HEWLETT® PACKARD

User's Guide

HP E2465A PowerPC 604 PGA Preprocessor Interface

User's Guide

Publication Number E2465-97001
First Edition, October 1996

For Safety Information, Warranties, and Regulatory Information, see the pages at the end of this manual.

HP E2465A PowerPC 604 PGA Preprocessor Interface

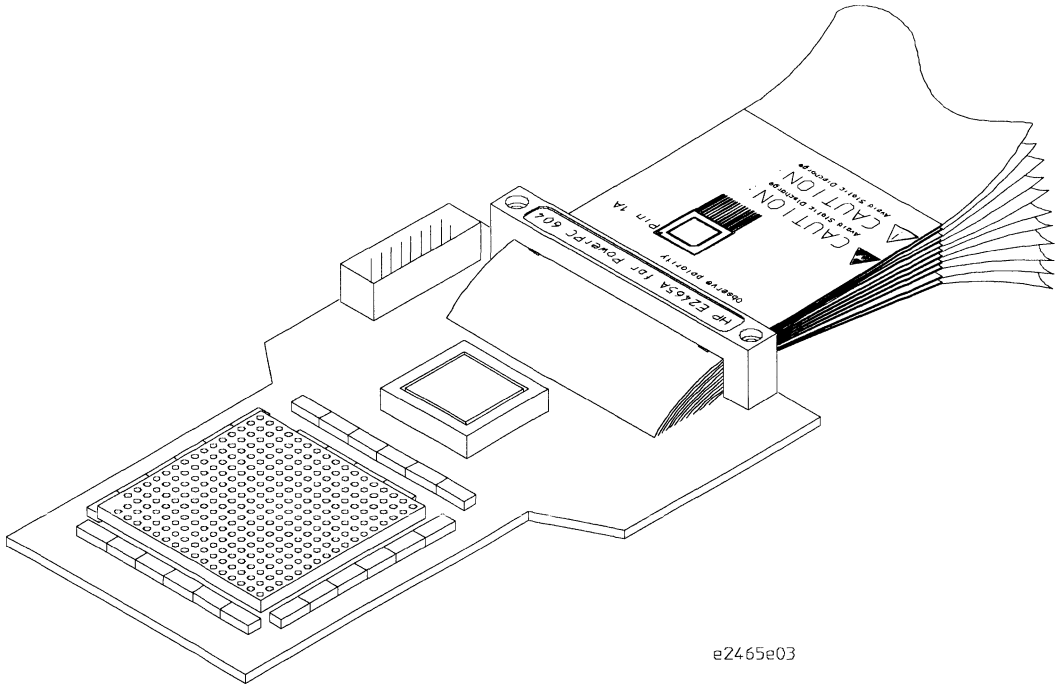
The HP E2465A Preprocessor Interface — At a Glance

The HP E2465A Preprocessor Interface provides a complete interface for state or timing analysis between any PowerPC 604 PGA target system and the following HP logic analyzers:

- HP 1660A/AS/C/CS
- HP 1670A/D
- HP 16550A (two cards)
- HP 16554A (two or three cards)
- HP 16555A/D (two or three cards)
- HP 16556A/D (two or three cards)

The preprocessor interface provides the physical connection between the target microprocessor and the logic analyzer. The configuration software on the flexible disk sets up the logic analyzer for compatibility with the preprocessor interface. The inverse assembler allows you to obtain displays of the microprocessor operations in PowerPC 604 mnemonics.

For additional information on the supported logic analyzers or microprocessors, refer to the appropriate reference manuals for those products.



HP E2465A Preprocessor Interface

In This Book

This book is the user's guide for the HP E2465A Preprocessor Interface. It assumes that you have a working knowledge of the logic analyzer used and the microprocessor being analyzed.

This user's guide is organized into three chapters:

Chapter 1 explains how to attach the preprocessor to the target system and logic analyzer, and how to load the preprocessor interface software into the logic analyzer.

Chapter 2 provides reference information on the format specification and symbols configured by the preprocessor interface software, and information about the inverse assemblers and status encoding.

Chapter 3 contains reference information on the preprocessor interface hardware, including the characteristics and signal mapping for the preprocessor interface.

Appendix A contains information on troubleshooting problems or difficulties which may occur with the preprocessor interface.

For more information on the logic analyzers or microprocessor, refer to the appropriate reference manual for those products.

Contents

2 Analyzing the PowerPC 604

3 Preprocessor Interface Hardware Reference

A If You Have a Problem

1

Setting Up the Preprocessor Interface

Setting Up the Preprocessor Interface

This chapter explains how to set up the HP E2465A Preprocessor Interface hardware and software, and how to connect the preprocessor to supported logic analyzers.

Before You Begin

This section lists the logic analyzers supported by the HP E2465A and provides other information about the analyzers and the preprocessor interface.

Equipment Supplied

- The preprocessor interface circuit board.
- The configuration and inverse assembler software on a 3.5-inch disk.
- Configuration software for the HP 16505A Prototype Analyzer on a 3.5-inch disk.
- A 287-pin PGA plastic pin protector/extender socket (HP part number 1200-1975).
- This User's Guide.

The preprocessor interface socket assembly pins are covered for shipment with a conductive foam wafer or conductive plastic pin protector. This is done to protect the delicate gold-plated pins from damage due to impact. When you're not using the preprocessor interface, protect the socket assembly pins from damage by covering them with the pin protector.

Minimum Equipment Required

- The HP E2465A preprocessor interface.
- The configuration and inverse assembler software.
- One of the logic analyzers listed in the table below. For the mainframe logic analyzers, an HP 16500B or HP 16500C mainframe is required.

  For the HP 16500B mainframe, software revision 3.4 or higher is recommended. For the HP 16500C mainframe, software revision 1.0 or higher is recommended.

Logic Analyzers Supported

Logic Analyzer	Channel Count	State Speed	Timing Speed	Memory Depth
1660A/AS/C/CS	136	100 MHz	250 MHz	4 k states
1670A	136	70 MHz	125 MHz	64 k or .5 M states
1670D	136	70 MHz	125 MHz	64 k or 1 M states
16550A (two cards)	204	100 MHz	250 MHz	4 k states
16554A (two or three cards)	68/card	70 MHz	125 MHz	512 k states
16555A (two or three cards)	68/card	110 MHz	250 MHz	1 M states
16555D (two or three cards)	68/card	110 MHz	250 MHz	2 M states
16556A (two or three cards)	68/card	100 MHz	200 MHz	1 M states
16556D (two or three cards)	68/card	100 MHz	200 MHz	2 M states

Optional	Software Version
HP 16505A Prototype Analyzer	A.01.22 or higher

Connecting to the Target System

Connecting to the target system consists of the following steps:

- Turn off the logic analyzer and the target system.
- Connect the preprocessor interface to the target system.

The remainder of this section describes these general steps in more detail.

To power up or power down

To protect your equipment, remove the power from both the logic analyzer and the target system before you make or break connections. The logic analyzer provides power to the circuits on the preprocessor interface, and should always be powered up before the target system. When powering down, power down the target system first and then power down the logic analyzer.

To connect the preprocessor to the target system

The microprocessor connector on the preprocessor interface will connect directly to the PGA socket on the target system. The PowerPC 604 microprocessor connects to a pin protector on top of the preprocessor interface.

1 To prevent equipment damage, remove power from both the logic analyzer and the target system.

2 Remove the 604 PGA microprocessor from its socket on the target system and store it in a protected environment.

3 Install the preprocessor interface into the PGA socket on the target system. Ensure that pin A1 is properly aligned (see figure on next page). The socket on the preprocessor interface is designed with low-insertion-force pins to allow insertion and removal with minimum force.

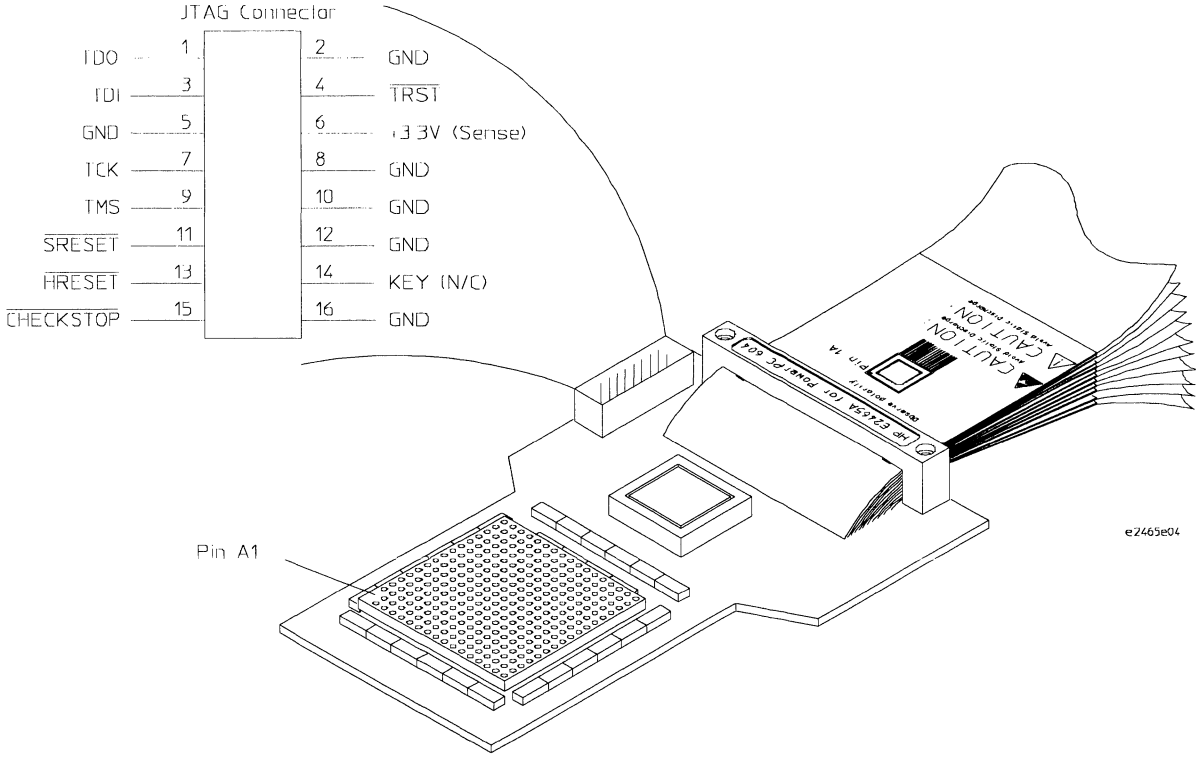
CAUTION

Serious damage to the target system or preprocessor interface can result from incorrect connection. Note the position of pin A1 on the target system, preprocessor interface, and microprocessor prior to making any connection. Also, take care to align the preprocessor interface connector pins with the target system PGA socket so that all pins are making contact.

If the preprocessor interface connector interferes with components of the target system or if a higher profile is required, additional plastic pin protector sockets can be added. Plastic pin protector sockets can be ordered from Hewlett-Packard using the part number 1200-1975. However, any 287-pin PGA IC socket with a 604 footprint and gold-plated pins can be used.

4 Plug the PowerPC 604 microprocessor into a pin protector, and plug that into the PGA socket on the preprocessor interface, again ensuring that pin A1 is properly aligned.

The PowerPC 604 microprocessor can be difficult to remove if you connect it directly to the preprocessor interface. Always use a pin protector between the microprocessor and the PGA socket on the preprocessor interface.



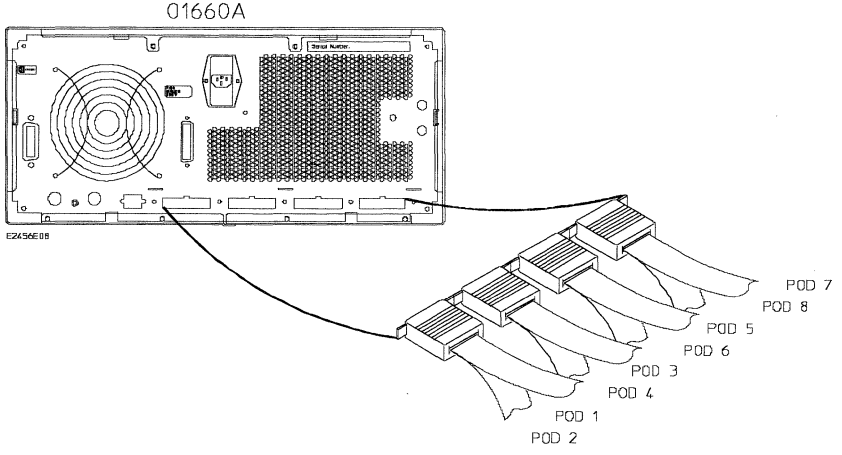
Pin A1 Location/Connecting to the Target System

Connecting to the Logic Analyzer

Use the tables and figures in the following sections to connect the logic analyzer pod cables to the flexible cable assemblies on the preprocessor interface. Refer to the pod diagram for the analyzer you are using. The configuration file for the logic analyzer is listed below the corresponding connection table.

To connect to the HP 1660A/AS/C/CS logic analyzers

Use the table below to connect the preprocessor to the HP 1660A/C logic analyzers.

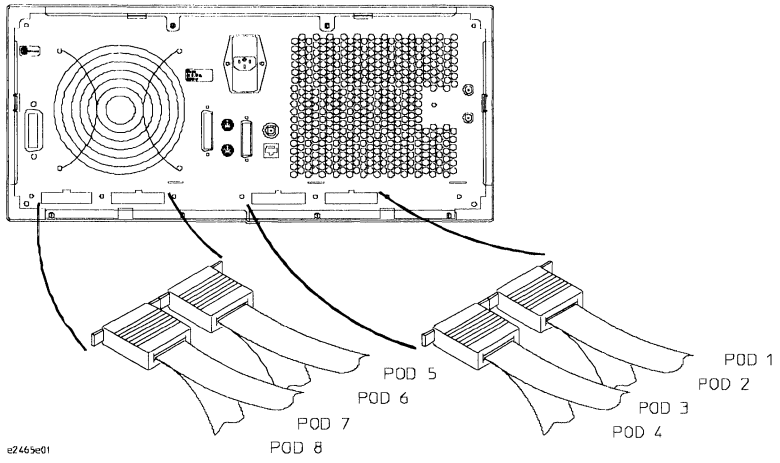


HP 1660A/C Pod	Pod 1	Pod 2	Pod 3	Pod 4	Pod 5	Pod 6	Pod 7	Pod 8
HP E2465A Connector	P8 STAT clk ↑	P7 STAT	P1 ADDR	P2 ADDR	P3 DATA_B	P4 DATA_B	P5 DATA	P6 DATA

Configuration File
Use configuration file C604J for the HP 1660A/AS/C/CS logic analyzers.

To connect to the HP 1670A/D logic analyzer

Use the table below to connect the preprocessor to the HP 1670A/D logic analyzers.

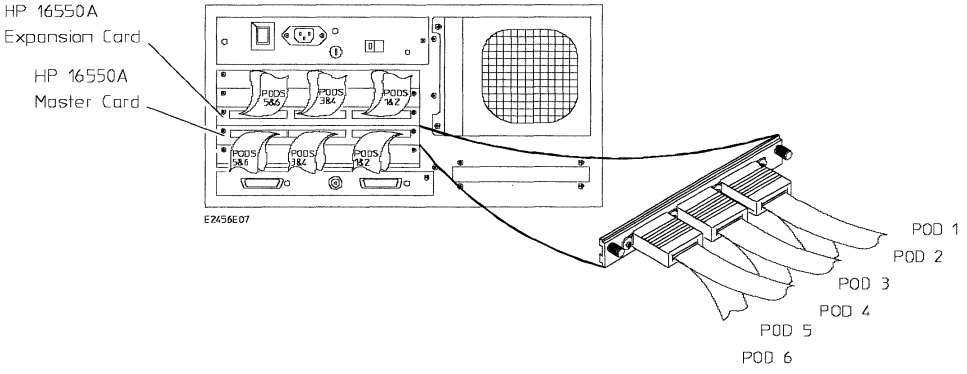


HP 1670A Pod	Pod 8	Pod 7	Pod 6	Pod 5	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	P6 DATA	P5 DATA	P4 DATA_B	P3 DATA_B	P2 ADDR	P1 ADDR	P7 STAT	P8 STAT clk ↑

Configuration File
Use configuration file C604M for the HP 1670A/D logic analyzers.

To connect to the HP 16550A two-card analyzer

Use the table below to connect the preprocessor to the HP 16550A logic analyzer.



HP 16550A Expansion Card Pod	Pod 6	Pod 5	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	P6 DATA	P5 DATA	P4 DATA_B	P3 DATA_B	P2 ADDR	P1 ADDR

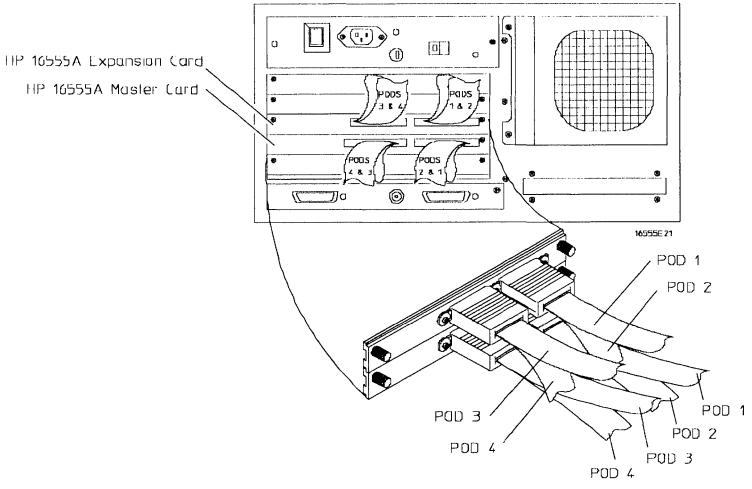
HP 16550A Master Card Pod	Pod 6	Pod 5	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	P7 STAT	P8 STAT clk ↑	P9 Parity	P10 JTAG	P11 PLL	--

Configuration File
Use configuration file C604F for the HP 16550A logic analyzer.

Second Analyzer Machine
The configuration file C604F can support a second analyzer machine. To configure a second machine, use the logic analyzer Configuration menu to reassign Master Card pods 1 through 4 to Analyzer 2.

To connect to the HP 16554/55/56 (two-card)

Use the table below to connect the preprocessor to the two-card HP 16554A/55A/56A or HP 16555D/56D logic analyzers.



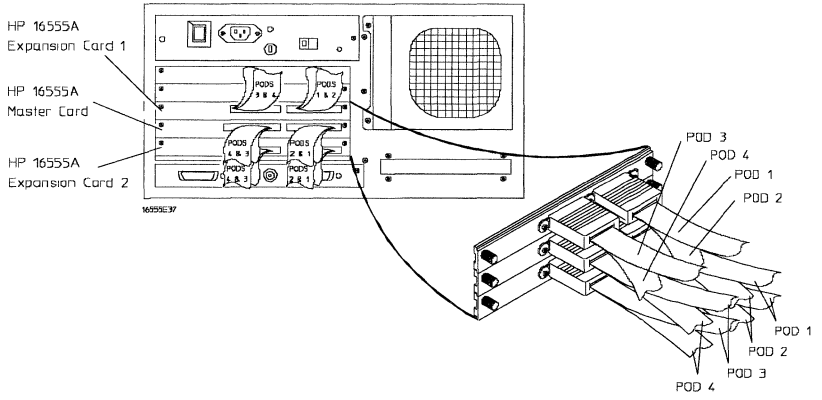
HP 16554A/55A/56A Expansion Card Pod	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	P6 DATA	P5 DATA	P4 DATA_B	P3 DATA_B

HP 16554A/55A/56A Master Card Pod	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	P2 ADDR	P1 ADDR	P7 STAT	P8 STAT clk ↑

Configuration File
Use configuration file C604M for the two-card HP 16554/55/56.

To connect to the HP 16554/55/56 (three-card)

Use the table below to connect the preprocessor to the three-card HP 16554A/55A/56A OR HP 16555D/56D logic analyzers.



Exp. Card 1 Pod	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	P6 DATA	P5 DATA	P4 DATA_B	P3 DATA_B
Master Card Pod	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	--	P11 PLL	P7 STAT	P8 STAT clk ↑
Exp. Card 2 Pod	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	P9 PARITY	P10 JTAG	P2 ADDR	P1 ADDR

Configuration File
Use configuration file C604M3 for the three-card HP 16554/55/56.

Second Analyzer Machine
The configuration file C604M3 can support a second analyzer machine. To configure a second machine, use the logic analyzer Configuration menu to reassign Master Card pods 3 and 4 and Expansion Card 2 pods 3 and 4 to Analyzer 2.

Configuring the Logic Analyzer

Configuring the logic analyzer consists of loading the software by inserting the floppy disk into the logic analyzer disk drive and loading the proper configuration file. The configuration file you use is determined by the logic analyzer you are using.

To load the configuration and inverse assembler

The first time you set up the preprocessor interface, make a duplicate copy of the master disk. For information on duplicating disks, refer to the reference manual for your logic analyzer.

For logic analyzers with a hard disk (HP 1670A/D, HP 1660C/CS, or an analyzer in the HP 16500B/C frame), you might want to create a directory such as PPC604 on the hard drive and copy the contents of the floppy onto the hard drive. You can then use the hard drive for loading files.

1. **Insert the floppy disk in the front disk drive of the logic analyzer.**
2. **Go to the Flexible Disk menu.**
3. **Configure the menu to load.**
4. **Use the knob to select the appropriate configuration file.**

   Choosing the correct configuration file depends on which analyzer you are using. The configuration files are shown with the logic analyzer connection tables, and are also in the table on the next page.
5. **Select the appropriate analyzer on the menu. The HP 16500 logic analyzers are shown in the table on page 1-15.**

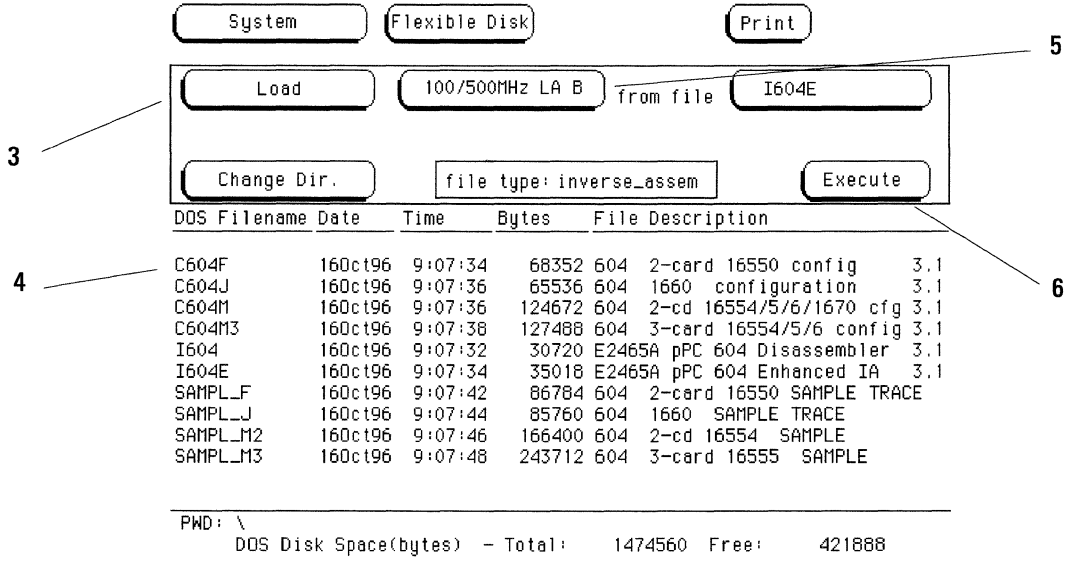
6 **Execute the load operation on the menu to load the file into the logic analyzer.**

The logic analyzer is configured for PowerPC 604 analysis by loading the appropriate configuration file. Loading this file also automatically loads an inverse assembler.

7 If you want to load a different inverse assembler, repeat steps 2 through 6, selecting the inverse assembler file in step 4. Refer to "Using the Inverse Assembler" in Chapter 2 for information on inverse assemblers.

8 **If you are using the HP 16505A Prototype Analyzer, insert the "16505 Prototype Analyzer" flexible disk into the disk drive of the prototype analyzer and update the HP 16505A from the Session Manager. You must close your workspace to run the update.**

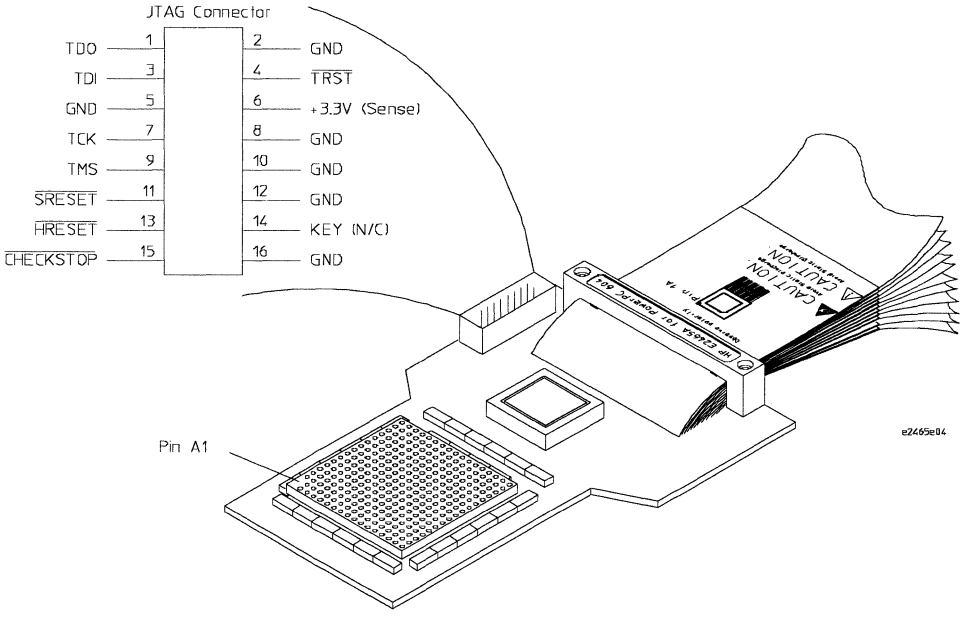
The HP 16505A Prototype Analyzer requires software version A.01.22 or higher to work with the HP E2465A.



Analyzer Model	16500 Analyzer Description	604 Configuration File
16550A (two card)	100 MHz STATE 500 MHz TIMING	C604F
16554A (two card)	0.5M SAMPLE 70/250 MHz LA	C604M
16555A/D (two card)	1.0M SAMPLE 110/500 MHz LA	C604M
16556A/D (two card)	1.0M SAMPLE 100/400 MHz LA	C604M
16554A (three card)	0.5M SAMPLE 70/250 MHz LA	C604M3
16555A/D (three card)	1.0M SAMPLE 110/500 MHz LA	C604M3
16556A/D (three card)	1.0M SAMPLE 100/400 MHz LA	C604M3
1660A/AS/C/CS		C604J
1670A/D		C604M

Connecting to the JTAG signals

The JTAG signals are routed to the JTAG connector. These signals can be probed using the GP [General Purpose] probes that are shipped with your logic analyzer. The figure below shows the location of the connector and signals.



JTAG Connector and Signals

2

Analyzing the PowerPC 604

Analyzing the PowerPC 604

This chapter describes modes of operation for the HP E2465A Preprocessor Interface. It also describes preprocessor interface data, symbol encodings, and information about the inverse assembler.

Modes of Operation

The PowerPC 604 preprocessor interface can be used in three different analysis modes: State-per-ack, State-per-clock, or Timing. The following sections describe these operating modes and how to configure the logic analyzer for each mode.

State-per-ack mode

In State-per-ack mode, the logic analyzer uses trigger sequencer store qualification to capture only address and data-acknowledge cycles. This is the default mode set up by the configuration files.

State-per-ack mode provides the greatest information density in the logic analyzer acquisition memory.

State-per-clock mode

In State-per-clock mode, every clock cycle is captured by the logic analyzer, including idle and wait states between and during tenures. To configure the logic analyzer for State-per-clock mode, use the Trigger menu to change the trigger store qualification to "anystate". For additional information, refer to the "Trigger Menu" section.

Timing mode

In Timing mode, the logic analyzer samples the microprocessor pins asynchronously, typically with 4 ns resolution. To configure the logic analyzer for timing analysis, select the Configuration menu of the logic analyzer, select the Type field for Analyzer 1, and select Timing.

Format Menu

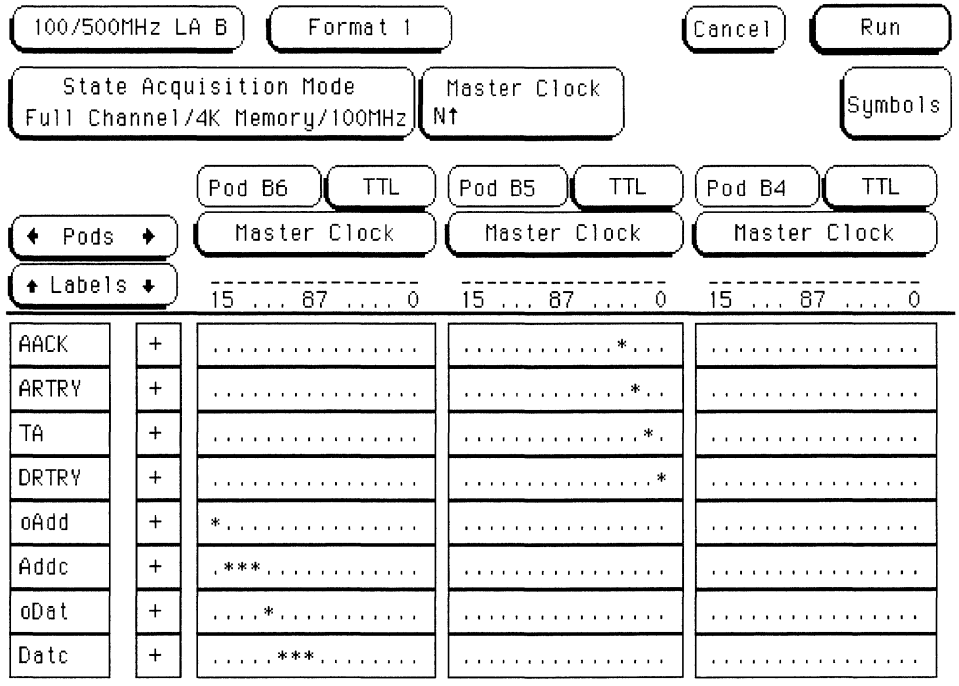
This section describes the organization of PowerPC 604 signals in the logic analyzer's Format Menu.

The HP logic analyzers and the PowerPC use opposite conventions to designate individual signals on a bus. In PowerPC nomenclature, bit 0 is the most significant; in the logic analyzers, bit 0 is the least significant. In PowerPC, A0 is the most significant bit of the address bus; on the analyzer, this bit is called ADDR31.

Most Significant	Least Significant	
A0	A31	PowerPC
ADDR31	ADDR0	Analyzer

This may cause confusion in the waveform menus when using Channel Mode Sequential or Individual.

The configuration software sets up the analyzer format menu to display either eight or eleven pods of data, depending on the analyzer. The following figure shows the Format Menu for the PowerPC 604 as configured on the HP 16550A.



The following table describes the signals that comprise the analyzer's STAT label. The inverse assembler uses STAT bits TC0, TSIZ0...2, TT0...3 , TBST, TA, ARTRY, DRTRY, ABB, DBB, and AACK. The signal-to-connector tables in Chapter 3 list all the PowerPC 604 signals probed and their corresponding analyzer channels.

Status Bit Description

Status Bit	Description
BR	The PowerPC 604 asserts Bus Request to indicate that it has business to conduct on the address bus.
BG	The memory system asserts Bus Grant to allow the 604 onto the address bus.
ABB	Address Bus Busy indicates that the address bus is in use.
TS	The PowerPC 604 asserts TS for one cycle to commence a transaction. It also serves as the data bus request signal if the TT signals indicate a data transfer.
XATS	XATS commences a "programmed i/o" (PIO) sequence in the extended address transfer protocol.
DBG	The memory system asserts Data Bus Grant to allow the 604 onto the data bus.
DBWO	The memory system may assert Data Bus Write Only to allow the 604 to envelope a data write (snoop push, typically) between the address and data phases of a data read.
DBB	Indicates Data Bus Busy.
AACK	The memory system asserts AACK for one cycle to acknowledge an address.
ARTRY	The memory system may assert ARTRY to cause the 604 to back off the bus and retry the transaction.
TA	The memory system asserts TA to acknowledge a data transaction.
DRTRY	The memory system may assert DRTRY to cancel the effect of a TA in the previous cycle.
TEA	The memory system may assert TEA to indicate a transfer error, e.g. an unmapped part of the address space.
TT 0:4	The Transfer Type signals indicate the direction and purpose of a bus transaction.
R/W (TT1)	TT1 is high for a read, low for a write.

Status Bit	Description
TC 0:2	The Transfer Code outputs provide further information about the current transfer. For a read, they indicate whether instructions or operands are being fetched.
TBST	When asserted, TBST indicates a four-beat burst transfer of eight words.
TSIZ 0:2	Indicates the size for the data transfer, in conjunction with TBST.
WT	Write Through indicates a write-through transaction that should be pushed through local caches to shared memory.
CI	Cache Inhibit indicates that the 604 will not cache a read.
GBL	Global indicates the transaction is global, i.e., should be snooped by all other caching devices on the bus.
SRESET	Input asserted causes the 604 to undergo a soft reset.
HRESET	Input asserted causes the 604 to undergo a hard reset.
CKSTP_IN	Input asserted causes the 604 to undergo machine check processing.
INT	Input indicates an external interrupt is pending.
CKSTP_OUT	Output indicates that the 604 has entered the machine check state, i.e., stopped.

The configuration files also define a number of symbols which make several of the STAT fields easier to interpret. The following table lists the symbol descriptions.

Symbol Description

Label	Symbol	Encoding
acks	idle	1111
	ARTRY	xxx0
	DRTRY	0xxx
	TA AACK	x00x
	AACK	xx0x
	TA	x0xx
TT	Clean Block	00000
	Flush Block	00100
	SYNC	01000
	Kill Block	01100
	EIEIO	10000
	ECOWX write	10100
	TLBIE	11000
	ECIWX rd	11100
	LWARX hit	00001
	reserved	00101
	TLBSYNC	01001
	?ICBI	01101
	?reserved	1xx01
	Wr/Flush	00010
	Wr/Kill	00110
	Read	01010
	Rd/Modify	01110
	STWCX.	10010
	?reserved	10110
	LWARX cach inhib	11010
	LWARX miss	11110
	?reserved	00x11
	?Rd/No cache	01011
	?reserved	01111
	?reserved	1xx11

Label	Symbol	Encoding
R/W	rd	1
	wr	0
TSIZ	burst	xxx0
	8 byte	0001
	1 byte	0011
	2 byte	0101
	3 byte	0111
	4 byte	1001
	5?byte	1011
	6?byte	1101
	7?byte	1111
STAT	inst fetch	1xxx xxxx xxxx xxxx xxxx x1xx x1xx 0xxx

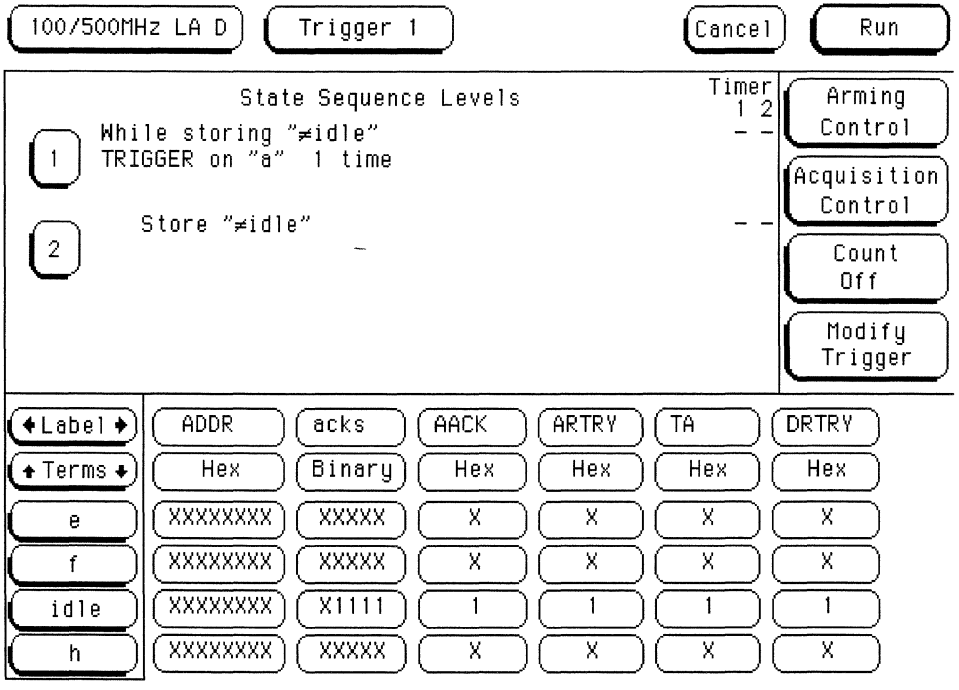
The least significant bit of the TSIZ label is the TBST signal.

An instruction fetch is indicated by AACK asserted (address & qualifiers valid), R/W (TT1) asserted for read, and TC0 and oAdd asserted.

Trigger Menu

This section describes some PowerPC 604-specific considerations in triggering the analyzer.

The trigger menu determines what will be acquired by the analyzer and when it will be acquired. The E2465A software preconfigures a storage qualification term to exclude wait and idle states from the analyzer's memory. The following figure shows the trigger menu as configured by the HP 16550A.



The configuration software renames pattern term "g" to "idle" and assigns it a pattern with AACK, ARTRY, TA, and DRTRY, all high (de-asserted). The sequencer is programmed to store only states ≠ idle. That is, only states where one or more of these signals is asserted will be stored.

To capture an address

To accurately trigger on a specific address, enter the address in the ADDR field of a trigger term and also enter 0 in the AACK field of the term. This will ensure against false triggering on a floating address bus.

The instruction addresses presented on the PowerPC 604 address bus always end in hex 0 or hex 8. When the instruction cache is enabled, the 604 will burst four data beats per address and will not update the address as it bursts. To accurately trigger on the fetch of a particular address when bursting, the least significant five bits of the address should be "don't cares. " Change the base of the ADDR label to Binary to enter the 5 X's.

To change the trigger conditions

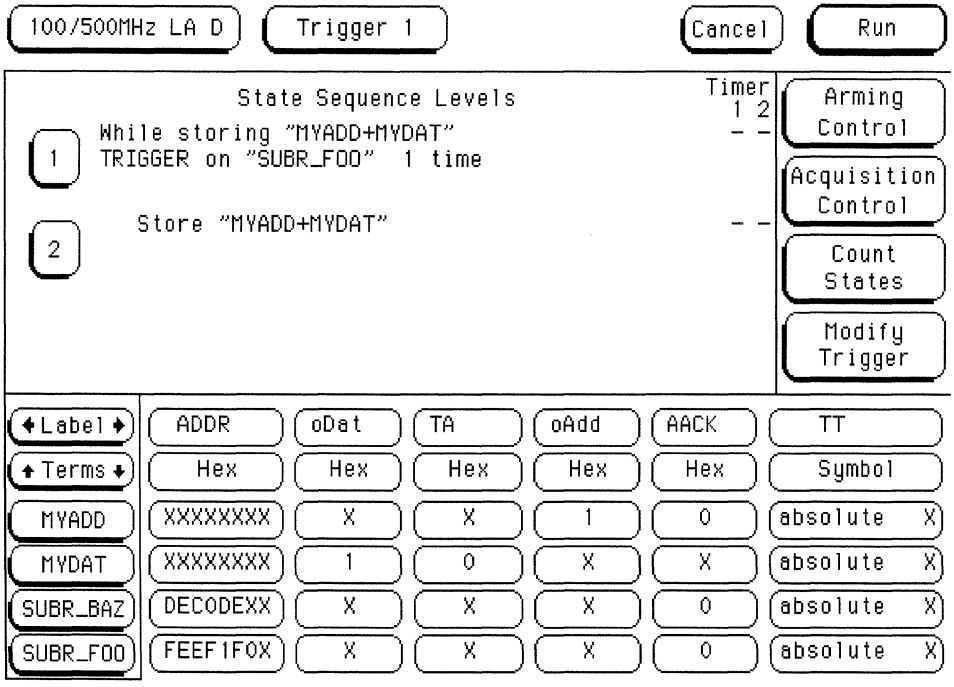
You can use the trigger menu to change the triggering and storage qualification to include or exclude specified cycles.

Excluding tenure from other bus masters

In a system in which devices other than the PowerPC 604 assume bus mastership, you can exclude tenures by other masters by defining new terms and changing store qualification. Define the terms MYDAT and MYADD, where:

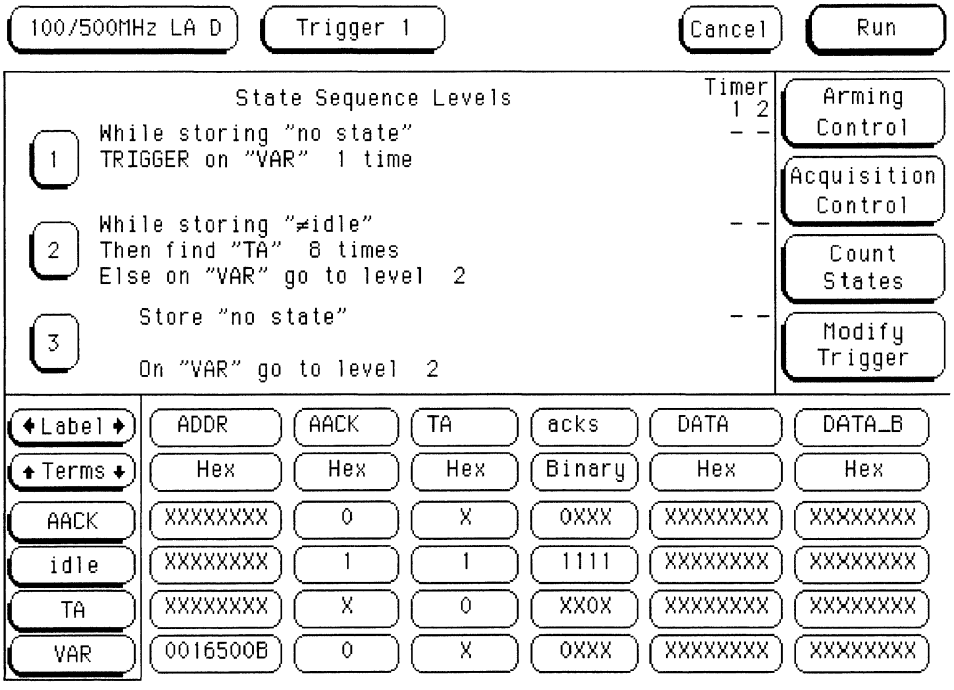
MYDAT = (oDat = 1)• (TA = 0)
MYADD = (oAdd = 1)• (AACK = 0)

Change the store qualification to the ORed combination MYDAT + MYADD, as shown in the following figure.



Capturing Isolated Addresses

The loose coupling of the address and data buses on the PowerPC 604 makes it more difficult to trace only activities associated with a given address, such as writes to a variable. Depending on how deep the pipeline is, an address of interest may be followed by up to eight data beats before the data associated with the address appears on the bus. One technique to trace writes to a variable is shown below. (The data cache is off or in write-through mode.)



Configuring for State-per-clock mode

To configure the analyzer to store wait and idle states, change the storage qualification from "≠ idle" to "anystate". Doing so will capture all states (state-per-clock).

Using the Inverse Assembler

This section discusses the general output format of the inverse assembler and processor-specific information.

To select a different inverse assembler

There are two inverse assemblers provided, a generic inverse assembler (I604) and an enhanced inverse assembler (I604E). The enhanced inverse assembler only works with certain configurations of logic analyzer, frame, and software revision. One of these inverse assemblers is loaded by default when you load the analyzer configuration file.

The generic disassembler provides basic disassembly, with overfetch marking and high/low alignment selection. The enhanced disassembler adds the ability to show or suppress various states from the state listing, and allows you to select mnemonic and numeric display formats.

The generic inverse assembler is loaded automatically on the HP 1660-series equipment with a revision level lower than 2.00. The generic inverse assembler may be manually loaded on any of the logic analyzers, using the procedure described in chapter 1.

The enhanced inverse assembler is loaded automatically on the HP 16500B/C and the HP 1670A/D, and on the HP 1660-series equipment with a software revision of 2.00 or higher.

To load a different inverse assembler, refer to "Loading the Configuration" in Chapter 1, except select the desired inverse assembler instead of the configuration file for the load.

To display captured state data

- **Select the Listing Menu for your logic analyzer.**

  The logic analyzer displays captured state data in the Listing Menu. The inverse assembler display is obtained by setting the base for the DATA label to Invasm. The following figure shows the Listing Menu as configured on the HP 16550A.

100/500MHz LA B | Listing 1 | Invasm | Cancel | Run

Markers Off

Acquisition Time
25 Apr 1996 14:58:31

Label> / Base>	ADDR / Hex	604 DATA Bus 10=hex 10.=decimal %10=binary k9	acks / Symbol	T / S
208	FFFC0007	5D8 mfspr r0 lr	TA	R
		5DC >bl FFF065E0		
209	FFF065E0	wait	AACK	R
210	FFFC0007	5E0 mfspr r3 lr	TA	R
		5E4 addi r4 r3 0000		
211	FFF065E8	wait	AACK	R
212	FFFC0007	5E8 addi r4 r4 0020	TA	R
		5EC lwz r4 0000(r4)		
213	FFF065F0	wait	AACK	R
214	FFFC0007	5F0 subi r3 r3 0008	TA	R
		5F4 add r3 r3 r4		
215	FFF065F8	wait	AACK	R
216	FFFC0007	5F8 mtspr lr r0	TA	R
		5FC blr		
217	FFF06600	wait	AACK	R
218	FFF06600	600 -fsub? f31 f31 f19	TA	R

The columns on the left of the inverse assembly data display are the least significant hexadecimal digits of an instruction or burst address. These may be useful for matching an execution trace to an assembly listing. Because the PowerPC 604 presents one address and then reads two or eight instructions for each address, the less-significant bits are synthesized by the disassembler. On the HP 16505A Prototype Analyzer, the entire synthesized address appears under the label "PC". The actual address bits presented by the PowerPC 604 may be observed under the ADDR label.

The fourth column may contain an underscore "_", which indicates a break in the sequential flow of instruction addresses.

The fifth column displays overfetch and branch-and-link indicators as described in the overfetch marking section on page 2-17. The remaining disassembly listing resembles an assembly listing.

Interpreting Data

General purpose registers are displayed as r0, r1, ..., r31. Floating point registers are displayed as f0, f1, ..., f31. Condition registers are displayed as cr0, cr1, ..., cr7. Special purpose registers are displayed using their mnemonic.

Most numerical data is displayed in hexadecimal, e.g., "lwz r28 0044(r1)."

Bit numbers and shift counts are displayed in decimal with a dot suffix, e.g., "cror 31. 31. 31."

A few instructions display their operands in binary with a "%" prefix, e.g., "mtfsfi 4 %0101."

The disassembler decodes the full PowerPC instruction set architecture, including 64-bit mode instructions and optional instructions not implemented on the 604. When these unimplemented opcodes are encountered, the instruction mnemonic has a "?" prefix. If a reserved bit is set in an instruction opcode field, a "?" is appended most often to the mnemonic, but in some cases to an operand.

An instruction word of 00000000 is decoded as "illegal." Otherwise, if an opcode is invalid, it is shown as "Undefined Opcode".

Branch Instructions

If the address of a branch relative instruction is known, its target is presented as an absolute hex address (or as a symbol if it matches an ADDR pattern or range symbol). If the address of a branch relative instruction is not known, its target is displayed as a hexadecimal offset such as +00000C30 or -00000048.

If a branch hint is encoded, a "+" (for predicted taken) or a "-" (for predicted not taken) is appended to the conditional branch mnemonic.

Overfetch Marking

Overfetch refers to instructions which are fetched but not executed by the processor. They may arise from the following sources:

- When bursting, the 604 first fetches the critical doubleword of an eight-word cache line. The memory system then provides succeeding doublewords. If the critical doubleword was not the first doubleword of the line, the memory system wraps at the line boundary to the first doubleword. Doublewords fetched after the line wrap are not in the sequential execution path and are marked with an asterisk "*".
- When the 604 executes a branch instruction, the instructions between the branch and the branch target are not executed. These instructions are indicated with a hyphen "-", or if the bus trace is ambiguous, with an interrogation point "?". If the instruction cache is enabled, the branch target may already be in the cache and will not be fetched over the bus. The remaining cache line containing the branch will be marked as overfetch.

An exception to the above includes branches with the link bit set, that record the next instruction address in the link register ("lr"). Frequently, these are subroutine branches which will return to the instructions following the branch. These branch-and-link instructions are indicated by a ">".

For conditional branches whose target addresses are not known, or are known but not seen in the bus traffic, the inverse assembler cannot always determine if the branch was taken and will not mark ensuing states as overfetch.

Little-Endian Mode

The inverse assembler is designed to support the native big-endian mode of operation on the PowerPC 604. When operating in little-endian mode, the 604 uses a technique known as "address munging" to convert internal little-endian addresses into external big-endian addresses. Internal and external addresses may differ from one another in the three least significant bits.

In little-endian operation, in a given data beat, the instruction word from DL0..31 (DATA_B label; external address xxx4) will be dispatched before the instruction word from DH0..31 (DATA label; external address xxx0). You can compensate for this by exchanging the DATA and DATA_B labels in the Format menu. However, while this will correctly order 32-bit word reads on the 64-bit data bus, it will cause byte- and half- word reads and writes to appear on the opposite side of the bus, and swap the halves of double-word reads and writes.

To use the Invasm key

The disassembler may occasionally mispredict a conditional branch instruction as taken and incorrectly mark subsequent states as overfetch. The following steps may be taken to correct this:

- **Roll the first incorrectly marked state to the top of the listing screen and select the Invasm key.**
- **Select High or Low as the first or second word of the double word that is incorrectly marked.**

  Note that the PowerPC 604 may branch to the second word of a doubleword without the disassembler detecting it. This could be a branch from cache, or via the lr, ctr, or srr0 registers. If the first word of the target doubleword is a branch, the inverse assembler may incorrectly mark the second word as overfetch.

To use the Invasm Options key

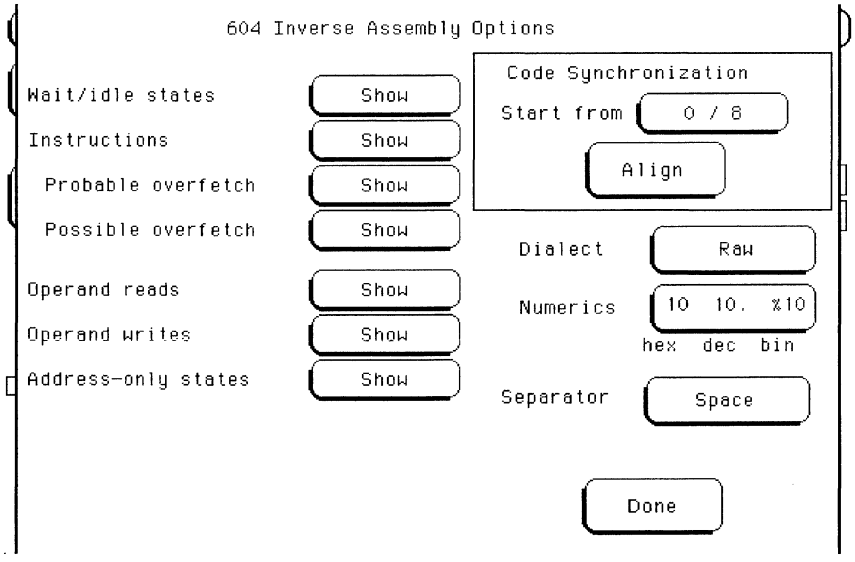
The enhanced inverse assembler contains additional features which use the increased capabilities of some of the logic analyzers. These features are accessed through the Invasm Options menu. Note that all the features in the generic inverse assembler are included in the enhanced inverse assembler (see previous section).

The I604E Inverse Assembly Options menu contains three functions: display filtering with Show/Suppress selections, Code Synchronization, and Display Options. The following sections describe these functions.

If the X or O pattern markers are turned on, and the designated pattern is found in a state that has been Suppressed with display filtering, the following message will appear on the logic analyzer display: "X (or O) pattern found, but state is suppressed."

Show/Suppress

The enhanced inverse assembler (I604E) supports selective suppression of certain states in the state listing display. The show/suppress settings do not affect the data that is stored by the logic analyzer; they only affect whether that data is displayed or not. The same data can be examined with different settings, for different analysis requirements. The figure below shows the states that have the Show/Suppress option.



Suppressing wait/idle states is useful for obtaining a state-per-cycle display of acquired data. Suppressing overfetched instructions may assist in following program execution trace more clearly. Suppressing instructions may be useful if your primary interest is data operand reads and writes.

"Probable overfetch" means states marked with a "*" or "-". "Possible overfetch" means states marked with a "?".

When instructions are suppressed, the overfetch show/suppress controls have no effect.

This function allows faster analysis in two ways. First, unneeded information can be filtered out of the display. Second, particular operations can be isolated by suppressing all other operations. For example, memory writes can be shown, with all other operations suppressed, allowing quick analysis of memory writes.

Code Synchronization

The Code Synchronization enables the inverse assembler to resynchronize with the microprocessor code. In some cases the prefetch marking algorithm in the inverse assembler may lose synchronization, and unused prefetches or executed instructions may be incorrectly marked. If any of the Code Reads are suppressed, this could cause some executed instructions to be missing from the display.

To resynchronize the inverse assembler, select "0/8" (for the high word) or "4/C" (for the low word) and press "Align".

Display Formats

The enhanced inverse assembler (I604E) allows you to configure various aspects of the disassembler display. The configuration options are listed below.

+ Separator Either blanks or commas may be used to separate operands.

```
andis. r3 r3 F000
andis. r3,r3,F000
```

The generic disassembler (I604) always uses blanks as separators.

+ Numerics Either of two conventions may be selected to display numeric data. The conventions are dense and common.

	dense	common
hex	FFF00230	0xFFF00230
decimal	31.	31
binary	%00100	0b00100

Since most of the numeric data produced by the inverse assembler is hexadecimal, the dense format uses a little less screen space. The generic disassembler (I604) always uses dense numerics.

+ Dialect Three mnemonic dialects are available: Raw, Simplified, and Extended. The generic disassembler (I604) always uses the Extended dialect.

The Raw dialect provides mnemonics and operands as specified by the PowerPC programming environment, eg the rotate left word immediate then and with mask instruction:

```
rlwinm r30 r30 16. 16. 31.
```

The Simplified dialect simplifies the operands for the rotate instructions: the rlwinm bit mask is presented as the hex value of the mask generated by the two decimal bit numbers:

```
rlwinm r30 r30 16. 0000FFFF
```

Additionally, a number of common extensions to PowerPC assembly language are decoded:

- Conditional traps and branches decode the condition mnemonically when possible. For some conditions which have no conventional mnemonics (for example, "signed less than or unsigned greater than"), the condition field is displayed in binary.
- The L bit is omitted as a compare operand. Instead, compares are decoded as "cmpw" (or "?cmpd").
- "Add immediate" instructions with a negative immediate operand are decoded as subtract immediate ("subi").
- "Subtract from" instructions subf and subfc are decoded as subtract instructions sub and subc with the operands exchanged so that "sub r3 r4 r5" is mnemonically interpreted as "r3 = r4 - r5."
- ori r0 r0 0000 is decoded as "nop".
- add immediate and add immediate shifted instructions, addi and addis, with a null source register are decoded as load immediate and load immediate shifted, li and lis.
- or instructions with identical source registers are decoded as move register, mr.
- nor instructions with identical source registers are decoded as not register, not.
- xor and eqv instructions with identical source and destination registers are decoded as clear and set, clr and set, respectively.
- The cror, crnor, crxor, and creqv instructions map analogously to crmv, crnot, crclr, and crset.
- When the mtcrf instruction field mask specifies the entire cr, it is decoded as mtcr.

The Extended dialect adds several extended opcodes for the rotate instructions. For example, the function of the rlwinm instruction

```
rlwinm r30 r30 16. 16. 31.
```

is to shift right word immediate, eg

```
srwi  r30 r30 16.
```

The following listing shows the extended mnemonics for the integer rotate instructions.

Mnemonic	Decoded As	
rlwimi (rotate left word immediate then mask insert)	inslwi insrwi	insert from left immediate insert from right immediate
rlwinm (rotate left word immediate then AND with mask)	rotlwi rotrwi slwi srwi extlwi extrwi clrlwi clrrwi clrlslwi	rotate left immediate rotate right immediate shift left immediate shift right immediate extract and left justify immediate extract and right justify immediate clear left immediate clear right immediate clear left and shift left immediate
rlwnm (rotate left word then AND with mask)	rotlw	rotate left

PowerPC assemblers support a number of extended mnemonics for some popular assembly language instructions as described in the PowerPC Architecture. The HP E2465A disassembler supports the following extensions of dialect-sensitive instructions.

raw	simplified	extended
bc %00100,2,FFF00230	bne cr0,FFF00230	bne cr0,FFF00230
tw %10000,r5,r6	tw lt,r5,r6	tw lt,r5,r6
cmp cr1,0,r0,r16	cmpw cr1,r0,r16	cmpw cr1,r0,r16
ori r0,r0,0000	nop	nop
addi r6,r6,FCFC	subi r6,r6,0304	subi r6,r6,0304
subf r7,r19,r16	sub r7,r16,r19	sub r7,r16,r19
addi r3,0,7000	li r3,7000	li r3,7000
addis r3,0,7000	lis r3,7000	lis r3,7000

mtcrf %11111111,r5	mtcr r5	mtcr r5
or r4,r5,r5	mr r4,r5	mr r4,r5
nor r4,r5,r5	not r4,r5	not r4,r5
xor r7,r7,r7	clr r7	clr r7
eqv r8,r8,r8	set r8	set r8
creqv 7,7,7	crset 7	crset 7
crxor 8,8,8	crclr 8	crclr 8
cror 7,8,8	crmv 7,8	crmv 7,8
crnor 8,9,9	crnot 8,9	crnot 8,9
rlwnm r8,r7,r6,0,31.	rlwnm r8,r7,r6,FFFFFFFF	rotlw r8,r7,r6
rlwimi r3,r3,24.,8,23.	rlwimi r3,r3,24.,00FFFF00	inslwi r3,r3,16.,8
rlwimi r8,r3,17,8,23.	rlwimi r3,r3,17.,00FE0000	insrwi r8,r3,7,8
rlwinm r6,r4,8,0,14	rlwinm r6,r4,8,0xFFFE0000	extlwi r6,r4,15,8
rlwinm r6,r4,16,24,31	rlwinm r6,r4,16,0x000000FF	extrwi r6,r4,8,8
rlwinm. r6,r4,4,0,31	rlwinm. r6,r4,4,0xFFFFFFFF	rotlwi. r6,r4,4
rlwinm r6,r4,28,0,31	rlwinm r6,r4,4,0xFFFFFFFF	rotrwi r6,r4,4
rlwinm r6,r4,1,0,30	rlwinm r6,r4,1,0xFFFFFFFE	slwi r6,r4,1
rlwinm r6,r4,31,1,31	rlwinm r6,r4,31,0x7FFFFFFF	srwi r6,r4,1
rlwinm r6,r4,0,1,31	rlwinm r6,r4,0,0x7FFFFFFF	clrlwi r6,r4,1
rlwinm r6,r4,0,0,7	rlwinm r6,r4,0,0xFF000000	clrrwi r6,r4,14
rlwinm r6,r4,6,6,25	rlwinm r6,r4,6,0x03FFFFC0	clrlslwi r6,r4,12,6

Done Field

When you are finished with the Invasm Options pop-up menu, use the Done key to return to the Listing menu.

Disabling the Instruction Cache on the PowerPC

When the instruction cache is enabled, many PowerPC instructions are executed from the cache and do not appear on the external bus. To get an execution trace on the bus, the instruction cache can be disabled. This must be done in supervisor mode.

- Disable the cache with the following code:

```
mfspr   r3 hid0
rlwinm  r3 r3 0 17 15  # clear bit 16 (ICE)
mtspr   hid0 r3
isync
```

- To also disable the data cache use:

```
mfspr   r3 hid0
rlwinm  r3 r3 0 18 15  # clear ICE and DCE
mtspr   hid0 r3
isync
```

- To invalidate and disable the caches use:

```
mfspr   r3 hid0
ori     r3 0C00         # set ICFI and DCFI
mtspr   hid0 r3
rlwinm  r3 r3 0 22 19  # clear ICFI and DCFI
mtspr   hid0 r3
rlwinm  r3 r3 0 18 15  # clear ICE and DCE
mtspr   hid0 r3
isync
```

3

Preprocessor Interface Hardware Reference

Preprocessor Interface Hardware Reference

This chapter contains reference information on the HP2465A hardware including product, electrical, and environmental characteristics, signal mapping, circuit board dimensions, and repair information.

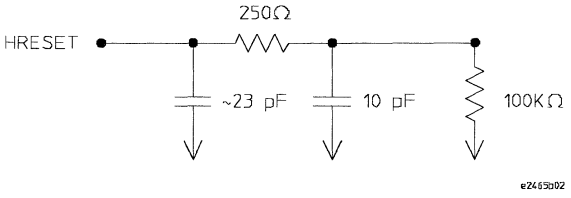
Operating characteristics

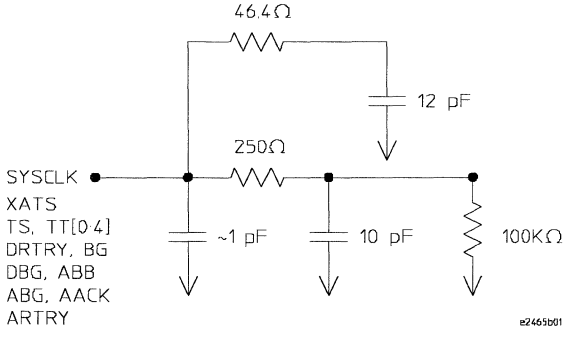
The following operating characteristics are not specifications, but are typical operating characteristics for the preprocessor interface.

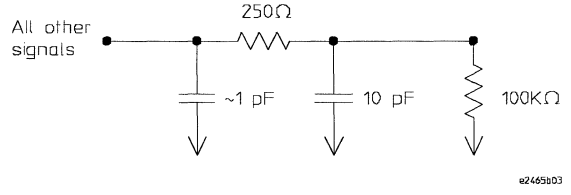
Product Characteristics		
Microprocessors Supported	PowerPC 604	
Package Supported	287-pin PGA	
Logic Analyzers Supported	HP 1660A/AS,C/CS, HP 1670A/D, HP 16550A (two cards), HP 16554A/55A/56A (two or three cards), HP 16555D/56D (two or three cards)	
Probes Required	8 required for disassembly. 11 available.	
Accessories required	None	
Electrical Characteristics		
Power Requirements	120 mA @ 5V, supplied by the logic analyzer.	
Signal Line Loading	See diagrams on next page.	
Setup/Hold Requirement	For address and data alignment, the PLD requires that the ACK line have a 10.9 ns setup/0 s hold.	
Environmental Characteristics		
Temperature	Operating	0 to + 50 degrees C +32 to +131 degrees F
Altitude	Operating	4,600 m 15,000 feet
Humidity	Up to 75% noncondensing. Avoid sudden, extreme temperature changes which could cause condensation on the circuit board.	

Signal line loading

The following diagrams show the loading for all signals.







Theory of operation

The HP E2465A Preprocessor Interface contains a programmable logic device (PLD) which tracks bus ownership, and tags address and data tenures owned by the PowerPC 604 with three-bit counts. These counts are used by the disassembler to correlate address and data tenures, and appear on the logic analyzer as labels "Addc" and "Datc". The bus mastership signals are true high (asserted with a logic level 1), and are labeled oAdd and oDat. A block diagram of the PLD is shown on the next page.

Target Mastership Of Data/Address Bus

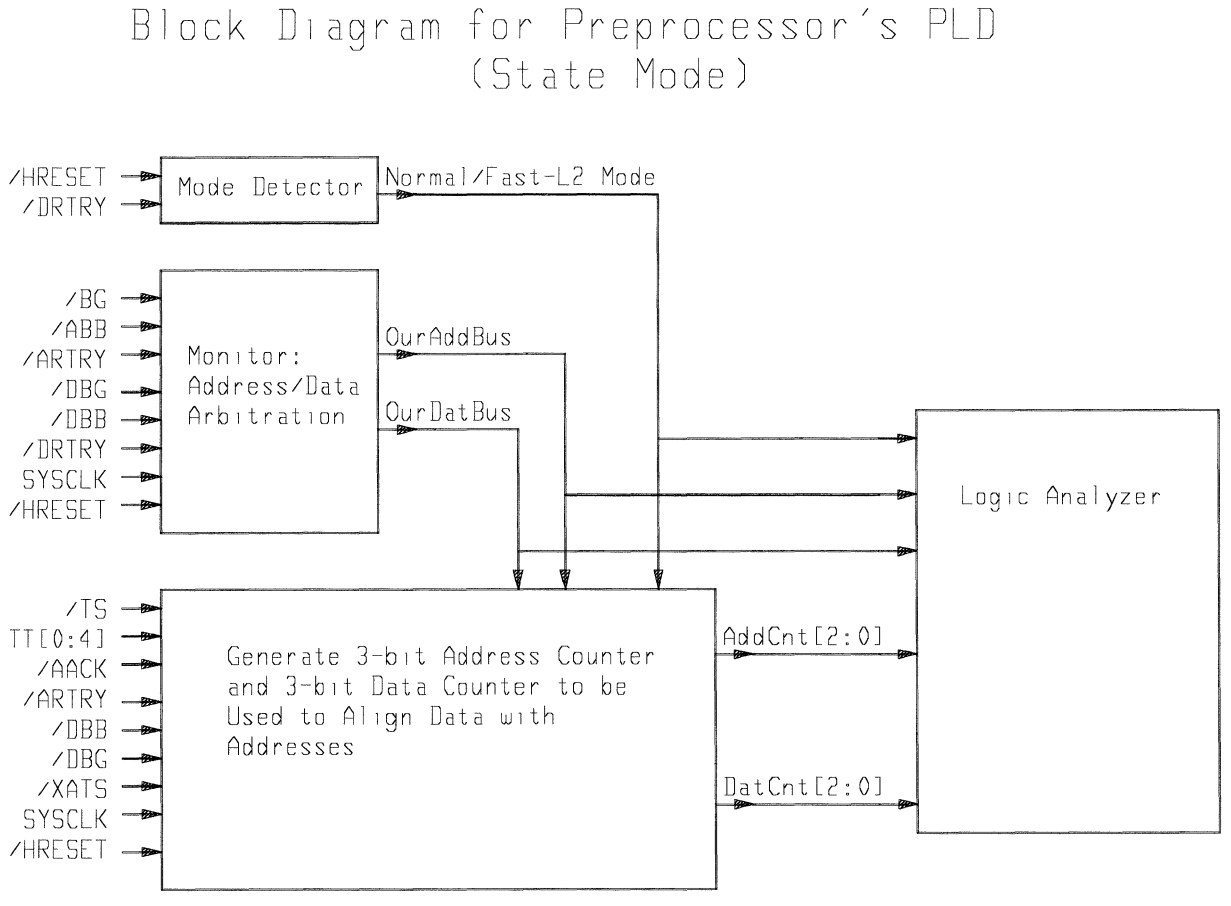
Mastership of the data bus is indicated after the target receives a qualified data bus grant. Mastership of the address bus is indicated by sampling the ABB signal one cycle after BG is asserted. If ABB is asserted, then the target processor has the bus. This will be indicated to the logic analyzer one cycle later (two cycles after BG is asserted). If ABB is NOT asserted the cycle following BG assertion, then the target chose not to take over the address bus and mastership of the address bus will not be indicated to the logic analyzer.

Normal Mode Vs. Fast-l2 Mode Of Operation

On power-up, the PLD detects whether the 604 is running in the Normal mode or Fast-L2 mode of operation by sampling DRTRY on the rising edge of HRESET. If DRTRY is asserted, then the processor will run in Fast-L2 mode instead of Normal mode. There are no data retries in Fast-L2 mode. In Normal mode, the clock to the data counter is a rising DBB. In Fast-L2 mode, the data counter clock is also a rising DBB. As a result of using DBB as the data counter clock when in Fast-L2 mode, there must always be a DBB pulse between burst operations in Fast-L2 mode. By using DBB, the PLD does not need to look at DRTRY or TBST. Using DBB also eliminates the need for pipelining the TBST bit.

Direct Store Mode

Direct store operations occur in either Normal mode or Fast-L2 mode. Direct store operations occur when XATS is strobed low. The address and data counters are not affected when direct store transactions are in progress.



HP E2465A Block Diagram

Signal-to-Connector Mapping

The following table shows the the preprocessor PGA socket pin mapping.

PowerPC 604 Signal List

Pod	2x20 pin	Analyzer Bit	PGA Pin #	Signal Name	Analyzer Labels	Analyzer Labels
P6	3	CLK1	D7	NAPRUN		NAPRUN
P6	7	15	P13	DH0	DATA	
P6	9	14	N12	DH1	DATA	
P6	11	13	T15	DH2	DATA	
P6	13	12	U15	DH3	DATA	
P6	15	11	R13	DH4	DATA	
P6	17	10	U14	DH5	DATA	
P6	19	9	N10	DH6	DATA	
P6	21	8	P11	DH7	DATA	
P6	23	7	T11	DH8	DATA	
P6	25	6	U11	DH9	DATA	
P6	27	5	R10	DH10	DATA	
P6	29	4	U10	DH11	DATA	
P6	31	3	U9	DH12	DATA	
P6	33	2	T9	DH13	DATA	
P6	35	1	N9	DH14	DATA	
P6	37	0	P9	DH15	DATA	
P5	3	CLK1	D15	MCP		MCP
P5	7	15	R9	DH16	DATA	
P5	9	14	U8	DH17	DATA	
P5	11	13	R8	DH18	DATA	
P5	13	12	U7	DH19	DATA	
P5	15	11	N8	DH20	DATA	
P5	17	10	P7	DH21	DATA	
P5	19	9	T7	DH22	DATA	
P5	21	8	U6	DH23	DATA	
P5	23	7	R7	DH24	DATA	
P5	25	6	R6	DH25	DATA	
P5	27	5	N7	DH26	DATA	
P5	29	4	U5	DH27	DATA	
P5	31	3	T5	DH28	DATA	
P5	33	2	U4	DH29	DATA	
P5	35	1	R5	DH30	DATA	
P5	37	0	U3	DH31	DATA	

Pod	2x20 pin	Analyzer Bit	PGA Pin #	Signal Name	Analyzer Labels	Analyzer Labels
P4	3	CLK1	E14	INT	INT	
P4	7	15	L16	DL0	DATA_B	
P4	9	14	K15	DL1	DATA_B	
P4	11	13	M17	DL2	DATA_B	
P4	13	12	L14	DL3	DATA_B	
P4	15	11	N17	DL4	DATA_B	
P4	17	10	M15	DL5	DATA_B	
P4	19	9	N16	DL6	DATA_B	
P4	21	8	L13	DL7	DATA_B	
P4	23	7	M13	DL8	DATA_B	
P4	25	6	N15	DL9	DATA_B	
P4	27	5	P17	DL10	DATA_B	
P4	29	4	R17	DL11	DATA_B	
P4	31	3	N14	DL12	DATA_B	
P4	33	2	P15	DL13	DATA_B	
P4	35	1	R16	DL14	DATA_B	
P4	37	0	U16	DL15	DATA_B	
P3	3	CLK1	C9	CKSTP_IN		CHKIN
P3	7	15	R14	DL16	DATA_B	
P3	9	14	N11	DL17	DATA_B	
P3	11	13	T13	DL18	DATA_B	
P3	13	12	R12	DL19	DATA_B	
P3	15	11	U13	DL20	DATA_B	
P3	17	10	R11	DL21	DATA_B	
P3	19	9	U12	DL22	DATA_B	
P3	21	8	N3	DL23	DATA_B	
P3	23	7	P3	DL24	DATA_B	
P3	25	6	N4	DL25	DATA_B	
P3	27	5	R2	DL26	DATA_B	
P3	29	4	T1	DL27	DATA_B	
P3	31	3	T3	DL28	DATA_B	
P3	33	2	R4	DL29	DATA_B	
P3	35	1	P5	DL30	DATA_B	
P3	37	0	N6	DL31	DATA_B	

Pod	2x20 pin	Analyzer Bit	PGA Pin #	Signal Name	Analyzer Labels	Analyzer Labels
P2	3	CLK1	E9	CKSTP_OUT		CHKOUT
P2	7	15	F13	A0	ADDR	
P2	9	14	E1	A1	ADDR	
P2	11	13	D17	A2	ADDR	
P2	13	12	F3	A3	ADDR	
					ADDR	
P2	15	11	E16	A4	ADDR	
P2	17	10	F1	A5	ADDR	
P2	19	9	E17	A6	ADDR	
P2	21	8	G5	A7		
P2	23	7	F15	A8	ADDR	
P2	25	6	G4	A9	ADDR	
P2	27	5	G13	A10	ADDR	
P2	29	4	G3	A11	ADDR	
P2	31	3	F17	A12	ADDR	
P2	33	2	G2	A13	ADDR	
P2	35	1	G14	A14	ADDR	
P2	37	0	G1	A15	ADDR	
P1	3	CLK1	L15	TS	TS	
P1	7	15	G15	A16	ADDR	
P1	9	14	H1	A17	ADDR	
P1	11	13	G16	A18	ADDR	
P1	13	12	H3	A19	ADDR	
P1	15	11	G17	A20	ADDR	
P1	17	10	J1	A21	ADDR	
P1	19	9	H13	A22	ADDR	
P1	21	8	H5	A23	ADDR	
P1	23	7	H15	A24	ADDR	
P1	25	6	J2	A25	ADDR	
P1	27	5	H17	A26	ADDR	
P1	29	4	J3	A27	ADDR	
P1	31	3	J13	A28	ADDR	
P1	33	2	L1	A29	ADDR	
P1	35	1	K13	A30	ADDR	
P1	37	0	M1	A31	ADDR	

Pod	2x20 pin	Analyzer Bit	PGA Pin #	Signal Name	Analyzer Labels	Analyzer Labels	
P7	3	CLK1	K17	XATS		XATS	
P7	7	15	*	OurAddBus	STAT	oAdd	
P7	9	14	*	AddCnt2	STAT	Addc	
P7	11	13	*	AddCnt1	STAT	Addc	
P7	13	12	*	AddCnt0	STAT	Addc	
P7	15	11	*	OurDatBus	STAT	oDat	
P7	17	10	*	DatCnt2	STAT	Datc	
P7	19	9	*	DatCnt1	STAT	Datc	
P7	21	8	*	DatCnt0	STAT	Datc	
P7	23	7	B9	HRESET	STAT	HRSET	
P7	25	6	D13	SRESET	STAT	SRSET	
P7	27	5	E8	BR	STAT	BR	busarb
P7	29	4	J5	BG	STAT	BG	busarb
P7	31	3	L3	ABB	STAT	ABB	busarb
P7	33	2	K1	DBG	STAT	DBG	busarb
P7	35	1	L17	DBB	STAT	DBB	busarb
P7	37	0	J14	TEA	STAT	TEA	

* These signals are generated by the preprocessor interface.

P8	3	CLK1	C10	SYSCLK		SYSCLK	
P8	7	15	E11	TSIZ0	STAT	TSIZ	
P8	9	14	A15	TSIZ1	STAT	TSIZ	
P8	11	13	B15	TSIZ2	STAT	TSIZ	
P8	13	12	E12	TBST	STAT	TSIZ	TBST
P8	15	11	A16	TT0	STAT	TT	TT0
P8	17	10	C14	TT1	STAT	TT	R/W
P8	19	9	C16	TT2	STAT	TT	TT2
P8	21	8	C17	TT3	STAT	TT	TT3
P8	23	7	E15	TT4	STAT	TT	TT4
P8	25	6	C6	TC0	STAT	TC	TC0
P8	27	5	A5	TC1	STAT	TC	TC1
P8	29	4	C7	TC2	STAT	TC	TC2
P8	31	3	K3	AACK	STAT	ACKs	AACK
P8	33	2	K5	ARTRY	STAT	ACKs	ARTRY
P8	35	1	J16	TA	STAT	ACKs	TA
P8	37	0	J17	DRTRY	STAT	ACKs	DRTRY

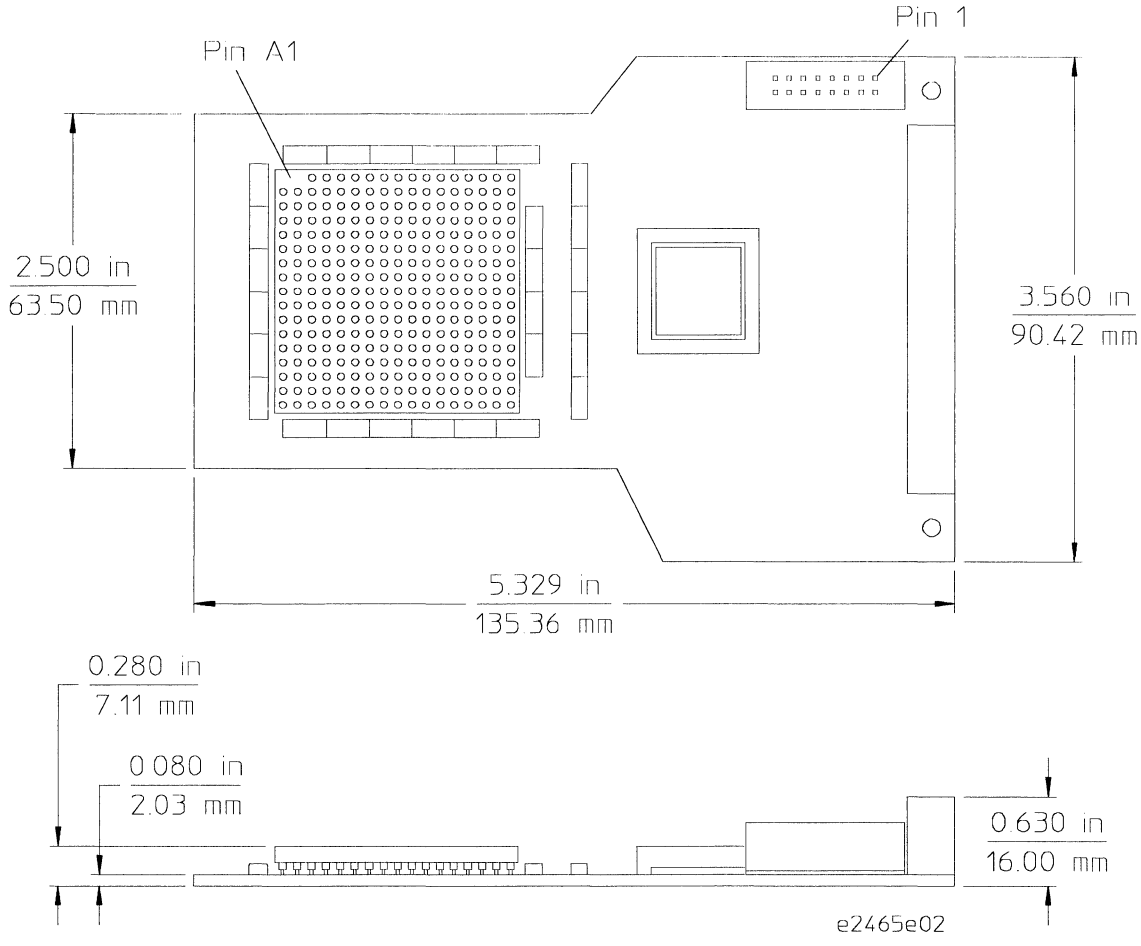
Pod	2x20 pin	Analyzer Bit	PGA Pin #	Signal Name	Analyzer Labels	Analyzer Labels
P9	3	CLK1	--			
P9	7	15	L4	DP0	DP	
P9	9	14	N1	DP1	DP	
P9	11	13	M3	DP2	DP	
P9	13	12	N2	DP3	DP	
P9	15	11	P1	DP4	DP	
P9	17	10	L5	DP5	DP	
P9	19	9	R1	DP6	DP	
P9	21	8	M5	DP7	DP	
P9	23	7	E6	AP0	AP	
P9	25	6	C4	AP1	AP	
P9	27	5	C5	AP2	AP	
P9	29	4	A4	AP3	AP	
P9	31	3	B7	DPE	DPE	
P9	33	2	C8	APE	APE	
P9	35	1	E7	CSE0	CSE	CSE0
P9	37	0	B5	CSE1	CSE	CSE1
P10		CLK1	C12	TCK	TCK	
P10		15	A12	TRST	TRST	
P10		14	C13	TMS	TMS	
P10		13	B13	TDI	TDI	
P10		12	A14	TDO	TDO	
P10		11	J15	DBDIS	DBDIS	
P10		10	J4	DBW0	DBW0	
P10		9	L2	SHD	SHD	
P10		8	F5	GBL	GBL	
P10		7	D1	WT	WT	
P10		6	E2	CI	CI	
P10		5	D5	RSRV	RSRV	
P10		4	E4	TBEN	TBEN	
P10		3	D9	HALTED	HALT	
P10		2	B17	SMI	SMI	
P10		1	A7	L2_INT	L2_INT	
P10		0	*	FastL2	FastL2	

* This signal is generated by the preprocessor interface.

Pod	2x20 pin	Analyzer Bit	PGA Pin #	Signal Name	Analyzer Labels	Analyzer Labels
P11	3	CLK1				
P11	7	15				
P11	9	14				
P11	11	13				
P11	13	12				
P11	15	11				
P11	17	10				
P11	19	9				
P11	21	8				
P11	23	7				
P11	25	6				
P11	27	5				
P11	29	4				
P11	31	3	A9	PLLCFG0	PLLCFG	PLLCFG0
P11	33	2	A10	PLLCFG1	PLLCFG	PLLCFG1
P11	35	1	A13	PLLCFG2	PLLCFG	PLLCFG2
P11	37	0	C11	PLLCFG3	PLLCFG	PLLCFG3

Circuit board dimensions

The following figure gives the dimensions for the preprocessor interface assembly. The dimensions are listed in inches and millimeters.



Circuit Board Dimension Diagram

Repair Strategy

The repair strategy for this preprocessor interface is board replacement. However, the following table lists some mechanical parts that may be replaced if they are damaged or lost. Contact your nearest Hewlett-Packard Sales Office for further information on servicing the board.

Exchange assemblies are available when a repairable assembly is returned to Hewlett-Packard. These assemblies have been set up on the "Exchange Assembly" program. This allows you to exchange a faulty assembly with one that has been repaired, calibrated, and performance verified by the factory. The cost is significantly less than that of a new assembly.

Replaceable Parts

HP Part Number	Description
E2465-66501	PowerPC 604 Interface Card
1200-1975	Pin Protector
E2465-68702	Configuration and Inverse Assembler Software

A

If You Have a Problem

If You Have a Problem

Occasionally, a measurement may not give the expected results. If you encounter difficulties while making measurements, use this chapter to guide you through some possible solutions. Each heading lists a problem you may encounter, along with some possible solutions.

If you still have difficulty using the analyzer after trying the suggestions in this chapter, please contact your local Hewlett-Packard service center.

CAUTION

When you are working with the analyzer, be sure to power down both the analyzer and the target system before disconnecting or connecting cables, probes, and preprocessors. Otherwise, you may damage circuitry in the analyzer, preprocessor, or target system.

Analyzer Problems

This section lists general problems that you might encounter while using the analyzer.

Intermittent data errors

This problem is usually caused by poor connections, incorrect signal levels, or marginal timing.

- ☐ **Remove and reseat all cables and probes, ensuring that there are no bent pins on the preprocessor interface or poor probe connections.**
- ☐ **Adjust the threshold level of the data pod to match the logic levels in the system under test.**
- ☐ **Use an oscilloscope to check the signal integrity of the data lines.**

  Clock signals for the state analyzer must meet particular pulse shape and timing requirements. Data inputs for the analyzer must meet pulse shape and setup and hold time requirements.

See Also See "Capacitive Loading" in this chapter for information on other sources of intermittent data errors.

Unwanted triggers

Unwanted triggers can be caused by instructions that were fetched but not executed.

- ☐ **Add the prefetch queue or pipeline depth to the trigger address to avoid this problem.**

  The logic analyzer captures prefetches, even if they are not executed. When you are specifying a trigger condition or a storage qualification that follows an instruction that may cause branching, an unused prefetch may generate an unwanted trigger.

No activity on activity indicators

- ☐ Check for loose cables, board connections, and preprocessor interface connections.
- ☐ Check for bent or damaged pins on the preprocessor probe.

No trace list display

If there is no trace list display, it may be that your trigger specification is not correct for the data you want to capture, or that the trace memory is only partially filled.

- ☐ Check your trigger sequencer specification to ensure that it will capture the events of interest.
- ☐ Try stopping the analyzer; if the trace list is partially filled, this should display the contents of trace memory.

Preprocessor Problems

This section lists problems that you might encounter when using a preprocessor. If the solutions suggested here do not correct the problem, you may have a damaged preprocessor. Contact your local Hewlett-Packard Sales Office if you need further assistance.

Target system will not boot up

If the target system will not boot up after connecting the preprocessor interface, the microprocessor (if socketed) or the preprocessor interface may not be installed properly, or they may not be making electrical contact.

- ☐ **Ensure that you are following the correct power-on sequence for the preprocessor and target system.**

  1 Power up the analyzer and preprocessor.

  2 Power up the target system.

  If you power up the target system before you power up the preprocessor, interface circuitry in the preprocessor may latch up and prevent proper target system operation.

- ☐ **Verify that the microprocessor and the preprocessor interface are properly rotated and aligned, so that the index pin on the microprocessor (such as pin 1 or A1) matches the index pin on the preprocessor interface.**

- ☐ **Verify that the microprocessor and the preprocessor interface are securely inserted into their respective sockets.**

- ☐ **Verify that the logic analyzer cables are in the proper sockets of the preprocessor interface and are firmly inserted.**

- ☐ **Reduce the number of extender sockets.**

See Also "Capacitive Loading" in this appendix.

Erratic trace measurements

There are several general problems that can cause erratic variations in trace lists and inverse assembly failures.

- ☐ **Ensure that the preprocessor configuration switches are correctly set for the measurement you are trying to make.**

  Some preprocessors include configuration switches for various features (for example, to allow dequeueing of the trace list). See Chapter 1 for information about setting configuration switches.

- ☐ **Do a full reset of the target system before beginning the measurement.**

  Some preprocessor designs require a full reset to ensure correct configuration.

- ☐ **Ensure that your target system meets the timing requirements of the processor with the preprocessor probe installed.**

  See "Capacitive Loading" in this chapter. While preprocessor loading is slight, pin protectors, extenders, and adapters may increase it to unacceptable levels. If the target system design has close timing margins, such loading may cause incorrect processor functioning and give erratic trace results.

- ☐ **Ensure that you have sufficient cooling for the microprocessor.**

  Microprocessors such as the i486, Pentium™, and MC68040 generate substantial heat. This is exacerbated by the active circuitry on the preprocessor board. You should ensure that you have ambient temperature conditions and airflow that meet or exceed the requirements of the microprocessor manufacturer.

Capacitive loading

Excessive capacitive loading can degrade signals, resulting in incorrect capture by the preprocessor interface, or system lockup in the microprocessor. All preprocessor interfaces add additional capacitive loading, as can custom probe fixtures you design for your application.

Careful layout of your target system can minimize loading problems and result in better margins for your design. This is especially important for systems that are running at frequencies greater than 50 MHz.

- ☐ Remove as many pin protectors, extenders, and adapters as possible.
- ☐ If multiple preprocessor interface solutions are available, use one with lower capacitive loading.

Inverse Assembler Problems

This section lists problems that you might encounter while using the inverse assembler.

When you obtain incorrect inverse assembly results, it may be unclear whether the problem is in the preprocessor or in your target system. If you follow the suggestions in this section to ensure that you are using the preprocessor and inverse assembler correctly, you can proceed with confidence in debugging your target system.

No inverse assembly or incorrect inverse assembly

This problem may be due to incorrect synchronization, modified configuration, incorrect connections, or a hardware problem in the target system. A locked status line can cause incorrect or incomplete inverse assembly.

- ☐ **Ensure that each logic analyzer pod is connected to the correct preprocessor connector.**

  There is not always a one-to-one correspondence between analyzer pod numbers and preprocessor cable numbers. Preprocessors must supply address (ADDR), data (DATA), and status (STAT) information to the analyzer in a predefined order. The cable connections for each preprocessor are often altered to support that need. Thus, one preprocessor might require that you connect cable 2 to analyzer pod 2, while another will require you to connect cable 5 to analyzer pod 2. See Chapter 1 for connection information.

- ☐ **Check the activity indicators for status lines locked in a high or low state.**

- ☐ **Verify that the STAT, DATA, DATA_B, and ADDR format labels have not been modified from their default values.**

  These labels must remain as they are configured by the configuration file. Do not change the names of these labels or the bit assignments within the labels. Some preprocessors also require other data labels. See Chapter 2 for more information.

- ☐ **Verify that all microprocessor caches and memory managers have been disabled.**

  In most cases, if the microprocessor caches and memory managers remain enabled you should still get inverse assembly. It may be incorrect because a portion of the execution trace was not visible to the logic analyzer.

- ☐ **Verify that storage qualification has not excluded storage of all the needed opcodes and operands.**

Inverse assembler will not load or run

You need to ensure that you have the correct system software loaded on your analyzer.

- ☐ **Ensure that the inverse assembler is on the same disk as the configuration files you are loading.**

  Configuration files for the state analyzer contain a pointer to the name of the corresponding inverse assembler. If you delete the inverse assembler or rename it, the configuration process will fail to load the disassembler.

  See Chapter 1 for details.

Intermodule Measurement Problems

Some problems occur only when you are trying to make a measurement involving multiple modules.

An event wasn't captured by one of the modules

If you are trying to capture an event that occurs very shortly after the event that arms one of the measurement modules, it may be missed due to internal analyzer delays. For example, suppose you set the oscilloscope to trigger upon receiving a trigger signal from the logic analyzer because you are trying to capture a pulse that occurs right after the analyzer's trigger state. If the pulse occurs too soon after the analyzer's trigger state, the oscilloscope will miss the pulse.

- ☐ **Adjust the skew in the Intermodule menu.**

  You may be able to specify a skew value that enables the event to be captured.

- ☐ **Change the trigger specification for modules upstream of the one with the problem.**

  If you are using a logic analyzer to trigger the scope, try specifying a trigger state one state before the one you are using. This may be more difficult than working with the skew because the prior state may occur more often and not always be related to the event you are trying to capture with the oscilloscope.

Messages

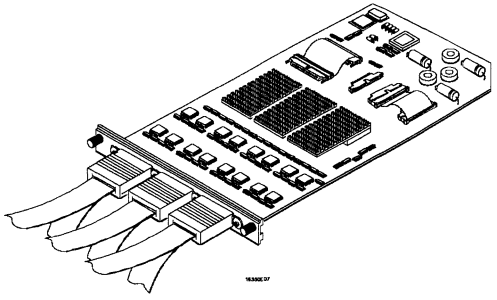
This section lists some of the messages that the analyzer displays when it encounters a problem.

". . . Inverse Assembler Not Found"

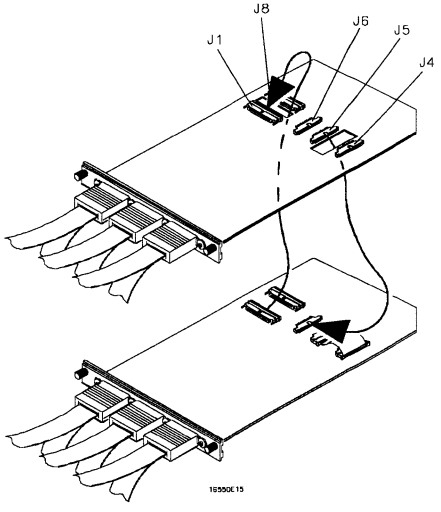
This error occurs if you rename or delete the inverse assembler file that is attached to the configuration file. Ensure that the inverse assembler file is not renamed or deleted, and that it is located in the same directory as the configuration file.

"Measurement Initialization Error"

This error occurs when you have installed the cables incorrectly for one or two HP 16550A logic analysis cards. The following diagrams show the correct cable connections for one-card and two-card installations. Ensure that your cable connections match the silk screening on the card, and that they are fully seated in the connectors. Then, repeat the measurement.



Cable Connections for One-Card HP 16550A Installations



Cable Connections for Two-Card HP 16550A Installations

See Also

The *HP 16550A 100-MHz State/500-MHz Timing Logic Analyzer Service Guide*.

"No Configuration File Loaded"

This is usually caused by trying to load a configuration file for one type of module/system into a different type of module/system.

- ☐ Verify that the appropriate module has been selected from the Load {module} from File {filename} in the HP 16500A/B disk operation menu. Selecting Load {All} will cause incorrect operation when loading most preprocessor interface configuration files.

See Also Chapter 1 describes how to load configuration files.

"Selected File is Incompatible"

This occurs when you try to load a configuration file for the wrong module. Ensure that you are loading the appropriate configuration file for your logic analyzer.

"Slow or Missing Clock"

- ☐ This error message might occur if the logic analyzer cards are not firmly seated in the HP 16500A/B or HP 16501A frame. Ensure that the cards are firmly seated.
- ☐ This error might occur if the target system is not running properly. Ensure that the target system is on and operating properly.
- ☐ If the error message persists, check that the logic analyzer pods are connected to the proper connectors on the preprocessor interface. See Chapter 1 to determine the proper connections.

"Time from Arm Greater Than 41.93 ms"

The state/timing analyzers have a counter to keep track of the time from when an analyzer is armed to when it triggers. The width and clock rate of this counter allow it to count for up to 41.93 ms before it overflows. Once the counter has overflowed, the system does not have the data it needs to calculate the time between module triggers. The system must know this time to be able to display data from multiple modules on a single screen.

"Waiting for Trigger"

If a trigger pattern is specified, this message indicates that the specified trigger pattern has not occurred. Verify that the triggering pattern is correctly set.

- ☐ **When analyzing microprocessors that fetch only from word-aligned addresses, if the trigger condition is set to look for an opcode fetch at an address not corresponding to a word boundary, the trigger will never be found.**

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in subparagraph (C) (1) (ii) of the Rights in Technical Data and Computer Software Clause in DFARS 252.227-7013.
Hewlett-Packard Company, 3000 Hanover Street, Palo Alto, CA 94304 U.S.A.
Rights for non-DOD U.S. Government Departments and Agencies are set forth in FAR 52.227-19(c)(1,2).

Document Warranty

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose.

Hewlett-Packard shall not be liable for errors contained herein or for damages in connection with the furnishing, performance, or use of this material.

Safety

This apparatus has been designed and tested in according to Internationaol Safety Requirements. To ensure safe operation and to keep the product safe, the information, cautions, and warnings in this user's guide must be heeded. In addition, note the external markings on the product that are described under "Safety Symbols."

Safety Symbols

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

Hazardous voltage symbol.

Earth terminal symbol: Used to indicate a circuit common connected to grounded chassis.

WARNING

The Warning sign denotes a hazard. It calls attention to a procedure, practice, or the like, which, if not correctly performed or adhered to, could result in personal injury. Do not proceed beyond a Warning sign until the indicated conditions are fully understood and met.

CAUTION

The Caution sign denotes a hazard. It calls attention to an operating procedure, practice, or the like, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the product. Do not proceed beyond a Caution symbol until the indicated conditions are fully understood or met.

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

Product Warranty

This Hewlett-Packard product has a warranty against defects in material and workmanship for a period of one year from date of shipment. During the warranty period, Hewlett-Packard Company will, at its option, either repair or replace products that prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by Hewlett-Packard.

For products returned to Hewlett-Packard for warranty service, the Buyer shall prepay shipping charges to Hewlett-Packard and Hewlett-Packard shall pay shipping charges to return the product to the Buyer. However, the Buyer shall pay all shipping charges, duties, and taxes for products returned to Hewlett-Packard from another country.

Hewlett-Packard warrants that its software and firmware designated by Hewlett-Packard for use with an instrument will execute its programming instructions when properly installed on that instrument. Hewlett-Packard does not warrant that the operation of the instrument 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 the Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

No other warranty is expressed or implied. Hewlett-Packard specifically disclaims the implied warranties of merchantability or fitness for a particular purpose.

Exclusive Remedies

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

Assistance

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

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

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 Institute of Standards and Technology, to the extent allowed by the Institute's calibration facility, and to the calibration facilities of other International Standards Organization members.

About this edition

This is the first edition of the *HP E2465A PowerPC 604 Preprocessor Inteface User's Guide.*

Publication number
E2465-97001

Printed in USA.

Edition dates are as follows:

First edition, October 1996

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by you. The dates on the title page change only when a new edition is published.

A software or firmware code may be printed before the date. This code indicates the version level of the software or firmware of this product at the time the manual or update was issued. Many product updates do not require manual changes; and, conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.

The following list of pages gives the date of the current edition and of any changed pages to that edition.

All pages original edition

Your Comments Please

Your comments assist us in meeting your needs better. Please complete this questionnaire and return it to us. Feel free to add any additional comments that you might have. All comments and suggestions become the property of Hewlett-Packard. Omit any questions that you feel would be proprietary.

	Yes	No
1. Did you receive your product when expected?	[ ]	[ ]
2. Were you satisfied with the operation of the instrument at turn-on?	[ ]	[ ]

3. Were the proper accessories supplied with your product?

   If not, what was missing?

   Probes [ ] Manual(s) [ ] Other _____

4. What measurements will this instrument be used to make? _____

   _____

5. How will the instrument be controlled?

   Front Panel [ ] HP-IB [ ] RS-232C [ ] Controller Type _____

6. What do you like most about the instrument? _____

   _____

   _____

7. What would you like to see changed or improved? _____

   _____

   _____

8. Which manuals have you used?

   [ ] User's Guide
   [ ] User's Reference
   [ ] Programmer's Guide
   [ ] Service Guide

9. Please rate the manuals on the following:

   4=Excellent 3=Good 2=Adequate 1=Poor

   [ ] Breadth and depth of information
   [ ] Ability to easily find information
   [ ] Ability to understand and apply the information provided in the manual

   Please explain: _____

   _____

   _____

10. What is your experience with logic analyzers?

    [ ] No previous experience
    [ ] Less than 1 year experience
    [ ] More than 1 year's experience on one model
    [ ] More than 1 year's experience on several models

Name _____ Company _____

Address _____

_____ Zip Code _____

Phone _____ Instrument Serial # _____

THANK YOU FOR YOUR HELP

NO POSTAGE NECESSARY IF MAILED IN U.S.A.

Your cooperation in completing and returning this form will be greatly appreciated. Thank you.

FOLD HERE

HEWLETT PACKARD

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 1303 COLORADO SPRINGS, COLORADO

POSTAGE WILL BE PAID BY ADDRESSEE

HEWLETT-PACKARD
Colorado Springs Division
Attn: Publications Dept.
P.O. Box 2197
Colorado Springs, Colorado 80901-9959

Hewlett-Packard
Printed in the USA