$\overline{\text{CD1}}$/L1SY1	N12	73	50
$\overline{\text{CTS1}}$/L1GR	N13	72	51
$\overline{\text{RTS1}}$/L1RQ/GCIDCL	D11	42	79
BRG1	F13	45	76
SPRXD/$\overline{\text{CTS3}}$	N11	74	49
SPTXD/$\overline{\text{RTS3}}$	E12	43	78
SPCLK/$\overline{\text{CD3}}$	E11	44	77
$\overline{\text{IPL0}}$/$\overline{\text{IRQ1}}$	E7	23	97
$\overline{\text{IPL1}}$/$\overline{\text{IRQ6}}$	A8	24	96
$\overline{\text{IPL2}}$/$\overline{\text{IRQ7}}$	C8	25	95
BUSW	F10	47	74
DISCPU	F9	48	73
$\overline{\text{RESET}}$	B9	28	92
$\overline{\text{HALT}}$	D9	29	91
$\overline{\text{DTACK}}$	A12	35	85
$\overline{\text{BERR}}$	D8	26	94

Pinout for 169-pin PGA on HP 68302 Emulator Probe			
68302 Microprocessor Signal Name	**Emulator PGA Pin Number**	**68302 TQFP Pin Number**	**68302 PQFP Pin Number**
$\overline{AS}$	C7	14	104
$\overline{UDS}/\overline{A0}$	E6	12	106
$\overline{LDS}/\overline{DS}$	B7	13	105
R/$\overline{W}$ / IREQ	D5	15	103
$\overline{BR}$	A10	30	90
$\overline{BG}$	A11	33	87
$\overline{BGACK}$	C10	32	88
$\overline{BCLR}$	B11	34	86
$\overline{IAC}$	D1	138	122
$\overline{RMC}$	E4	137	123
$\overline{AVEC}$	A9	27	93
$\overline{FRZ}$	G12	49	72
EXTAL	D7	19	100
CLKO	C9	22	98
XTAL (no connect)	C6	18	101
NC1	B10	31	89
NC3	F12	46	75
NC4	N1	108	-
NC5	M1	109	-
V_{DD}	B4	5	112
V_{DD}	B8	20	99
V_{DD}	E8	21	-
V_{DD}	C13	38	83
V_{DD}	H13	60	62
V_{DD}	J8	84	39
V_{DD}	N6	96	28
V_{DD}	L6	97	-
V_{DD}	N2	107	18
V_{DD}	F2	128	131

Pinout for 169-pin PGA on HP 68302 Emulator Probe			
68302 Microprocessor Signal Name	**Emulator PGA Pin Number**	**68302 TQFP Pin Number**	**68302 PQFP Pin Number**
GND (V_{SS})	A2	1	116
GND (V_{SS})	B6	10	107
GND (V_{SS})	D6	11	-
GND (V_{SS})	A5	16	102
GND (V_{SS})	A7	17	-
GND (V_{SS})	A13	36	84
GND (V_{SS})	B13	37	-
GND (V_{SS})	F11	54	67
GND (V_{SS})	G10	55	-
GND (V_{SS})	J10	65	57
GND (V_{SS})	K13	66	-
GND (V_{SS})	M9	79	44
GND (V_{SS})	N7	89	34
GND (V_{SS})	L5	94	29
GND (V_{SS})	J7	95	-
GND (V_{SS})	N4	102	23
GND (V_{SS})	K3	114	13
GND (V_{SS})	J4	123	4
GND (V_{SS})	F3	133	-
GND (V_{SS})	F4	134	126
GND (V_{SS})	A1	144	-
Cable Ground	B2, B12, C3, C11, D4, D10, E5, E9, F6, F7, F8, G6, G7, G8, H6, H7, H8, J5, J9, K4, K10, L3, L11, M2, M12		

Pinout for 169-pin PGA on HP 68EN302 Emulator Probe		
68EN302 Microprocessor Signal Name	**Emulator PGA Pin Number**	**68EN302 TQFP Pin Number**
A1	H3	126
A2	G1	125
A3	J1	124
A4	G3	122
A5	G2	121
A6	H5	120
A7	H4	119
A8	H2	118
A9	H1	117
A10	J3	116
A11	J2	115
A12	K2	113
A13	K1	112
A14	L2	111
A15	L1	110
A16	N1	108
A17	N2	107
A18	M3	106
A19	N3	105
A20	M4	103
A21	N4	102
A22	K5	101
A23	M5	100
FC0	G4	127
FC1	F5	129
FC2	E3	130
$\overline{CS0}/\overline{IOUT2}$	G5	131
$\overline{CS1}$	F1	132
$\overline{CS2}$	F4	134
$\overline{CS3}$	E1	135

Pinout for 169-pin PGA on HP 68EN302 Emulator Probe		
68EN302 Microprocessor Signal Name	**Emulator PGA Pin Number**	**68EN302 TQFP Pin Number**
D0	M9	79
D1	L9	80
D2	N8	81
D3	M8	82
D4	J8	84
D5	M7	85
D6	L7	86
D7	K9	87
D8	N7	89
D9	L8	90
D10	K7	91
D11	M6	92
D12	L5	94
D13	J7	95
D14	N6	96
D15	L6	97
PA0/RXD2	J10	65
PA1/TXD2	J12	64
PA2/RCLK2	J13	63
PA3/$\overline{\text{TCLK2}}$	H10	62
PA4/$\overline{\text{CTS2}}$	H13	60
PA5/$\overline{\text{RTS2}}$	G9	59
PA6/$\overline{\text{CD2}}$	J11	58
PA7/BRG2/SDS2/$\overline{\text{RAS0}}$	E10	51
PA8/RXD3	H12	56
PA9/TXD3	G10	55
PA10/RCLK3	F11	54
PA11/TCLK3	G13	53
PA12/$\overline{\text{BRG3}}$/$\overline{\text{RAS1}}$	G11	50
PA13/$\overline{\text{DREQ}}$/WEL	F10	47
PA14/$\overline{\text{DACK}}$/$\overline{\text{WEH}}$	F9	48
PA15/$\overline{\text{DONE}}$/$\overline{\text{OE}}$	G12	49

Pinout for 169-pin PGA on HP 68EN302 Emulator Probe		
68EN302 Microprocessor Signal Name	**Emulator PGA Pin Number**	**68EN302 TQFP Pin Number**
PB0/IACK7/CAS0	B5	7
PB1/IACK6/CAS1	C4	6
PB2/IACK1/DRAMRW	B4	5
PB3/TIN1	A4	4
PB4/TOUT1/A0	A3	2
PB5/TIN2	A2	1
PB6/TOUT2	A1	144
PB7/WDOG	C2	142
PB8	C1	141
PB9	D3	140
PB10	D2	139
PB11	D1	138
RXD1/L1RXD	K13	66
TXD1/L1TXD	A13	36
RCLK1/L1CLK	B11	34
TCLK1/L1SY0/SDS1	A12	35
CD1/L1SY1	K11	68
CTS1/L1GR	K12	67
RTS1/L1RQ/GCIDCL	B13	37
BRG1/AMUX	D13	40
SPRXD/CTS3	L13	69
SPTXD/RTS3	C13	38
SPCLK/CD3	C12	39
IPL0/IRQ1	C6	18
IPL1/IRQ6	D7	19
IPL2/IRQ7	B8	20
BUSW/PARITY1	D11	42
DISCPU/PARITY0	E12	43
RESET	E7	23
HALT	A8	24
DTACK	B10	31
BERR	E8	21
PARITYE/THREES	F12	46

Pinout for 169-pin PGA on HP 68EN302 Emulator Probe		
68EN302 Microprocessor Signal Name	**Emulator PGA Pin Number**	**68EN302 TQFP Pin Number**
TCLK RX TENA CLSN RENA RCLK TX	L10 L12 M10 M11 M13 N10 N13	78 70 77 75 71 76 72
$\overline{\text{AS}}$ $\overline{\text{UDS}}$ $\overline{\text{LDS}}$ R/$\overline{\text{W}}$	D6 A6 B6 E6	11 9 10 12
$\overline{\text{BR}}$ $\overline{\text{BG}}$ $\overline{\text{BGACK}}$	C8 A9 D8	25 27 26
AVEC	C9	22
EXTAL CLKO	D5 A7	15 17
XTAL (no connect)	C7	14
TRST (no connect) TMS (no connect TCK (no connect) TDO (no connect) TDI (no connect)	A10 B9 D9 E2 E4	30 28 29 136 137

Pinout for 169-pin PGA on HP 68EN302 Emulator Probe		
68EN302 Microprocessor Signal Name	**Emulator PGA Pin Number**	**68EN302 TQFP Pin Number**
V_{DD}	A5	16
V_{DD}	A11	33
V_{DD}	B3	3
V_{DD}	E11	44
V_{DD}	F2	128
V_{DD}	H9	57
V_{DD}	M1	109
V_{DD}	N5	99
V_{DD}	N9	88
V_{DD}	N12	73
GND (V_{SS})	B1	143
GND (V_{SS})	B7	13
GND (V_{SS})	C5	8
GND (V_{SS})	C10	32
GND (V_{SS})	D12	41
GND (V_{SS})	E13	52
GND (V_{SS})	F3	133
GND (V_{SS})	F13	45
GND (V_{SS})	H11	61
GND (V_{SS})	J4	123
GND (V_{SS})	J6	93
GND (V_{SS})	K3	114
GND (V_{SS})	K6	98
GND (V_{SS})	K8	83
GND (V_{SS})	L4	104
GND (V_{SS})	N11	74
Cable Ground	B2, B12, C3, C11, D4, D10, E5, E9, F6, F7, F8, G6, G7, G8, H6, H7, H8, J5, J9, K4, K10, L3, L11, M2, M12	

10

MC68LC302 Specifications and Characteristics

Processor compatibility

The HP 64798 Emulator supports MC6830x microprocessors. The emulator supports MC68LC302 microprocessors operating at 25 MHz at 5 volts for the 100 TQFP using optional accessories.

Electrical

Clock speed

The maximum clock speed of the MC68LC302 emulator is 25 MHz with no wait states required for emulation or target memory. The minimum clock speed required is 0 MHz. These clock speeds are supported when using any of the following emulator SIMM types:

- 64171A 256-Kbyte 35-ns SIMM
- 64171B 1-Mbyte 35-ns SIMM
- 64172A 256-Kbyte 20-ns SIMM
- 64172B 1-Mbyte 20-ns SIMM
- 64173A 4-Mbyte 25-ns SIMM

HP 64798F maximum ratings

Characteristic	Symbol	Value	Unit
Supply Voltage	V_{CC}	–0.3 to +5.5	V
Input Voltage	V_{in}	–0.3 to +5.5	V
Maximum Operating Ambient Temperature	T_A	40	deg C
Minimum Operating Ambient Temperature	T_A	0	deg C
Storage Temperature Range	T_{stg}	–40 to +70	deg C

HP 64798F DC electrical characteristics

Characteristic	Signal	Symbol	64798F Emulator		Unit
			Min	Max	
Input High Voltage	All Except pins noted below	V_{IH}	2.0	V_{DD}	V
Input high voltage	$\overline{CD1}$, $\overline{CTS1}$, RXD1, TXD1, RCLK1, $\overline{RTS1}$, TCLK1, PA7-PA10, PA12, $\overline{CD2}$, RXD2, TXD2, RCLK2, $\overline{RTS2}$, TCLK2, TIN1, TIN2, TOUT2, WDOG, PB8-PB11, $\overline{RESET}$	V_{IH}	2.5	V_{DD}	V
Input Low Voltage	All Except EXTAL	V_{IL}	V_{SS} -0.3	0.8	V
Input High Voltage	EXTAL --- 3.3V or 5V part	V_{CIH}	0.8*V_{DD}	V_{DD}	V
Input Low Voltage	EXTAL	V_{CIL}	V_{SS} -0.3	0.6	V
Output High Voltage (I_{OH} = 400 uA)	$\overline{AS}$, $\overline{CS0}$-$\overline{CS3}$, $\overline{WEH}$, $\overline{WEL}$, $\overline{OE}$ (Unbuffered)	V_{OH}	V_{DD} -1.0	-	V
	$\overline{AS}$, $\overline{CS0}$-$\overline{CS3}$, $\overline{WEH}$, $\overline{WEL}$, $\overline{OE}$ (Buffered)		3.5	-	
	CLKO		VDD -1.0	-	
	All Other Outputs		VDD -1.0	-	
Output Low Voltage (I_{OL} = 3.2 mA)	A1-A19, PB3-PB11, $\overline{CS0}$-$\overline{CS3}$, $\overline{BG}$, RCLK1, RCLK2, TCLK1, TCLK2, $\overline{RTS1}$, $\overline{RTS2}$, SDS2, PA12, RXD2, $\overline{CTS2}$, $\overline{CD2}$	V_{OL}	-	0.5	V
	CLKO		-	0.4	

Characteristic	Signal	Symbol	64798F Emulator		Unit
			Min	Max	
Output Low Voltage (I_{OL} = 5.3 mA)	D0-D15, $\overline{AS}$, $\overline{WEH}$, $\overline{WEL}$, $\overline{OE}$, $\overline{BGACK}$, DTACK, RESET	V_{OL}	-	0.5	V
Output Low Voltage (I_{OL} = 7.0 mA)	TXD1, TXD2	V_{OL}	-	0.5	V
Output Low Voltage (I_{OL} = 8.9 mA)	$\overline{HALT}$, $\overline{BR}$ (as output)	V_{OL}	-	0.5	V
Input Low Current (V_{IL} = 0V) Note 4	A1	I_{IL}	-	0.4	mA
	A2, A3, $\overline{IPL0}$-$\overline{IPL2}$, BUSW, DISCPU		-	0.2	
	A4-A23, D0-D15		-	70	uA
	$\overline{AS}$		-	1.0	mA
	$\overline{WEH}$, $\overline{WEL}$, $\overline{OE}$		-	0.6	
	$\overline{RESET}$, $\overline{HALT}$, $\overline{DTACK}$		-	0.7	
	All Other Signals		-	20	uA
Input High Current (V_{IH} = V_{DD}) Note 4	A1, $\overline{AS}$, $\overline{WEH}$, $\overline{WEL}$, $\overline{OE}$	I_{IH}	-	60	uA
	A2, A3, $\overline{RESET}$, $\overline{HALT}$, $\overline{DTACK}$,		-	30	
	All Other Signals		-	20	

Characteristic	Signal	Symbol	64798F Emulator		Unit
			Min	Max	
Input Capacitance (With 144-pin TQFP or 132-pin PQFP adaptor cable)	A1-A3	C_{IN}	-	70	pF
	A4-A19		-	60	
	D0-D15		-	90	
	EXTAL Note 1		-	50	
	$\overline{\text{RESET}}$, $\overline{\text{HALT}}$, $\overline{\text{AS}}$ Note 1		-	105	
	$\overline{\text{WEH}}$, $\overline{\text{WEL}}$, $\overline{\text{OE}}$		-	75	
	$\overline{\text{IPL0}}$-$\overline{\text{IPL2}}$		-	55	
	BUSW, DISCPU, $\overline{\text{DTACK}}$		-	65	
	PB3-PB11		-	45	
	PA0-PA10, RXD1, RCLK1, TCLK1, $\overline{\text{CD1}}$, $\overline{\text{CTS1}}$		-	40	

Characteristic	Signal	Symbol	64798F Emulator		Unit
			Min	Max	
Load Capacitance (Additional capacitance added to processor outputs by emulator with 144-pin TQFP or 132-pin PQFP adaptor cable)	A1-A3	C_{EL}	-	60	pF
	A4-A19		-	50	
	D0-D15 Note 3		-	80	
	$\overline{CS0}$-$\overline{CS3}$ Note 3		-	60	
	$\overline{RESET}$, $\overline{HALT}$, $\overline{AS}$ Notes 1, 3		-	95	
	$\overline{WEH}$, $\overline{WEL}$, $\overline{OE}$ Note 3		-	65	
	$\overline{DTACK}$ Note 3		-	55	
	PB3-PB11		-	35	
	PA0-PA10, TXD1, RCLK1, TCLK1, $\overline{RTS1}$, SPCLK		-	30	
	CLKO (at processor) Note 2		-	50	
Power		V_{DD}	4.5	5.5	V
Common		V_{SS}	0	0	V
Power Supply Current Drawn from Target System		I_{DD}	-	200	mA

Notes

1 EXTAL and $\overline{AS}$ are additionally terminated with 100 ohms in series with 100 pF on the emulator.
2 CLKO is buffered before it is sent to the target.
3 $\overline{CS0}$-$\overline{CS2}$, $\overline{AS}$, $\overline{WEH}$, $\overline{WEL}$, and $\overline{OE}$ unbuffered.
4 The Three-State Leakage Current (I_{TSI}) specification for those signals that can be outputs is the same as specified for Input Current at the given input voltage.

HP 64798F AC Electrical Specifications

For the following tables *Buffered* and *Unbuffered* refer to whether or not the various processor control signals are buffered by the emulator or not. There are three configuration items that control buffering for the following three groups of signals:

- Buffer Strobes $\overline{AS}$, $\overline{OE}$
- Buffer Write Enables $\overline{WEH}$, $\overline{WEL}$
- Buffer Chip Selects $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$

The *All Unbuffered* column shows the emulator timing when all three configuration items are set to "unbuffered". The *All Buffered* column shows the emulator timing when all three configuration items are set to "buffered".

If some, but not all three configuration items are set to buffered, determine for each of the two signals in the timing relationship if it is buffered or unbuffered. If both are unbuffered, then use the *All Unbuffered* column, if both are buffered use the *All Buffered* column. If the first signal is buffered but the second is unbuffered, use the *Buffered / Unbuffered* column, and likewise if the first signal is unbuffered and the second signal is buffered use the *Unbuffered / Buffered* column. Footnotes help to make it clear which signal is buffered and which is unbuffered.

If neither or only one of the two signals in a given timing relationship can be configured for buffering/unbuffering, then the *Buffered / Unbuffered* and *Unbuffered / Buffered* columns are not used. In the case where just one signal can be buffered, use the *All Unbuffered* column if the signal is unbuffered, and the *All Buffered* column if the signal is buffered.

The timing specifications for some of the peripherals on the processor are not included here since they are unaffected by the emulator logic.

The subscript numbers in the *Characteristic* column refer to the notes in the Motorola data book.

Clock Timing

Num	Characteristic	Motorola 25 MHz 68LC302		64798F Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
	Frequency of Operation	dc	25	dc	25	dc	25					MHz
1	CLK0 Period (EXTAL)	40		40		40						ns
4,5	Clock Rise and Fall Times (EXTAL)	-	4	-	4	-	4					ns
5B	EXTAL to CLKO Delay 1,2	2	7	3.5 Note	12 Note	3.5 Note	12 Note					ns

Note

CLKO skew from the rising and falling edges of EXTAL will not differ from each other by more than 2.5 ns, if the EXTAL rise time equals the EXTAL fall time.

IMP Bus Master Cycles

Num	Characteristic	Motorola 25 MHz 68LC302		64798F Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
6	Clock High to Address Valid	0	30	-5	28.5	-5	28.5					ns
7	Clock High to Address, Data Bus High Impedance (Maximum)	-	33	-	31.5	-	31.5					ns
8	Clock High to Address Invalid	0	-	-5	-	-5	-					ns
9	Clock High to $\overline{AS}$, $\overline{DS}$ Asserted [1]	3	20	-2	19	-1	26					ns
11	Address Valid to $\overline{AS}$, $\overline{DS}$ Asserted [2]	10	-	10	-	11	-					ns
12	Clock Low to $\overline{AS}$, $\overline{DS}$ Negated [1]	-	20	-	19	-	26					ns
13	$\overline{AS}$, $\overline{DS}$ Negated to Address Invalid [2]	10	-	10	-	2.5	-					ns
14	$\overline{AS}$ (and $\overline{DS}$ Read) Width Asserted [2]	80	-	80	-	77	-					ns
14A	$\overline{DS}$ Width Asserted (Write) [2]	40	-	40	-	37	-					ns
15	$\overline{AS}$, $\overline{DS}$ Width Negated [2]	40	-	40	-	37	-					ns

Num	Characteristic	Motorola 25 MHz 68LC302		64798F Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
16	Clock High to Control Bus High Impedance	-	33	-	33	-	35 Note 2					ns
17	$\overline{AS}$, $\overline{DS}$ Negated to R/$\overline{W}$ Invalid 2	10	-	10	-	3.5	-	2.5$_c$	-	10.5$_d$	-	ns
18	Clock High to R/$\overline{W}$ High 1	-	20	-	19	-	26					ns
20	Clock High to R/$\overline{W}$ Low 1	-	20	-	19	-	26					ns
20A	$\overline{AS}$ Asserted to R/$\overline{W}$ Low (Write) 2,6	-	7	-	7	-	13.5	-	6.5$_c$	-	14.5$_d$	ns
21	Address Valid to R/$\overline{W}$ Low (Write) 2	10	-	10	-	11						ns
22	R/$\overline{W}$ Low to $\overline{DS}$ Asserted (Write) 2	20	-	20	-	16.5	-	15.5$_d$	-	21$_c$	-	ns
23	Clock Low to Data-Out Valid	-	20	-	19	-	19					ns
25	$\overline{AS}$, $\overline{DS}$ Negated to Data-Out Invalid (Write) 2	10	-	10	-	2.5	-					ns
26	Data-Out Invalid to $\overline{DS}$ Asserted (Write) 2	10	-	10	-	10.5	-					ns
27	Data-In Valid to Clock Low (Setup Time on Read) 5	5	-	10	-	10	-					ns

Num	Characteristic	Motorola 25 MHz 68LC302		64798F Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
28	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{DTACK}$ Negated (Asynchronous Hold) 2	0	75	0	75	-1	67.5					ns
29	$\overline{AS}$, $\overline{DS}$ Negated to Data-In Invalid (Hold Time on Read)	0 Note 1	-	0	-	-1	-					ns
31	$\overline{DTACK}$ Asserted to Data-In Valid (Setup Time) 2,5	-	33	-	33	-	33					ns
32	$\overline{HALT}$ and $\overline{RESET}$ Input Transition Time	-	150	-	150	-	150					ns
47	Asynchronous Input. Setup Time to Clock 5	7	-	12	-	12	-					ns
	Asynchronous Input ($\overline{IPL0}$-$\overline{IPL2}$). Setup Time to Clock 5	7	-	30	-	30	-					ns
53	Clock High to Data-Out Invalid (Hold Time on Write)	0	-	-5	-	-5	-					ns
55	R/$\overline{W}$ Asserted to Data Bus Impedance Change	0	-	0	-	-7.5	-					ns

Num	Characteristic	Motorola 25 MHz 68LC302		64798F Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
56	$\overline{\text{HALT}}$/$\overline{\text{RESET}}$ Pulse Width [4]	10	-	10	-	10	-					clks

Notes

[1] Motorola data sheet does not indicate a value for 25 MHz but shows a value of 0 for 16.67 MHz and 20 MHz.
[2] Actual value depends on the 1/2-clock period. The value given is based on a 1/2-clock period of 20 ns. To calculate this value, add 15 ns to the 1/2-clock period.
[3] If bus arbitration occurs while in the background monitor, this value is 0.5 clks.

Notes for *Buffered / Unbuffered* and *Unbuffered / Buffered* columns:

[c] $\overline{\text{AS}}$, $\overline{\text{DS}}$, $\overline{\text{IACK7}}$ buffered; R/$\overline{\text{W}}$ unbuffered.
[d] R/$\overline{\text{W}}$ buffered; $\overline{\text{AS}}$, $\overline{\text{DS}}$, $\overline{\text{IACK7}}$ unbuffered.

DMA

Num	Characteristic	Motorola 25 MHz 68LC302		64798F Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
83	Clock High to $\overline{BR}$ Low 3,4	-	20	-	18.5	-	18.5					ns
84	Clock High to $\overline{BR}$ High Impedance 3,4	-	20	-	18.5	-	18.5					ns
85	$\overline{BGACK}$ Low to $\overline{BR}$ High Impedance 3,4	20	-	20	-	20	-					ns
86	Clock High to $\overline{BGACK}$ Low	-	20	-	18.5	-	18.5					ns
87	$\overline{AS}$ and $\overline{BGACK}$ High (The Last One) to $\overline{BGACK}$ Low (When $\overline{BG}$ is Asserted)	1.5	2.5 +20	1.5	2.5 +20	1.5	2.5 +20					ns clks
88	$\overline{BG}$ Low to $\overline{BGACK}$ Low (No Other Bus Master) 3,4	1.5	2.5 +20	1.5	2.5 +20	1.5	2.5 +20					ns clks
89	$\overline{BR}$ High Impedance to $\overline{BG}$ High 3,4	0	-	0	-	0	-					ns
90	Clock on which $\overline{BGACK}$ Low to Clock on which $\overline{AS}$ Low	2	2	2	2	2	2					clks

Num	Characteristic	Motorola 25 MHz 68LC302		64798F Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
91	Clock High to $\overline{\text{BGACK}}$ High	-	20	-	18.5	-	18.5					ns
92	Clock Low to $\overline{\text{BGACK}}$ High Impedance	-	10	-	8.5	-	8.5					ns

External Master Internal Asynchronous Read/Write Cycles

Num	Characteristic	Motorola 25 MHz 68LC302		64798F Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
100	R/$\overline{W}$ Valid to $\overline{DS}$ Low	0	-	0	-	0	-					ns
101	$\overline{DS}$ Low to Data-In Valid	-	20	-	20	-	20					ns
102	$\overline{DTACK}$ Low to Data-In Hold Time	0	-	0	-	0	-					ns
103	$\overline{AS}$ Valid to $\overline{DS}$ Low	0	-	0	-	0	-					ns
104	$\overline{DTACK}$ Low to $\overline{AS}$, $\overline{DS}$ High	0	-	0	-	0	-					ns
105	$\overline{DS}$ High to $\overline{DTACK}$ High	-	30	-	30	-	30					ns
106	$\overline{DS}$ Inactive to $\overline{AS}$ Inactive	0	-	0	-	0	-					ns
107	$\overline{DS}$ High to R/$\overline{W}$ High	0	-	0	-	0	-					ns
108	$\overline{DS}$ High to Data High Impedance	-	30	-	30	-	30					ns
108A	$\overline{DS}$ High to Data-Out Hold Time	0	-	0	-	0	-					ns
109A	Data-Out Valid to $\overline{DTACK}$ Low	10	-	10	-	10	-					ns

External Master Internal Synchronous Read/Write Cycles

Num	Characteristic	Motorola 25 MHz 68LC302		64798F Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
110	Address Valid to $\overline{AS}$ Low	10	-	10	-	10	-					ns
111	$\overline{AS}$ Low to Clock High	20	-	25	-	25	-					ns
112	Clock Low to $\overline{AS}$ High	-	30	-	25	-	25					ns
113	$\overline{AS}$ High to Address Hold Time (Write)	0	-	0	-	0	-					ns
114	$\overline{AS}$ Inactive Time	1	-	1	-	1	-					clks
115	$\overline{DS}$ Low to Clock High 2	27	-	32	-	32	-					ns
116	Clock Low to $\overline{DS}$ High	-	30	-	25	-	25					ns
117	R/$\overline{W}$ Valid to Clock High 2	20	-	25	-	25	-					ns
118	Clock High to R/$\overline{W}$ High	-	30	-	25	-	25					ns
121	$\overline{AS}$ Low to $\overline{DTACK}$ Low (0 Wait States)	-	30	-	30	-	30					ns
122	Clock Low to $\overline{DTACK}$ Low (1 Wait State)	-	20	-	18.5	-	18.5					ns

Num	Characteristic	Motorola 25 MHz 68LC302		64798F Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
123	$\overline{AS}$ High to $\overline{DTACK}$ High	-	30	-	30	-	30					ns
124	$\overline{DTACK}$ High to DTACK High Impedance	-	10	-	10	-	10					ns
125	Clock High to Data-Out Valid	-	20	-	18.5	-	18.5					ns
126	$\overline{AS}$ High to Data High Impedance	-	30	-	30	-	30					ns
127	$\overline{AS}$ High to Data-Out Hold	0	-	0	-	0	-					ns
128	$\overline{AS}$ High to Address Hold Time (Read)	0	-	0	-	0	-					ns
129	$\overline{DS}$ Inactive Time	1	-	1	-	1	-					clks
130	Data-In Valid to Clock Low	20	-	25	-	25	-					ns
131	Clock Low to Data-In Hold Time	10	-	8.5	-	8.5	-					ns

Chip Select Timing Internal Master

Num	Characteristic	Motorola 25 MHz 68LC302		64798F Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
150	Clock High to $\overline{CS}$, $\overline{IACK}$, $\overline{OE}$, $\overline{WEL}$, $\overline{WEH}$ Low [2]	0	27	-5	26	-4	33					ns
151	Clock Low to $\overline{CS}$, $\overline{IACK}$, $\overline{OE}$, $\overline{WEL}$, $\overline{WEH}$ High [2]	0	27	-5	26	-4	33					ns
152	$\overline{CS}$ Width Negated	40	-	40	-	37	-					ns
153	Clock High to $\overline{DTACK}$ Low (0 Wait States)	-	30	-	28.5	-	28.5					ns
154	Clock High to $\overline{DTACK}$ Low (1-6 Wait States)	-	20	-	18.5	-	18.5					ns
155	Clock Low to $\overline{DTACK}$ High	-	27	-	25.5	-	25.5					ns
158	$\overline{DTACK}$ High to $\overline{DTACK}$ High Impedance	-	10	-	10	-	10					ns
171	Clock Low (end of S6) to Data-In Invalid (Hold Time on Read)	5	-	3.5	-	3.5	-					ns
172	$\overline{CS}$ Negated to Data-Out Invalid (Write)	10	-	10	-	-0.5	-					ns

Num	Characteristic	Motorola 25 MHz 68LC302		64798F Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
173	Address Valid to $\overline{CS}$ Asserted	5	-	5	-	6	-					ns
174	$\overline{CS}$ Negated to Address Invalid	7	-	7	-	4.5	-					ns
175	$\overline{CS}$ Low Time (0 Wait States)	80	-	80	-	77.5	-					ns
176	$\overline{CS}$ Negated to R/$\overline{W}$ Invalid	12	-	12	-	0.5	-	-0.5[i]	-	12[j]	-	ns
177	$\overline{CS}$ Negated to R/$\overline{W}$ Low (Write)	-	8	-	8	-	14.5	-	7.5[i]	-	15.5[j]	ns
178	$\overline{CS}$ Negated to Data-In Invalid (Hold Time on Read)	0	-	0	-	-7.5	-					ns

Notes for *Buffered / Unbuffered* and *Unbuffered / Buffered* columns:

[g] $\overline{CS}$ buffered; $\underline{FC}$ unbuffered..
[h] $\underline{FC}$ buffered; $\underline{CS}$ unbuffered.
[i] CS buffered; R/$\overline{W}$ unbuffered.
[j] R/$\overline{W}$ buffered; $\overline{CS}$ unbuffered.

Chip Select Timing External Master

Num	Characteristic	Motorola 25 MHz 68LC302		64798F Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
154	Clock High to $\overline{\text{DTACK}}$ Low (1-6 Wait States)	-	20	-	18.5	-	18.5					ns
160	$\overline{\text{AS}}$ Low to $\overline{\text{CS}}$ Low	-	20	-	20	-	20					ns
161	$\overline{\text{AS}}$ High to $\overline{\text{CS}}$ High	-	20	-	20	-	20					ns
162	Address Valid to $\overline{\text{AS}}$ Low	10	-	10	-	10	-					ns
164	$\overline{\text{AS}}$ Negated to Address Hold Time	0	-	0	-	0	-					ns
165	$\overline{\text{AS}}$ Low to $\overline{\text{DTACK}}$ Low (0 Wait States)	-	30	-	30	-	30					ns
167	$\overline{\text{AS}}$ High to $\overline{\text{DTACK}}$ High	-	20	-	20	-	20					ns

Parallel I/O

Num	Characteristic	Motorola 25 MHz 68LC302		64798F Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
180	Input Data Setup Time to Clock Low	14	-	19	-	19	-					ns
181	Clock Low to Input Data Hold Time	19	-	17.5	-	17.5	-					ns
182	Clock High to Data-Out Valid (CPU Writes Data, Control, or Direction)	-	24	-	22.5	-	22.5					ns

Interrupts

Num	Characteristic	Motorola 25 MHz 68LC302		64798F Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
190	Interrupt Pulse Width Low $\overline{\text{IRQ}}$ (Edge Triggered Mode)	34	-	34	-	34	-					ns
191	Minimum Time Between Active Edges	3		3		3						clks

Timers

Num	Characteristic	Motorola 25 MHz 68LC302		64798F Emulator								Unit
				All Unbuf.		All Buffered		Buffered / Unbuf.		Unbuf. / Buffered		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
200	Timer Input Capture Pulse Width	34	-	34	-	34	-					ns
201	TIN Clock Low Pulse Width	34	-	34	-	34	-					ns
202	TIN Clock High Pulse Width and Input Capture High Pulse Width	2	-	2	-	2	-					clks
203	TIN Clock Cycle Time	3		3	-	3	-					clks
204	Clock High to TOUT Valid	-	24	-	22.5	-	22.5					ns
205	$\overline{\text{FRZ}}$ Input Setup Time to Clock High [1]	14	-	29	-	29	-					ns
206	Clock High to $\overline{\text{FRZ}}$ Input Hold Time	7	-	4.5	-	4.5	-					ns

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.)

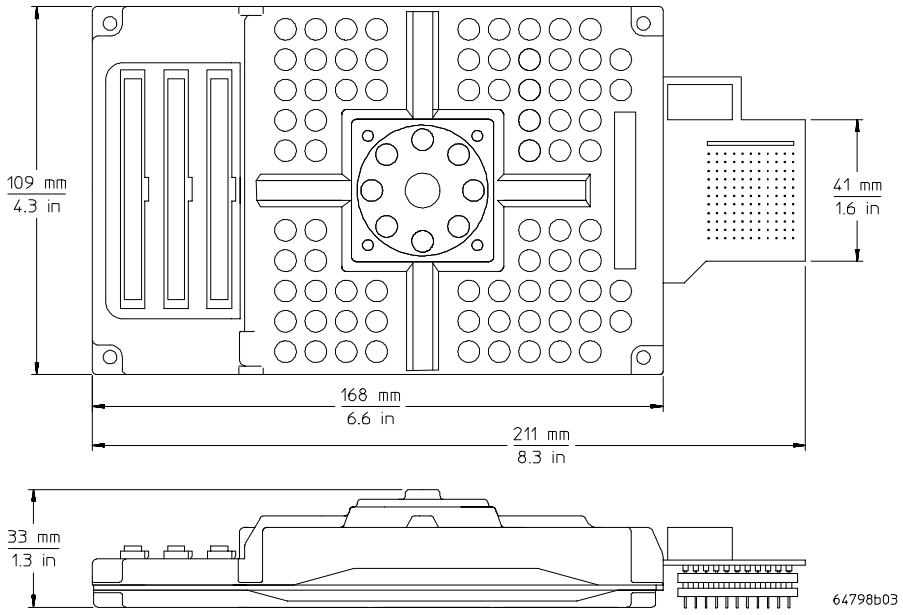
Emulator Weight

Probe alone: 0.3 kg (10 oz).

Cable Length

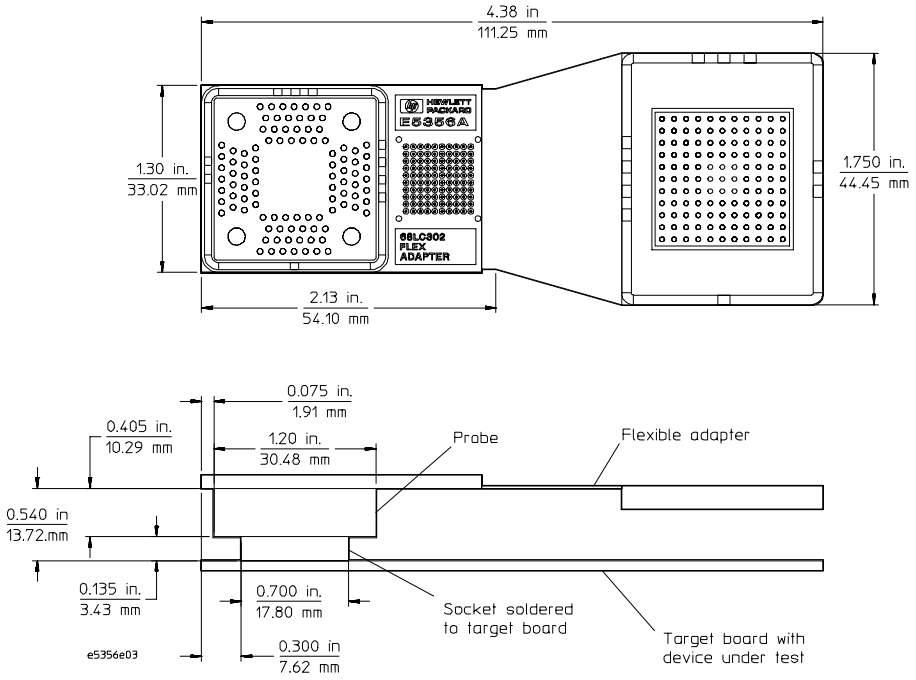
Emulation Control Card to Probe, approximately 914 mm (36 inches)

Probe dimensions



Keep-out area

You must solder the socket to your target board and install the target microprocessor in the socket before connecting the probe and flexible adapter assembly.



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

Input

74HCT132 with 135 ohms to ground in parallel.

Maximum input:

5V above Vcc; 5V 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.

Pinout for 121-pin PGA on HP 68LC302 Emulator Probe		
68LC302 Microprocessor Signal Name	**Emulator PGA Pin Number**	**68LC302 TQFP Pin Number**
A1	E2	93
A2	E1	92
A3	F4	91
A4	F3	90
A5	F2	89
A6	G3	87
A7	G2	86
A8	G1	85
A9	H3	84
A10	H1	82
A11	J3	81
A12	J2	80
A13	J1	79
A14	K2	78
A15	K1	77
A16	L2	75
A17	K3	74
A18	L3	73
A19	J4	72
$\overline{CS0}/\overline{IOUT2}$	E3	94
$\overline{CS1}$	D1	95
$\overline{CS2}$	D2	96
$\overline{CS3}$	D3	97

Pinout for 121-pin PGA on HP 68LC302 Emulator Probe		
68LC302 Microprocessor Signal Name	**Emulator PGA Pin Number**	**68LC302 TQFP Pin Number**
D0	L11	51
D1	L10	52
D2	K10	53
D3	L9	54
D4	J9	56
D5	L8	57
D6	K8	58
D7	J8	59
D8	L6	63
D9	K6	64
D10	J6	65
D11	H6	66
D12	K5	68
D13	J5	69
D14	L4	70
D15	K4	71
RXD2/PA0	H11	45
TXD2/PA1	G9	44
RCLK2/PA2	G10	43
$\overline{\text{TCLK2}}$/PA3	F8	41
$\overline{\text{CTS2}}$/PA4	F9	40
$\overline{\text{RTS2}}$/PA5	F10	39
$\overline{\text{CD2}}$/PA6	F11	38
BRG2/SDS2/PA7	E9	37
SPRD/PA8	H9	47
SPTXD/PA9	H10	46
SPCLK/PA10	C10	30
RXD1/L1RXD	J11	48
TXD1/L1TXD	D9	34
RCLK1/L1CLK	D11	32
TCL$\overline{\text{K1}}$/L1SY0/SDS1	D10	33
$\overline{\text{CD1}}$/L1SY1	K11	50
$\overline{\text{CTS1}}$/L1GR	J10	49
$\overline{\text{RTS1}}$/L1RQ/GCIDCL	C11	29

Pinout for 121-pin PGA on HP 68LC302 Emulator Probe		
68LC302 Microprocessor Signal Name	**Emulator PGA Pin Number**	**68LC302 TQFP Pin Number**
TIN1/PB3	C3	6
TIN2/PB5	B3	5
TOUT2/PB6	A3	4
WDOG/PB7	B2	3
PB8	A2	2
PB9	A1	1
PB10	C2	99
PB11	C1	98
IPL0/IRQ1	C6	15
IPL1/IRQ6	D6	16
IPL2/IRQ7	A7	17
BUSW	C8	22
DISCPU	B8	21
MODCLK/PA12	E11	35
VCCSYN	A10	25
RESET	B11	27
HALT	A11	26
DTACK	B10	28
AS	B5	11
WEH/A0	B4	8
WEL/WE	C4	9
OE	C5	12
EXTAL	B7	18
CLKO	A8	20
XTAL	-nc-	19
XFC	-nc-	24

Pinout for 121-pin PGA on HP 68LC302 Emulator Probe		
68LC302 Microprocessor Signal Name	**Emulator PGA Pin Number**	**68LC302 TQFP Pin Number**
V_{CC} V_{CC} V_{CC} V_{CC} V_{CC} V_{CC}	A4 A6 E10 L7 J7 H2	7 13 36 60 62 83
GNDSYN	A9	23
GND GND GND GND GND GND GND GND GND GND	A5 B6 C9 G11 K9 K7 L5 L1 F1 B1	10 14 31 42 55 61 67 76 88 100
Cable Ground	D4, D5, D7, D8, E4, E8, G4, G8, H4, H8	
Cable No Connect	E5, E6, E7, F5, F6, F7, G5, G6, G7	

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 move 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 reloaded 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.

DUT

Device Under Test (typically the target microprocessor).

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 special instruction placed in your software in RAM, replacing the normal instruction at the RAM address. Breakpoints for code in ROM are implemented by the Intel80386. When the breakpoint instruction 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.

PGA

Pin-Grid Array

Physical Address Space

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

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.

PQFP

Plastic Quad Flat Pack.

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.

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 (for example, 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

Thin Quad Flat Pack.

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 an exclamation mark (!) is shown in the trace list count column of the terminal interface or the PC interface, it 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: Emulator

Model Number(s): HP 64798C/E/F

Product Option(s): All

conforms to the following Product Specifications:

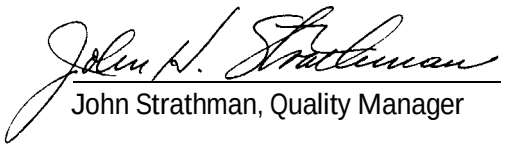
Safety:	IEC 1010-1:1990+A1 / EN 61010-1:1993	
	UL 3111	
	CSA-C22.2 No. 1010.1:1993	
EMC:	CISPR 11:1990 / EN 55011:1991	Group 1 Class A
	IEC 801-2:1991 / EN 50082-1:1992	4 kV CD, 8 kV AD
	IEC 801-3:1984 / EN 50082-1:1992	3 V/m, {1kHz 80% AM, 27-1000 MHz}
	IEC 801-4:1988 / EN 50082-1:1992	0.5 kV Sig. Lines,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

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

Colorado Springs, 4/16/96



John Strathman, Quality Manager

European Contact: Your local Hewlett-Packard Sales and Service Office or Hewlett-Packard GmbH, Department ZQ / Standards Europe, Herrenberger Strasse 130, D-71034 Böblingen Germany (FAX: +49-7031-14-3143)

Product Regulations

Safety

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

EMC

This Product meets the requirements 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) 8 kV AD	3	A, B, C
IEC 801-3 (Rad.) 3 V/m	1	C
IEC 801-4 (EFT) 1 kV	1	C

[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:
A Electrostatic discharge (ESD) to the 64700B mainframe may cause degraduation in performance requiring operator intervention.
B The active probe assembly is sensitive to ESD events. Use standard ESD preventative practices to avoid component damage.
C 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 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 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 you must follow 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 and Cautions, shown at the left-hand edges of pages in this user's guide, precede potentially dangerous procedures throughout this manual. Instructions contained in the warnings and cautions must be followed.

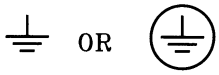
Safety Symbols Used In Manuals

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.

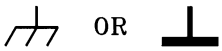
Hot Surface. This symbol means the part or surface is hot and should not be touched.

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.



Frame 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.