HEWLETT
PACKARD

User's Guide

Coordinated
Measurement Bus

HP 64700 Emulators Terminal Interface: CMB User's Guide

HEWLETT
PACKARD

Edition1

64306-90901
E0388
Printed in U.S.A. 3/88

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.

Logic Systems Division
8245 North Union Boulevard
Colorado Springs, CO 80920, U.S.A.

Printing History

New editions are complete revisions of the manual. The date on the title page changes only when a new edition is published.

A software code may be printed before the date; this indicates the version level of the software product at the time the manual was issued. Many product updates and fixes 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 revisions.

Edition 1	11/87	64306-90901	E1187
Edition 2	3/88	64306-90901	E0388

Using This Manual

This manual, the *HP 64700 Emulators Terminal Interface Coordinated Measurement Bus User's Guide*, explains how to make synchronous measurements between HP 64700 Series emulators/analyzers with the built-in Terminal Interface CMB firmware. It covers use of the CMB (and BNC), including:

- An Introduction - Chapter 1
- Function Of The CMB - Chapter 1
- CMB Requirements And Limitations - Chapter 1
- CMB Command Descriptions - Chapter 1
- Making CMB Connections - Chapter 2
- How The CMB Operates - Chapter 3
- Example Measurements - Chapter 3
- Transparent Configurations - Chapter 4
- Using The IMB And CMB - Chapter 5
- Status And Error Messages - Appendix A

The index contains terms and corresponding page numbers so that you can locate information quickly.

If you do not understand a term in this manual, refer to the *HP 64700 Emulators Glossary Of Terms* for a definition.

You may want to use this manual in conjunction with several others, including:

- *HP 64700 Emulators Terminal Interface User's Reference*
- *Terminal* or *PC Interface Emulator User's Guides*
- *Terminal* or *PC Interface Analyzer User's Guides*

The types of measurements you plan to make will determine which manuals you should use.

Refer To The Maps

The HP 64700 Series Manual Maps will lead you in the right direction for getting started with the various interfaces, and with using your emulator/analyzer. You can find the maps in the package marked Read Me First.

Contents

Chapter 1 Introducing The CMB

Chapter 2 Installation

Chapter 3 How The CMB Operates

Chapter 4 Transparent Configurations

Chapter 5 Using The IMB And CMB

Appendix A Status And Error Messages

Illustrations

Tables

Notes

1

Introducing The CMB

What Is In This Chapter?

- What Is The Coordinated Measurement Bus (CMB)?
- CMB Command Descriptions
- Getting Help On The CMB Commands
- Using The BNC Connector

What Is The Coordinated Measurement Bus (CMB)?

The Coordinated Measurement Bus (CMB) is a connection between multiple HP 64700 emulators/analyzers. CMB commands in the HP 64700 firmware are used to drive signals to and receive signals from any of the HP 64700s connected to this bus. The CMB is used to couple multiple HP 64700 Series emulators/ analyzers for making coordinated measurements, and to synchronize measurements between these HP 64700s.

Function Of The CMB

The CMB provides the means for coordinating measurements between multiple HP 64700s, or between an HP 64700 and another instrument. For target systems that contain multiple microprocessors, multiple HP 64700 emulators can perform synchronous runs, emulator breaks into the monitor, and HP 64700 analyzers can cross-trigger on the CMB. An HP 64700 Series analyzer can trigger or be triggered by some other instrument, such as the HP 1630 Logic Analyzer, via the Trigger signal on the BNC connector. Another instrument can also cause the emulator to break into the monitor by driving the BNC connector.

For example, you can set up the following types of measurements:

An emulator stops executing a user program when the emulation analyzer in another HP 64700, finds a specified trigger.

An external analyzer causes an HP 64700 Series emulator to break into the monitor by driving the BNC connector.

CMB Requirements And Limitations

You can connect 32 emulators to the CMB. Table 1-1 shows some rules for connecting multiple emulators to the CMB. All of those emulators can then participate in CMB measurements.

To connect more than 16 emulators to the CMB, your HP Customer Engineer must make the modification.

Table 1-1. Requirements For Emulators On The CMB

Number of HP 64700 Series Emulators:	Total length of CMB cable:	Restrictions on the CMB connection:
2 to 8	100 meters	None.
9 to 16	50 meters	None.
9 to 16	100 meters	Only 8 emulators may have rear panel pullups connected. *
17 to 32	50 meters	Only 16 emulators may have rear panel pullups connected. *

* Modification must be done by HP Customer Engineer.

All of the HP 64700 Series emulators connected to the CMB must be powered up before proper operation of the entire CMB configuration can be assured.

Each HP 64700 rear panel contains one CMB connector. If you want to chain multiple emulators to make synchronous measurements, we recommend that you use the HP 64023A cable. Refer to the *HP 64700 Emulators Hardware Installation And Configuration* manual for details, then return here.

Any HP 64700 physically connected to the CMB may participate in CMB interactive measurements. Through user configurations, you determine if the emulator, the analyzer, or both participate in CMB interactions.

CMB And Foreground Monitors

The CMB does not work with emulators running a foreground monitor. To operate your emulator (68000 or 80186), with the CMB, it must use a background monitor. While running a foreground monitor during CMB operation, if the emulator is instructed to break during a measurement, it will only temporarily break to the foreground monitor, but will not remain there. The emulator will then try to return to whatever operation it was performing before the break occurred.

Maximum CMB Cable Lengths

The available CMB cable is not supplied with every emulator. You must order it separately by part number 64023A. This cable is approximately 4 meters long. You can build your own compatible CMB cables using standard 9-pin D subminiature connectors and 26 AWG wire.

Note

Hewlett-Packard does not ensure proper CMB operation if you are using a self-built cable!

The entire length of CMB cable that you can use to connect all of the HP 64700 Series emulators in a measurement can be 100 meters long. See the *HP 64700 Emulators Hardware Installation And Configuration* manual for CMB cable specifics.

CMB Background Information

There are three bi-directional signal lines on the CMB, as well as an associated BNC connector on the rear panel of the emulator. These CMB signals are:

CMB Trigger line

READY line

EXECUTE line

The CMB Trigger line is not affected by the **cmb** command. The **cmb** command enables/disables interaction on the READY line, and also allows EXECUTE to start an emulation run.

The CMB Trigger line is low true. It may be driven or received by any HP 64700 on the CMB. This signal also goes true briefly following the receipt of an EXECUTE signal.

Note

Because of this, if the EXECUTE function is being used, the analyzer trigger should not be used to trigger external instruments.

CMB Trigger can be used directly as a break source to the emulator, or can indirectly be a break source through the internal trigger lines. When used as a break source, the driving function must be cleared before the emulator can resume running.

The CMB READY line is high true. It is open collector and performs an ANDing of the ready state of all enabled emulators on the CMB. Each emulator on the CMB releases this line when it is ready to run. This line goes true when all enabled emulators are ready to run, providing for a synchronized start. When CMB is enabled, each emulator is required to break when CMB READY goes false, and will wait for CMB READY to go true before returning to the run state. When an enabled emulator breaks, it will drive CMB READY false. The emulator that drives CMB READY false holds it false until it is ready to resume running. When the emulator is reset, it also drives CMB READY false.

When an emulator will be ready to run depends on the cause of the break. For example, if the break is temporary, such as a display of registers on one emulator, all other emulators on the CMB will temporarily break, then resume running when the register command has completed. Break conditions that cause the emulator to remain in the monitor, such as the **b** command, require you to intentionally request the emulator to run or break on receiving the emulation or external analyzer trigger.

The CMB EXECUTE line is low true. Any HP 64700 on the CMB can drive this line. It serves as a global interrupt, and is processed by both the analyzer and the emulator. The analyzer, if **tx** is enabled, will start a new trace upon receipt of a CMB EXECUTE signal. The emulator, if CMB is enabled, will receive a break, then attempt to run from the address specified by the **rx** command, when CMB READY returns true. The default address, if no "rx" address is specified, is the current program counter value.

Note These three CMB signals should only be used with a properly constructed CMB cable.

The BNC input can either drive the analyzer trigger, receive an analyzer arm, receive an analyzer trigger, or receive a break request for the emulator. The BNC input is edge sensitive, and will accept a positive pulse with a minimum width of 25 nS.

The BNC connector is designed to be used with 50-ohm coaxial cable.

Comparison Between The CMB And BNC Triggers

The CMB Trigger and the BNC Trigger lines have the same logical purpose: to provide a means to connect the internal trigger signals (trig1 and trig2) to the outside world. The CMB Trigger and the BNC Trigger lines are bidirectional. Either may be used directly as a break condition. Both are configured through Terminal Interface commands. The CMB Trigger is level sensitive, while the BNC Trigger is edge sensitive. The CMB Trigger line will output a true pulse following the receipt of EXECUTE, regardless of the commands you used to configure it. This pulse is internally ignored. The BNC Trigger line activity always reflects the commands you used to configure the BNC Trigger.

In addition, either of these lines can cause a break when connected to an internal trigger line (trig1 or trig2) and that line is enabled to break.

CMB Command Descriptions

General descriptions of the CMB commands are included here. For detailed descriptions, refer to the *HP 64700 Emulators Terminal Interface User's Reference.*

CMB Run Control Commands

These are the CMB run control commands:

cmb (enables/disables CMB run control interaction)

rx (starts an emulator running upon CMB execute)

bc (sets or displays emulator break conditions)

When "cmb" is enabled, the emulator interacts with the activities on the CMB. The emulator will break when READY goes false, and will not run until READY goes true again. The emulator will drive READY false when it breaks or is reset, and will not release it until it is ready to run. If READY is false when "cmb" is enabled, the emulator will be instructed to break. When "cmb" is disabled, the emulator always drives READY true.

The **rx** command allows you to specify an address from which the emulator will run, following the receipt of an EXECUTE signal. If an address is not specified with the **rx** command, the emulator will run from the current program counter value. If "cmb" is disabled, the emulator will ignore the EXECUTE signal. The **rx** command automatically turns on CMB interaction by internally executing a **cmb -e** command.

The **bc** command can enable either the BNC Trigger or the CMB Trigger to initiate a break. The CMB trigger is level sensitive. If the CMB trigger is true when it is enabled as a break condition, it will immediately break the emulator.

The BNC Trigger and CMB Trigger can also initiate a break indirectly, if connected to trig1 or trig2, when trig1 or trig2 are enabled.

CMB Analyzer Control Commands

These are the analyzer control commands:

cmbt (specifies drivers and receivers of the CMB trigger signal)

bnct (specifies drivers and receivers of the BNC trigger signal)

tarm (specifies the arm condition for the analyzer)

tgout (specifies which signals, trig1 or trig2 or none, are driven after analyzer trigger is found)

tx (starts an emulation analyzer measurement when the emulator receives CMB EXECUTE)

The **cmbt** command allows the CMB trigger signal to be driven, and/or to drive either of the two internal analyzer trigger signals (trig1 and trig2).

The **bnct** command allows the BNC trigger signal to be driven, and/or to drive either of the two emulation analyzer trigger signals (trig1 and trig2).

The **tarm** command allows the emulation analyzer to receive either of the two emulation analyzer trigger signals.

The **tgout** command directs the trigger output of the analyzer to the trig1 or trig2 lines.

The **tx** command, when enabled, instructs the emulation analyzer to restart its measurement upon receipt of an EXECUTE signal.

CMB Global Command

x (initiates a synchronous CMB execution)

The **x** command will always cause an EXECUTE signal to be sent to all HP 64700 Series emulators physically attached to the CMB, and will always cause an internal EXECUTE to occur.

Details About The CMB Commands

Refer to the *HP 64700 Emulators Terminal Interface User's Reference* manual for details about any of these commands.

Getting Help On The CMB Commands

To get help on any of the CMB commands, just type **help <command>**. For example, to display help for the **cmbt** command, you would type: **help cmbt**

The short version of help information for the command is displayed on screen, along with available options. If you need additional details, refer to the *HP 64700 Emulators Terminal Interface User's Reference* manual.

Using The BNC Connector

You can make measurements using an external logic analyzer or oscilloscope connected to the BNC connector on the back of the HP 64700. Because this line is bi-directional, you can use the BNC connector to drive another device, such as an external logic analyzer, or you can receive an arm from this connector. The BNC input is rearmed whenever its configuration is changed.

The **bnct** command specifies which of the internal trigger signals (trig1 and/or trig2) will drive and receive the BNC Trigger signal. This allows for flexible interconnections with the emulation and external analyzers, and the CMB Trigger.

The BNC connector is capable of driving TTL level signals into a 50-ohm load. It requires a driver than can supply at least 4 mA at 2 volts when used as a receiver. The BNC connector is configured as an open-emitter structure which allows for multiple drivers to be connected. It can be used for cross triggering between multiple HP 64700s when no other cross-measurements are needed. The output of the BNC connector is short-circuit protected, and is protected from TTL level signals when the emulator is powered down.

Notes

2

Installation

Making The CMB Connections

When connecting multiple HP 64700s for making measurements, you must connect all of the emulators together. Make sure that all of the HP 64700s included in the measurement(s) you are planning to make are connected together with CMB cables. Chapter 1 contains basic rules and limitations for connecting multiple emulators to the CMB. Chapters 3 and 4 contains additional information on example CMB setups, and making example measurements. Refer to the *HP 64700 Emulators Hardware Installation and Configuration* manual for details on making the cable connections. Then return here.

About The CMB Cable

The CMB cable part number is 64023A. You can find more information about the CMB cable in the *HP 64700 Emulators Hardware Installation And Configuration* manual.

About The HP 64306A

For details about installing and running Performance Verification on the HP 64306A IMB/CMB Interface, which is used to make IMB-CMB measurements, refer to the Installation Notice supplied with the HP 64306A, and chapter 5 in this manual.

Power And Connection Requirements

All HP 64700 Series emulators/analyzers physically connected to the CMB must be powered up to ensure correct operation. The rules listed below will help you know that your instruments are operating correctly on the CMB.

When connecting to the CMB and powering up, either:

> Apply power to all HP 64700s first, then connect each powered up HP 64700 to the CMB.
>
> Or, first power down all HP 64700s involved. Connect all powered down HP 64700s to the CMB. Then sequentially power up each HP 64700.

Caution ✋

If you connect the CMB cable to an HP 64700 that is powered down, all other HP 64700s that were operating properly on the CMB will no longer be functional. You must then cycle power on all HP 64700s connected to the CMB to resume normal operation.

When disconnecting from the CMB, and powering down:

> With power already applied to the HP 64700, remove the CMB cable from the HP 64700 that you want to power down. Then power down that HP 64700.

Caution ✋

When you have achieved a functional CMB connection, do not connect an HP 64700 that is powered down to the CMB! This can result in unpredictable operation. Also, do not power down an HP 64700 that is already connected to the CMB, or else unpredictable operation of all other HP 64700s connected to the CMB will occur.

3

How The CMB Operates

Topics In This Chapter

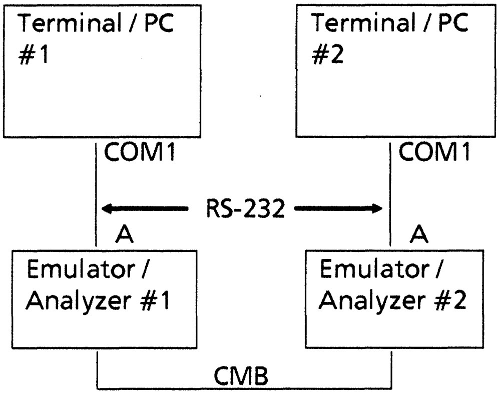
- CMB Configurations
- Example CMB Measurements
- Displaying Results Of Measurements
- Making Repetitive Measurements
- How To Halt A Measurement
- Using Command Files To Make Measurements
- Using Macros
- CMB Command Syntax
- What To Do If Problems Occur

CMB Configurations

You may be using any one of the following three types of configurations for making coordinated measurements.

Figure 3-1 shows an example CMB configuration using two terminals. Each terminal connects to an emulator/analyzer through communication port #1 (COM1). Both connections are made with

RS-232 cables (you could also use RS-422 cables). With this configuration, to execute CMB commands for one emulator, you enter the commands on the terminal connected to it.

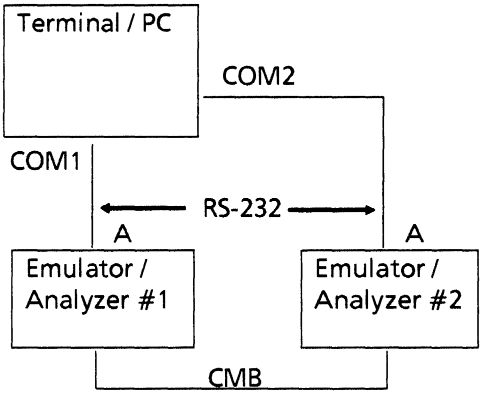


"A" indicates an emulator

Figure 3-1. CMB Setup Using Two Terminals

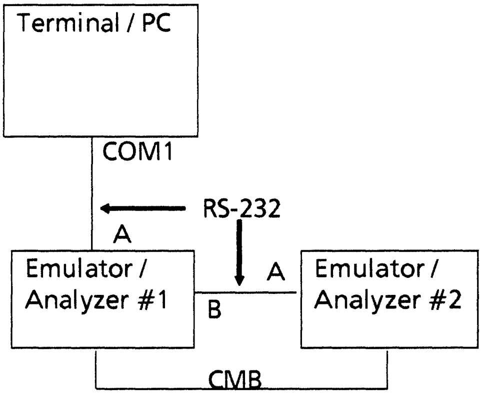
Figure 3-2 shows an example CMB configuration using a single terminal or Personal Computer (PC). Two emulators are connected to two ports on the computer. To execute commands for one of the emulators, you must be communicating with the correct port.

Figure 3-3 shows an example CMB configuration using a single computer. The emulators are connected in series. You must use transparent configuration codes to communicate with each emulator. This is covered in chapter 4.



"A" indicates an emulator

Figure 3-2. CMB Setup Using Two Terminal Ports



"A" and "B" indicate emulator

Figure 3-3. CMB Setup Using One Terminal Port

Note

If you don't understand the functions of the CMB signals, make sure you have read chapter 1 before continuing.

Example Measurements

Some example measurements that will help familiarize you with the capabilities of the CMB are listed here and explained in detail on the following pages.

1. Synchronously starting and breaking emulators.
2. Enabling and disabling emulator CMB interaction.
3. Emulator response to EXECUTE.
4. Cross-triggering HP 64700 analyzers.
5. Analyzer response to EXECUTE.
6. Break an emulator and trigger the analyzer using BNC Trigger.

You can combine some commands, such as using the **rx** and **tx** commands at the same time. The examples in this chapter are simplified by using the commands separately.

Example #1

Synchronously starting and breaking emulators.

This example uses two HP 64700s connected to the CMB, with user programs loaded into memory.

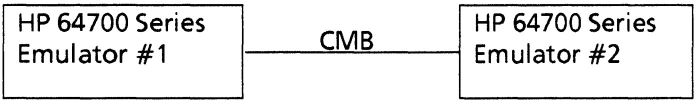


Figure 3-4. Example #1

Even though both emulators are connected to the CMB cable, neither can participate in CMB measurements until you enable them.

Each emulator should be reset or executing in the monitor, with a user program loaded into memory, the stack set up, and the program counter set to an address in the user program.

To allow both emulators to participate in CMB measurements, send the command **cmb -e** to (type **cmb -e** on) each emulator.

To set up #1 to run, type **r** on that emulator.

Observe the status on #1 by typing **es**. The status will indicate that the emulator is waiting for CMB to become ready.

To start both emulators running, type **r** on #2. Observe the status on both emulators to make sure they are running.

To break both emulators at once, type **b** on either emulator.

Observe the status of each emulator (with the **es** command) to see that they have both stopped running. The one you typed **b** on will be executing in the monitor, and the other will be waiting for CMB to become ready.

To start both emulators running again, type **r** on the emulator that is currently executing in the monitor.

You can observe the status of each emulator to see that they are both running.

Example #2

Enabling and disabling emulator CMB interaction.

This example uses three emulators connected to the CMB.

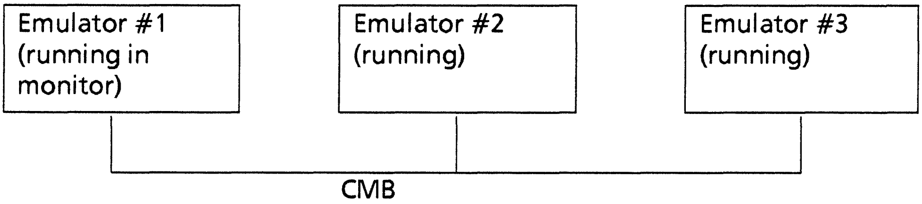


Figure 3-5. Example #2

Let's assume that #1 is enabled to participate in CMB measurements, but is currently running in the monitor.

Let's also assume that #2 and #3 are disabled on the CMB, and are both running a user program.

Enable #2 on the CMB by typing **cmb -e**.

Emulator #2 will break into the monitor, and await a return to running the user program.

To start #1 running, type **r** on that emulator.

When #1 begins to run, #2 will start running.

Enable #3 on the CMB by typing **cmb -e**. It will continue running.

Disable #2 by typing **cmb -d**.

Break #2 by typing **b**.

Emulators #1 and #3 will continue running. Observe status.

Enable #2 by typing **cmb -e**. #1 and #3 will have broken, and are in the wait state.

Example #3

Emulator response to EXECUTE.

This example uses three emulators connected to the CMB.

All three emulators are enabled to participate in CMB measurements. All three are currently running.

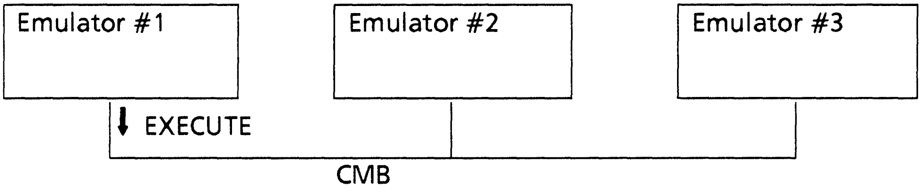


Figure 3-6. Example #3

To set up #1 to run from a specified address when the CMB EXECUTE signal goes true, type **rx 1000h** on #1. In this example, 1000h is the address at which the emulator will begin executing following EXECUTE.

Cause #1 to emit the EXECUTE signal by sending **x** to (typing **x** on) #1. Any HP 64700 can emit the EXECUTE signal.

All emulators will break.

Emulator #1 will set its program counter to 1000h. Because there is no **rx** command specified on #2 and #3, their program counters are not altered. All emulators will begin running when READY goes true.

Reset #1 by typing **rst**. #2 and #3 will break.

Cause #2 to emit the EXECUTE signal by typing **x** on #2. #1 will begin running from 1000h, and #2 and #3 will resume running, when READY goes true.

Example #4

Cross triggering HP 64700 analyzers.

Figure 3-7 shows the connections between the CMB and BNC trigger signals, the internal trigger signals (trig1 and trig2), within the HP 64700.

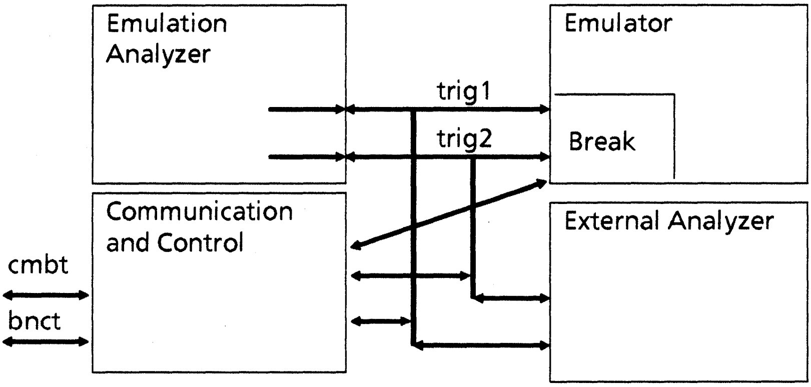


Figure 3-7. Internal Analyzer Trigger Signals

This example cross triggers two analyzers connected on the CMB. The HP 64700s are connected as shown in figure 3-4. Analyzer #1 will find its trigger condition, then trigger analyzer #2.

On #1, establish a valid trigger specification using the **tg** command. Execute **tgout trig1** and **cmbt -r trig1** on #1 to output the analyzer trigger to the CMB Trigger line.

On #2, establish CMB Trigger as the analyzer trigger by typing **cmbt -d trig1** and **tarm = trig1** and **tg arm**.

Note

The command **tgout trig1** causes the analyzer to drive the trig1 signal. The command **cmb -r trig1** allows the CMB interface to receive trig1, and send it to the external CMB connector.

Start a trace on analyzer #2 by typing **t**. Observe the trace status by typing **ts**. The status will indicate that the trigger is not in memory.

Start a trace on #1 by typing **t**. Observe the trace status by typing **ts**. When #1 has found its trigger, the status will indicate that the trigger is in memory. Typing **ts** on #2 will also show that the trigger is in memory. This completes the cross trigger.

An incorrect trigger on #2 can occur if #1 has previously completed a trace and is driving tgout.

Tgout stops driving when you execute a **th** (trace halt) command, and start a new trace with **t**. If tgout has caused the emulator to break as a result of driving tgout, it will also stop driving after the emulator responds to the break request. A status request, while the emulator is running in the monitor during a valid tgout break request, will also clear tgout.

When this is connected to trigger #2, analyzer #2 will trigger immediately when the trace is started. To prevent this, use tx instead of **t** to start the trace, as shown in the next example.

More Information About Analyzers

To find out more about how the analyzers interact when making coordinated measurements, refer to the *HP 64700 Analyzer User's Guide.*

Example #5

Analyzer response to EXECUTE.

This example uses two HP 64700s connected on the CMB as shown in figure 3-8. This example performs a similar measurement to example #4.

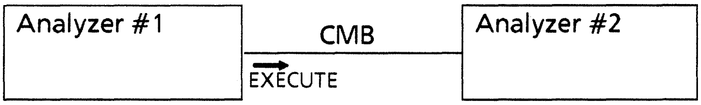


Figure 3-8. Example #5

On #1, establish a valid trigger specification using the **tg** command. Execute **tgout trig1** and **cmbt -r trig1** on #1 to output the analyzer trigger to the CMB Trigger line.

On #2, establish CMB Trigger as the analyzer trigger by typing **cmbt -d trig1** and **tarm = trig1** and **tg arm**.

Initiate a trace on analyzer #2 by typing **tx -e**. Observe the trace status by typing **ts**. The trace has not yet started.

Initiate a trace on #1 by typing **tx -e**. Observe the trace status by typing **ts**. The trace has not yet started.

Start both analyzers by typing **x** on either HP 64700. The CMB protocol ensures that all drivers of CMB Trigger are disabled, all analyzers receiving CMB Trigger are started, then all drivers are started.

When #1 has found its trigger, the status will indicate that the trigger is in memory. Typing **ts** on #2 will also show that the trigger is in memory. This completes the cross trigger.

Example #6

Break an emulator and trigger the analyzer using BNC Trigger.

This example uses one HP 64700.

Connect an external device, that is capable of driving a positive edge signal, to the BNC connector on the HP 64700.

Load and begin running a user program.

Enable BNC Trigger to break the emulator by typing **bc -e bnct**.

Have the external device drive the BNC Trigger. Observe the emulation status by typing **es**. The emulator will be executing in the monitor. The cause of the break will be the BNC Trigger.

Disable BNC Trigger from breaking the emulator by typing **bc -d bnct**. Resume the emulator running the user program by typing **r**.

Enable the analyzer to trigger on the BNC Trigger by typing **bnct -d trig1**. Arm the analyzer by typing **tarm = trig1**. Define the trigger as the arm by typing **tg arm**.

Start a trace by typing **t**. Observe the trace status. The trigger is not in memory.

Have the external device drive the BNC Trigger. Observe the trace status by typing **ts**. The trigger will now be in memory.

Note

You can also use the BNC Trigger as an output. When connected to the internal trigger signals (trig1 and/or trig2), BNC Trigger will be driven to an external device when the internal trigger signal goes true.

Some CMB Details You Should Know

CMB Trigger, EXECUTE, and READY are the only signals on the CMB.

Internal trigger signals trig1 and trig2 can drive arm to and receive Trigger from the emulation analyzer, and external analyzer. These signals may also drive and/or receive the BNC and CMB Triggers.

A CMB Execute pulse on one emulator/analyzer starts all enabled emulators and analyzers in the CMB configuration running. They will continue running until instructed otherwise.

When you cause one emulator/analyzer in the CMB configuration to break to the monitor, all other emulators break to the wait state, and remain in the wait state until instructed otherwise.

When a break condition is enabled by a **bc -e** command, any pending breaks on that condition are cleared. An immediate break will take place only if CMB Trigger is tied to that break condition, and is being driven externally.

Displaying Results Of Measurements

There are several ways you can display results of measurements that you make:

1. The **es** command displays the current state of the emulator.

Typing es reveals the current status of emulation activity. If the prompt has changed to "W>", the emulator is waiting for the CMB Trigger to become ready.

2. The **ts** command displays the emulation trace status.

Typing **ts** displays the current status of the emulation analyzer. The resulting trace status listing will indicate whether or not the CMB trace was completed, halted, or is still running.

Refer to the *HP 64700 Emulators Terminal Interface User's Reference* manual for additional details on these two commands.

Breaking The Emulator When It Receives CMB Trigger

If you configure the emulator to break when it receives a CMB Trigger signal, after the emulator receives the CMB Trigger signal, a status message will be printed indicating that the emulator broke to the monitor because it received CMB Trigger.

You set this up by executing the command: **bc -e cmbt**

Making Repetitive Measurements

To make repetitive CMB measurements, use the **rep** command.

The rep command allows you to repeat a command or group of commands a specified number of times. You can specify that the command(s) repeat indefinitely by specifying 0 as the repeat count.

For example:

To have an emulator emit a CMB Execute pulse every 10 seconds for 20 repetitions....

TYPE: **rep 20 {x;w 10}**

To have an emulator repeat this same command indefinitely....

TYPE: **rep 0 {x;w 10}**

Refer to your *HP 64700 Emulators Terminal Interface User's Reference* manual for details on the **rep** command.

How To Halt A Measurement

The easiest method for halting an analyzer measurement is to halt the trace by typing **th**. In fact, if the analyzer trigger has not been found, you must use the **th** command to halt the analyzer before you can display the trace list.

To halt a CMB measurement, for example, you could have an emulator break to the monitor upon occurrence of a certain condition. This halts the CMB measurement by driving READY false. You can manually cause an emulator break by typing **b**.

There is no other "halt" command that you can execute to halt a measurement.

Using Command Files To Make Measurements

Command files are source files that you create. They allow you to accomplish and duplicate activities without having to enter all of the commands manually

You can execute command files in two ways:

1. using the **po** command
2. from the PC Interface

Using The po Command

The command **po -f** allows the HP 64700 to read a command file from the port not currently acting as the command source port. This is useful for configurations where a terminal is connected to one port on the HP 64700, and a host computer is connected to the other port.

When you execute **po -f**, the HP 64700 is immediately put into transparent "mode", allowing communication with the host computer. You can now issue a command to the host computer which sends the command file to the HP 64700.

For example, you could enter:

po -f

cat cmdfile <ESC> <CHAR>

Refer to the *HP 64700 Emulators Terminal Interface User's Reference* manual for additional details about the **po** command.

Using The PC Interface

If you are using the PC Interface, this section applies to you.

Note

Using PC Interface command files implies that your system setup includes a Personal Computer (HP Vectra or IBM compatible) or another supported host computer, and that you are operating the PC Interface. If you are using only a terminal to control your HP 64700 Series emulator(s), you cannot use PC Interface command files, and should skip this section.

In PC Interface command files:

- "!" indicates a wait period of 1 second.
- "@" in column 1 specifies data.
- "#" preceding text signifies a comment.
- You can add spaces between multiple parameters on one line.
- Pressing ESC terminates command file execution.
- You can nest 8 levels of command files.

The following lines make up a command file. It contains some of the commands executed on emulator # 1 in the first example shown earlier:

```
cmb -e
r
es
b
```

You can execute PC Interface command files in two ways:

1. You can specify the command file name when you access the PC Interface.

2. You can use the PC Interface command file option from within the PC Interface.

Let's suppose you want to execute the command file shown in the first example. If you have not already started the PC Interface, you would specify the command file name when accessing the emulator by way of the PC Interface. For example, if you are using the Z80 emulator you would type:

pcz80 /c <command_file> z80emul

Note

If you are using a Personal Computer or host computer, refer to the *HP 64700 Emulators PC Interface User's Reference* for additional details about creating and using command files.

Using Macros

In the Terminal Interface, you can create macros that will execute multiple commands for you simply by typing the macro name. To do this, use the **mac** command.

For example:

To create a macro named "cmbgo"....

TYPE: **mac cmbgo={xp -c 1; xp -e; xp -c 2; cmbt -r trig1; bc -e trig1}**

To execute the macro....

TYPE: **cmbgo**

Refer to the *HP 64700 Emulators Terminal Interface User's Reference* for more information about the **mac** command.

CMB Command Syntax

Syntax diagrams and detailed descriptions of the CMB commands mentioned earlier in this chapter are included in the *HP 64700 Emulators Terminal Interface User's Reference.*

What To Do If Problems Occur

If you encounter any problems while using the CMB features, do the following:

1. Read the command descriptions for the commands you are using. You can find all of the Terminal Interface commands described in the *HP 64700 Emulators Terminal Interface User's Reference.*

2. Verify that all connections of the CMB cables are stable.

3. Refer to the *HP 64700 Emulators Hardware Installation And Configuration* manual to make sure that you are using the correct CMB cables.

4. Verify that you have the desired connectivity for all of the resources involved in the measurement. This means that you should have the proper command setup for the **bnct**, **cmbt**, **tgout**, **xtgout**, **tarm**, and **xtarm** commands.

5. Verify that the **cmb**, **rx**, **tx**, and **bc** commands are properly enabled or disabled. Errors in the command setup can cause unexpected actions in measurements.

6. If you are using just a terminal with the emulator/analyzer, cycle power on the emulator/analyzer.

7. Refer to the *HP 64700 Emulators Support Services* manual to find the phone number for the closest HP Response Center. Then call them directly.

Notes

4

Transparent Configurations

Topics In This Chapter

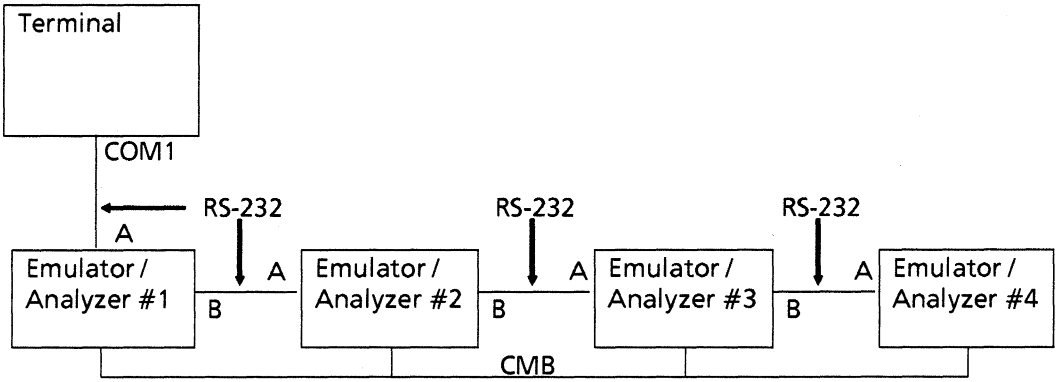
- Setting Up The Communication Paths
- How To Communicate With Each Emulator/Analyzer
- Making Measurements
- Where To Find More Information

Setting Up The Communication Paths

If you have multiple HP 64700 emulators/analyzers connected in a serial configuration, before you can configure them in a CMB measurement, you must allow each of them to recognize the commands that you are going to execute, and the commands that they will receive during the measurement. To do this, you must set up a communication path for each HP 64700 involved in the CMB measurements.

For the examples in this chapter, up to four emulators/analyzers are used.

Depending on how many emulator/analyzer systems YOU are using, your configuration should look something like figure 4-1.



A and B are emulator ports.

Figure 4-1. An Example CMB Configuration

A Bit Of Background On Transparent Configurations

When you configure multiple emulators/analyzers on the CMB, you must have a method for communicating with each of them. We call this mode of communication "transparent". You can read more about this in the *HP 64700 Emulators Hardware Installation And Configuration* manual.

In this configuration, when you are communicating with (executing commands on) one emulator/analyzer enabled on the CMB, none of the other emulators/analyzers recognize any of those commands. Therefore they are "transparent" to those commands.

A transparent configuration can be used to:

1. Provide a communication path between a terminal and a host computer, by way of an HP 64700.

2. Download absolute files and command files from a host computer into the HP 64700 emulator.

3. Stop communicating with one emulator/analyzer (HP 64700) and begin communicating with a second, or a third, or a fourth, and so on. You can communicate with one HP 64700 at a time,

and keep all other HP 64700s transparent to the commands you execute, until you request to communicate with them.

The function mentioned in the third method is the one used for making CMB measurements. We will not use the first two methods here.

Note

Refer to the *HP 64700 Emulators Terminal Interface User's Reference* manual for details about the **xp** command.

Now let's get started.

To learn how to set up transparent communications, just follow the steps below.

Note

With the **xp** command, you can define a two-character escape code. The default code is **<ESC> g**. The **-f** option allows you to change the primary escape code character (<ESC>). The **-s** option allows you to change the secondary escape code character (g). The values used to define the escape code characters must be specified as a hexadecimal number. For example, the command to define the default escape code is: **xp -f 01b -s 067**

The **-e** option enables transparency on an HP 64700. The **-d** option disables transparency on an HP 64700.

For the examples in this chapter, we will keep the primary escape code character as <ESC>, and will only redefine the secondary character for each emulator (with **xp -s <character>**).

1. At the terminal or host computer prompt type **xp -s 31** and press **Enter**. The value "31" is the hexadecimal equivalent for the number "1".

   What occurred? You have just set up a secondary escape character for the first HP 64700 emulator/analyzer in your configuration. This means that each time you access this HP 64700 for communication, you will designate it as "1" (with **ESC 1**).

Note

You can define 2 characters for the transparent configuration escape code. The default secondary escape code character is "g" (067 hexadecimal). If you are including more than two HP 64700s in a measurement, you must give each one a unique secondary escape code character so that you can escape to any HP 64700 at any time. By doing this, you can be sure that you have achieved a desired communication connection.

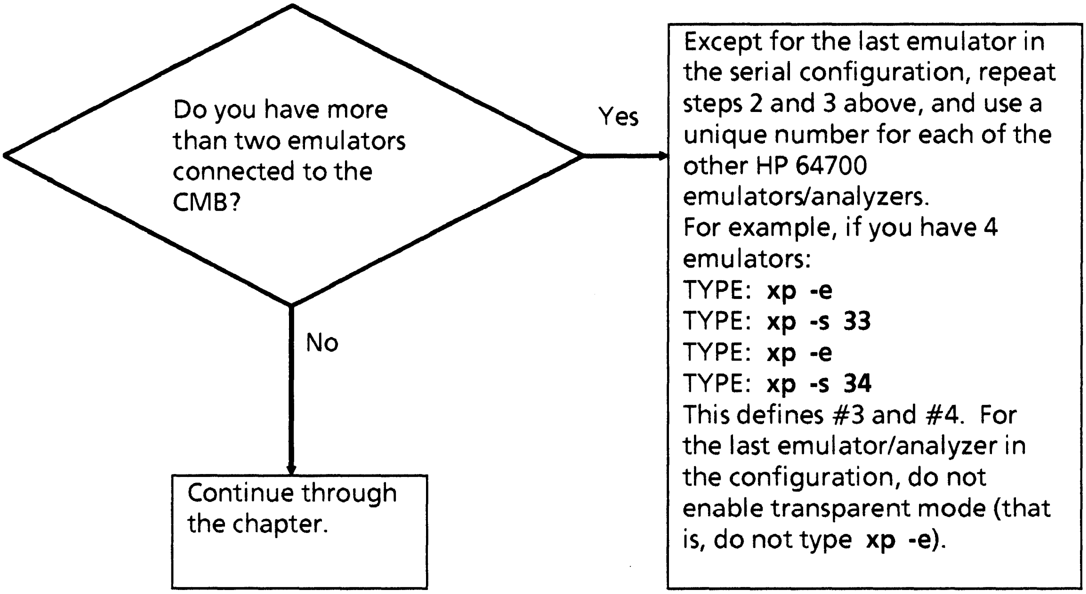
---

2. Begin communication with the second emulator/analyzer by typing **xp -e** and pressing **Enter**.

   What occurred? The first HP 64700 is now in transparent mode. We will use the term "transparent mode" whenever an emulator is in this state.

   You are now communicating directly with the second HP 64700. The first emulator/analyzer will not recognize any commands you type until you escape to it. You'll find out how to escape to another emulator/analyzer shortly.

3. Set up a secondary escape code character for the second emulator/analyzer by typing **xp -s 32** and pressing **Enter.** The value "32" is the hexadecimal equivalent for the number "2".



Do you have more than two emulators connected to the CMB?

Except for the last emulator in the serial configuration, repeat steps 2 and 3 above, and use a unique number for each of the other HP 64700 emulators/analyzers.

For example, if you have 4 emulators:

TYPE: **xp -e**
TYPE: **xp -s 33**
TYPE: **xp -e**
TYPE: **xp -s 34**

This defines #3 and #4. For the last emulator/analyzer in the configuration, do not enable transparent mode (that is, do not type **xp -e**).

Continue through the chapter.

Figure 4-2. How Many Emulators Are Connected To CMB?

A Summary Of What Is Going On

For this example (of four emulators/analyzers in the CMB configuration), here is a summary of how you would set up the communication path for each emulator/analyzer, and what occurs.

What You Typed	What Occurred?
Emulator/Analyzer #1:	
xp -s 31	Set up escape character for #1
xp -e	Put #1 in transparent mode Started communication with #2
Emulator/Analyzer #2:	
xp -s 32	Set up escape character for #2
xp -e	Put #2 in transparent mode Started communication with #3
Emulator/Analyzer #3:	
xp -s 33	Set up escape character for #3
xp -e	Put #3 in transparent mode Started communication with #4
Emulator/Analyzer #4:	
xp -s 34	Set up escape character for #4

You do not enable transparent mode for the last HP 64700 in the configuration. Why? Because #4 would try to communicate with #5, which doesn't exist.

If you unintentionally enabled transparent mode on #4, when you press **Enter** you would probably find that nothing happens because there is nothing connected to the second port of emulator #4. To resolve this:

PRESS: **ESC** and then type **4**

TYPE: **xp -d**

This will break the connection of #4 trying to continue the communication path to nowhere. You are now communicating with #4.

How To Access Each HP 64700

When communicating with each HP 64700 you can either:

1. Use the escape code to access an emulator/analyzer to execute multiple commands. This involves stopping communication with one emulator/analyzer, and starting communication with another.

2. Use the escape code to temporarily access an HP 64700 emulator/analyzer to execute a single command. This involves continuing communication with one HP 64700 while temporarily communicating with another.

Note

If you want to continue communicating with one HP 64700 emulator/analyzer, but want to execute a single command on another emulator/analyzer, use method #2. The key is to not disable transparent mode on the emulator/analyzer with which you are currently communicating. Just temporarily access the other emulator/analyzer to execute the command.

Execute Multiple Commands

When accessing an emulator/analyzer to execute multiple commands you will:

PRESS: **ESC** (don't press Enter)

TYPE: **<emulator designator>** (don't press Enter)

TYPE: **xp -d**

For example:

With our example configuration, assume you are currently communicating with emulator/analyzer #4, but want to access #2 to execute multiple CMB commands. You would:

PRESS: **ESC** (don't press Enter)

TYPE: **2** (don't press Enter)

TYPE: **xp -d**

TYPE: **cmb**

TYPE: **cmb -e**

To return to communicating with #4, after you have issued these commands, you would type **xp -e**.

Execute A Single Command

When temporarily accessing an emulator/analyzer to execute a single command you will:

PRESS: **ESC** (don't press Enter)

TYPE: **<emulator designator>** (don't press Enter)

TYPE: **<command to execute on that emulator>**

For example:

With the example configuration shown earlier, assume you are communicating with emulator/analyzer #4, but want to execute a single CMB command on #2. You would:

PRESS: **ESC** (don't press Enter)

TYPE: **2** (don't press Enter)

TYPE: **cmb -e** and press **Enter**

Note

You are still communicating directly with emulator/analyzer #4 because you did not disable transparent mode on #2 (you did not type **xp -d** after you temporarily accessed #1). Therefore, even when executing this command on #2, it is still in transparent mode.

Making Measurements

Imagine that you have two HP 64700 Series emulators/analyzers configured on the CMB.

Also, assume your goals for these two emulators/analyzers are:

1. Have emulator/analyzer #1 trigger emulator/analyzer #2 with the CMB Trigger signal.

2. Have emulator/analyzer #2 break to the monitor when it receives the CMB Trigger from #1.

Note

If you are performing the examples in this chapter, follow each command in this chapter by pressing **Enter** (on your PC) or **Return** (on your terminal) unless instructed otherwise.

Here's how you would set up the entire process. Starting at the prompt:

```
1.  TYPE  xp  -s  31              (sets up secondary escape character for #1)
2.  TYPE  xp  -e                  (starts communication with #2)
3.  TYPE  xp  -s  32              (sets up secondary escape character for #2)
4.  TYPE  cmbt  -r  trig1         (sets #2 to receive trig1)
5.  TYPE  bc  -e  trig1           (#2 waiting to break into monitor)
6.  PRESS  ESC  and type  1       (escapes from #2 to #1)
7.  TYPE  xp  -d                  (starts communication with #1)
8.  TYPE  cmbt  -d  trig1         (drives trig1 to #2)
9.  PRESS  xp -e                  (escapes from #1 to #2)
10.  TYPE  es                     (shows #2 began executing in monitor)
```

Here's how this setup appears on each HP 64700:

Commands You Executed On Each Emulator/Analyzer:

1. xp -s 31
2. xp -e

7. xp -d
8. cmbt -d trig1
9. xp -e

3. xp -s 32
4. cmbt -r trig1
5. bc -e trig1
6. ESC 1

10. es

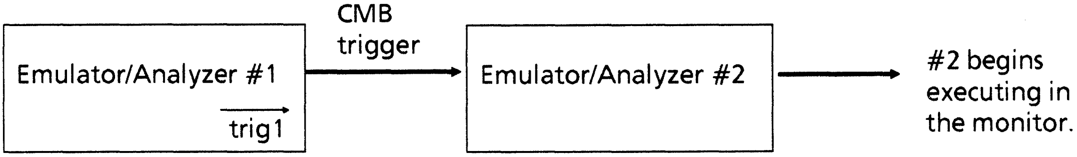


Figure 4-3. Measurements Between Two Emulators

Let's try another example. Imagine that you have three HP 64700 Series emulators/analyzers and an external analyzer configured on the CMB.

Also, assume your goals for three emulators/analyzers are:

1. Have the external analyzer trigger when emulator/analyzer #3 drives Trigger out the BNC connector.

2. Cause emulator/analyzer #3 to trigger when it receives a trig1 signal from emulator/analyzer #2.

3. Instruct emulator/analyzer #2 to set up a trace specification, trigger on a certain condition, start a measurement, and drive trig1 to #3.

4. Have emulator/analyzer #1 drive Trigger and Execute onto the CMB when it receives either a trig1 or trig2 signal.

Here's how you would set up the entire process. At the prompt:

```
1.  TYPE  xp  -s  31               (sets up secondary escape character for #1)
2.  TYPE  xp  -e                   (starts communication with #2)
3.  TYPE  xp  -s  32               (sets up secondary escape character for #2)
4.  TYPE  xp  -e                   (starts communication with #3)
5.  TYPE  xp  -s  33               (sets up secondary escape character for #3)
6.  TYPE  bnct  -r  trig1          (when #3 receives trig1, it drives
                                          BNC Trigger to analyzer)
7.  PRESS  ESC  and type  2        (escape to #2)
8.  TYPE  xp  -d                   (starts communication with #2)
9.  TYPE  tcf  -e                  (sets analyzer configuration)
10.  TYPE  rx 0                    (run emulator from address 0 when
                                          CMB Execute pulse is received)
11.  TYPE  tg  addr=03             (analyzer will trigger on address 03)
12.  TYPE  tx  -e                  (starts measurement when #2 receives Execute)
13.  TYPE  tgout  trig1            (#2 drives trig1 when analyzer triggers)
14.  TYPE  xp -e                   (puts #2 in transparent mode)
15.  PRESS  ESC  and type  1       (escape to #1)
16.  TYPE  xp  -d                  (starts communication with #1)
17.  TYPE  cmbt  -r  trig1,trig2   (#1 drives CMB Trigger when it
                                          receives trig1 or trig2)
18.  TYPE  x                       (starts synchronous CMB execution of all
                                          emulators on CMB)
```

Here's how this setup appears on each HP 64700:

Commands You Executed On Each Emulator/Analyzer:

```
1. xp -s 31
2. xp -e

16. xp -d
17. cmbt -r trig1, trig2
18. x
```

```
3. xp -s 32
4. xp -e

8. xp -d
9. tcf -e
10. rx 0
11. tg addr = 03
12. tx -e
13. tgout trig1
14. xp -e
15. ESC 1
```

```
5. xp -s 33
6. bnct -r trig1
7. ESC 2
```

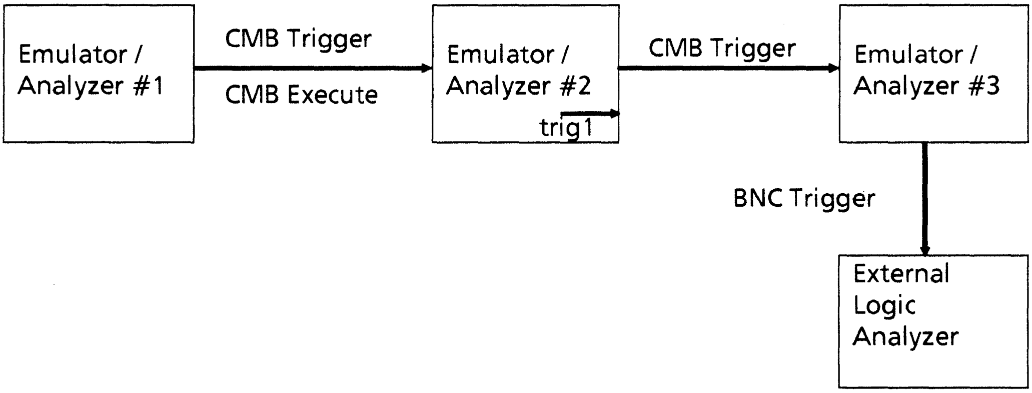


Figure 4-4. Measurements Between Several Emulators

Where To Find More Information

To find out more about transparent configurations, refer to the *HP 64700 Emulators Hardware Installation And Configuration* manual.

To find out more about the **xp** command, refer to the *HP 64700 Emulators Terminal Interface User's Reference* manual.

Using The IMB And CMB

Topics In This Chapter

- Components You Need

- Before Making Measurements
- How The IMB And CMB Operate
- How To Make IMB/CMB Measurements

Figure 5-1 shows the connections between the HP 64120A Instrumentation Cardcage in the HP 64000-UX Microprocessor Development Environment and an HP 64700 Series Emulator/ Analyzer for making IMB/CMB measurements.

Figure 5-2 is a diagram of the HP 64306A IMB/CMB Interface Control and Connector Boards.

Figure 5-3 shows the HP 64306A installed in the HP 64120A Instrumentation Cardcage. Part numbers of the cables used are also shown.

Figure 5-4 shows the HP 64306A as part of a configuration consisting of an HP 64000-UX emulator and analyzer.

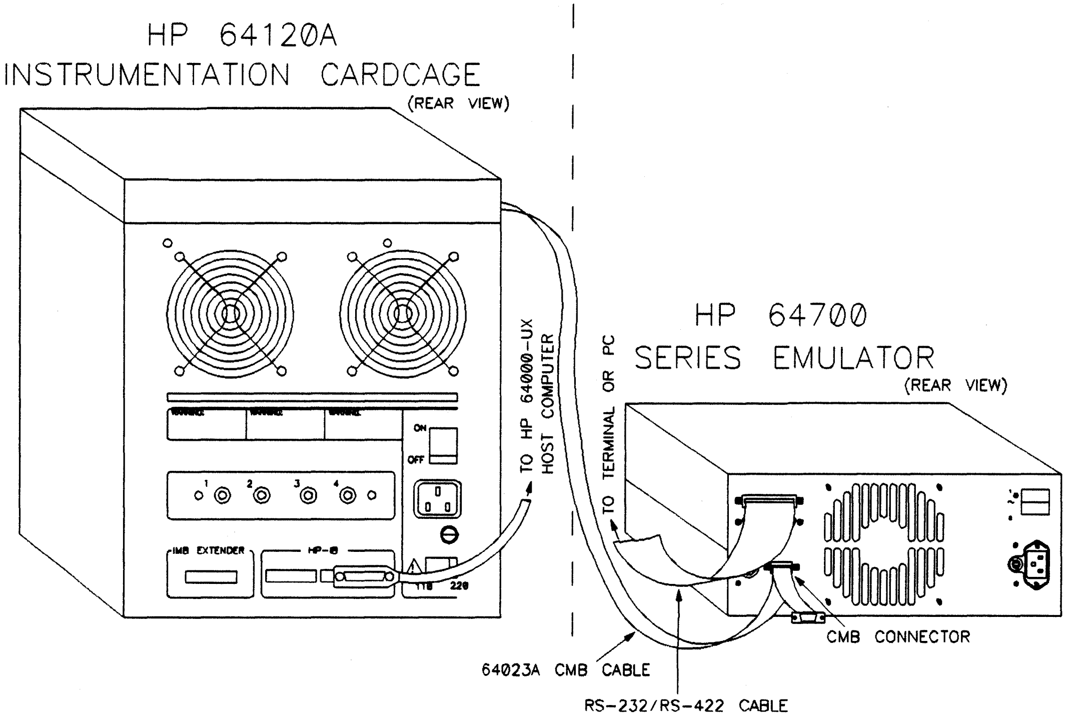


Figure 5-1. IMB/CMB Connections

Components You Need

The hardware, cable, and software components needed for making IMB/CMB measurements are listed in table 5-1.

Table 5-1. IMB/CMB Components You Need

Hardware	See figure
HP 64120A Instrumentation Cardcage	5-1
HP 64306A IMB/CMB Interface Control and Connector Board	5-2
HP 64000-UX Emulator and Analyzer	5-4
HP 64700 Series Emulator/Analyzer	5-1

Cables	Connection	See figure
HP-IB	HP 64120A to host computer	5-1
RS-232	HP 64700 to terminal or PC	5-1
CMB	HP 64700 to HP 64306A	5-1,5-3
assembly	HP 64306A Control to Connector Board	5-2
IMB	HP 64120A Internal IMB connector to HP 64000-UX Analyzer(s) and HP 64306A	5-3,5-4

Software
HP 64801 Operating System for HP 64000-UX (Revision 2.00 or greater)
HP 64000-UX Emulation and Analysis (for the components involved)

About The HP 64306A

A diagram of the HP 64306A IMB/CMB Interface is shown in figure 5-2. For details about installing and testing the IMB/CMB Interface, refer to the Installation Notice supplied with the product. Then return here.

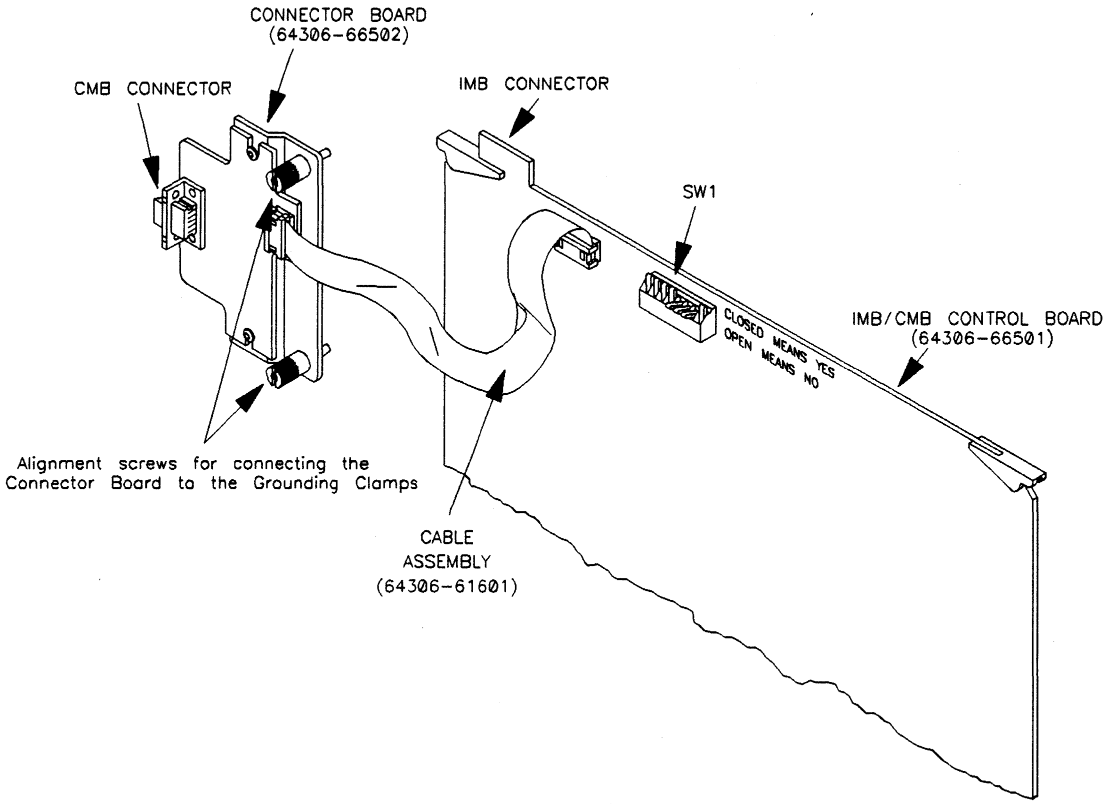


Figure 5-2. HP 64306A IMB/CMB Interface Boards

Other Sources Of Information

The *HP 64700 Series Hardware Installation And Configuration Manual* contains details about installing HP 64700 Series emulators, and connecting multiple HP 64700s to the CMB.

The *HP 64000-UX Installation And Configuration Manual* contains details about installing the HP 64120A Instrumentation Cardcage and HP 64000-UX emulators and analyzers, and about IMB cable setup and part numbers.

The *Installation Notice* supplied with the HP 64306A IMB/CMB Interface contains details about installing the HP 64306A in the HP 64120A Instrumentation Cardcage, and running Performance Verification on the board.

Before Making Measurements

Before making IMB/CMB measurements, make sure that you have installed all of the required hardware as described in the Installation Notice supplied with the HP 64306A IMB/CMB Interface.

Note

Make sure that your HP 64000-UX system and the HP 64700(s) involved are operating properly before continuing. Do this by running Performance Verification tests on the equipment.

For a multiple cardcage configuration, you must connect the IMB cable to the IMB Extender connector as shown in figure 5-3.

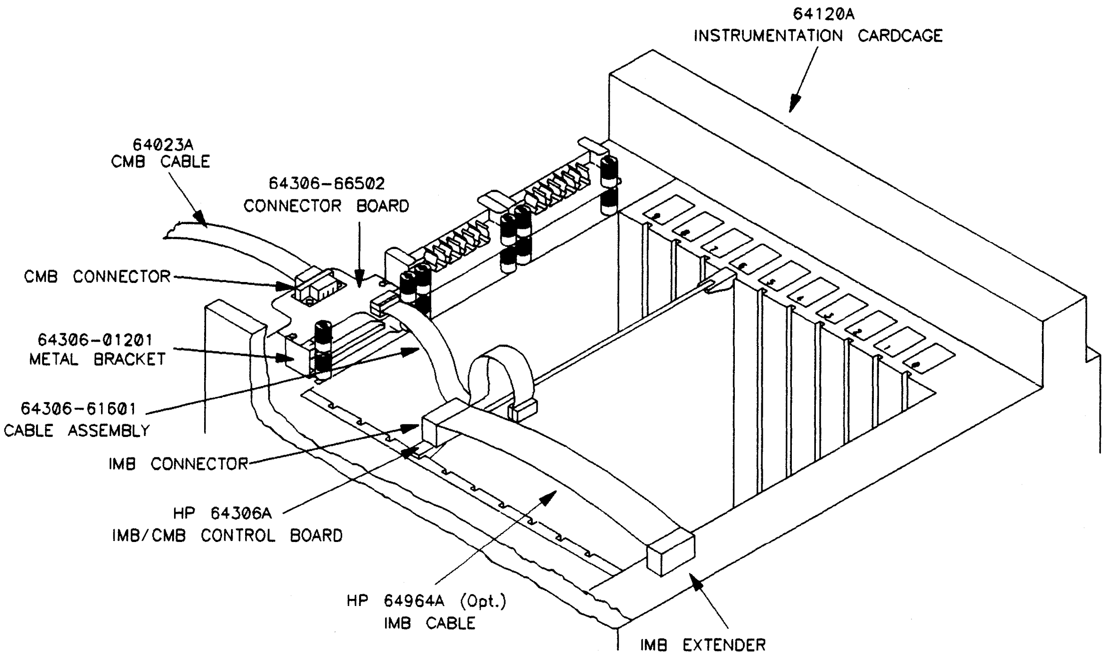


Figure 5-3. HP 64306A Installed In HP 64120A

Figure 5-4 shows the connection for a single cardcage that contains the HP 64306A, an emulator with memory, and an analyzer. In this case the IMB cable connects the analysis board with the IMB/CMB Interface Control Board to allow analyzer cross triggering. The IMB cable is also connected to the IMB extender, implying that a second HP 64120A Instrumentation Cardcage is involved in the measurement. If a second cardcage is not needed, the connection to the IMB Extender is not necessary.

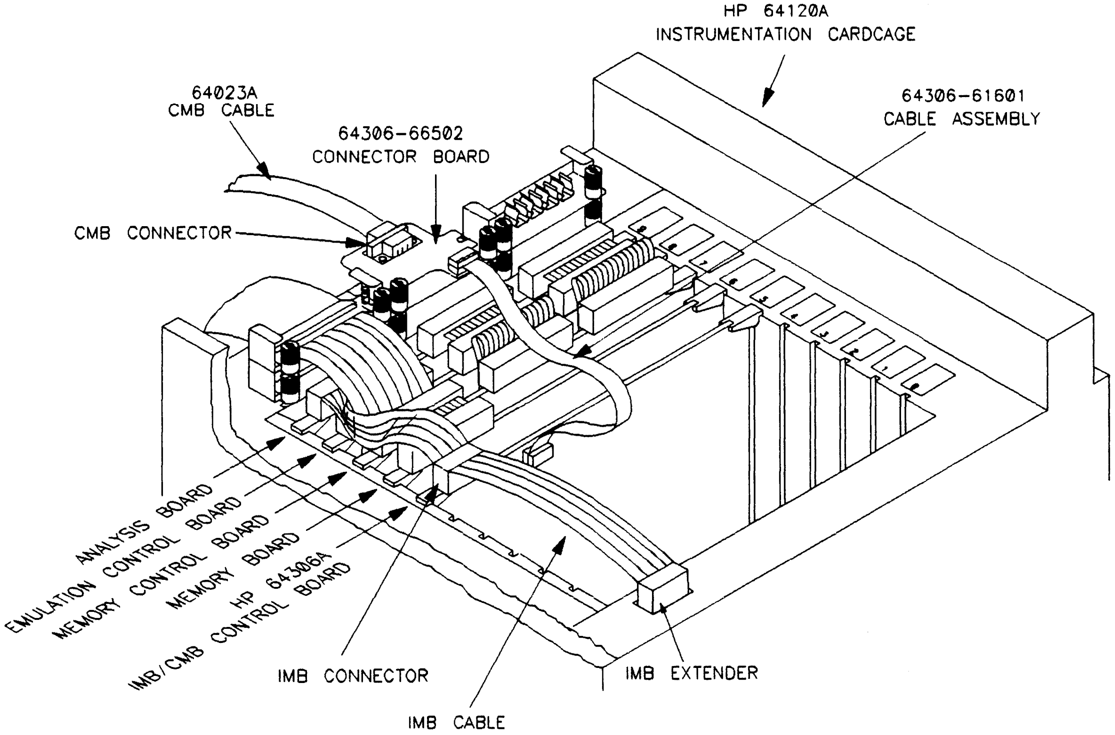


Figure 5-4. HP 64306A In A Complete System

If Problems Occur

Verify that all of the boards in the HP 64120A Instrumentation Cardcage are seated properly.

Check all cable connections to verify that they are valid, and that the cables are making a valid connection.

If the PV tests do NOT pass on the HP 64306A, or if the two status lights are not visible through the front panel of each HP 64700 Series emulator/analyzer on the CMB, call your Hewlett-Packard Representative and report the problem. Your *Support Services* manual tells you how.

How The IMB And CMB Operate

The HP 64306A IMB/CMB Interface Control and Connector Boards allow the Coordinated Measurement Bus (CMB) and InterModule Bus (IMB) to operate together. With the HP 64306A you can cross-trigger HP 64000-UX and HP 64700 Series analyzers, and can start HP 64000-UX and HP 64700 Series emulators in a coordinated manner.

Hardware Interaction

Chapter 1 of this manual explains the signals involved in the Coordinated Measurement Bus. The HP 64000-UX Measurement System Operating Manual explains the InterModule Bus signals.

This section explains how the HP 64306A IMB/CMB Interface uses the IMB and CMB signals to make measurements. The HP 64306A IMB/CMB Interface interacts with the IMB and CMB only when it is included in a measurement system with other HP 64000-UX modules that are involved in IMB measurements. This is explained later in this chapter.

Trigger Interaction

Whenever an HP 64306A IMB/CMB Interface is involved in a trigger measurement, it is both a driver and receiver of IMB and CMB trigger, but never the source of trigger. An analyzer on the IMB or CMB must be the source of trigger.

As a user of the HP 64000-UX modules and HP 64700 instruments, you must decide which analyzers will be the source of the trigger and which will receive trigger. This is explained in the "Request IMB Resource Usage" and "Set Up The HP 64700" sections of this chapter.

When the IMB sources the trigger, the HP 64306A IMB/CMB Interface immediately drives trigger to the CMB. The propagation time from IMB trigger received to CMB trigger driven (measured from connector to connector) is typically 28-33 nS. The transmission times on the IMB and CMB cables depend on the exact hardware configuration.

When the CMB sources the trigger, the IMB Master Enable (ME) signal must be true before trigger can be driven on the IMB. Some modules such as the HP 64610 Timing and State analyzers provide a very useful windowing function that allows triggers only during a specified window. Such modules may have ME false when the CMB trigger arrives and thus the HP 64306A IMB/CMB Interface holds trigger until ME goes true. If ME is true when the CMB trigger is received, the HP 64306A IMB/CMB Interface immediately drives trigger on the IMB. The propagation time from CMB trigger received to an immediate IMB trigger driven (measured from connector to connector) is typically 29-34 nS. The transmission times on the IMB and CMB cables depend on the exact hardware configuration.

Coordinated Starts

Both the IMB and CMB have coordinated start capabilities. The HP 64306A IMB/CMB Interface allows the IMB and CMB to work together for combined coordinated emulator starts.

The CMB READY line is used to coordinate the start of the HP 64700 instruments. The IMB LES (Low Emulation Start) line is used to coordinate the start of the HP 64000-UX modules. When any HP 64000-UX module requests a coordinated start, the HP 64306A IMB/CMB Interface is programmed to prevent the CMB READY line from going true.

The HP 64306A IMB/CMB Interface does not take CMB READY false; it only keeps it false if it is already false or ever becomes false. Thus any HP 64700 instrument that is not running user code (has READY false) will not be allowed to run user code until the HP 64306A IMB/CMB Interface releases READY true. The HP 64306A IMB/CMB Interface releases CMB READY to go true when it is called to start or when the IMB LES line goes true. At the same moment, the other HP 64000-UX modules that receive LES also start. The time between two emulators starting depends on how much each emulator must do before returning to execute user code. Measurements of several emulators show a typical start skew of 5 to 30 microseconds.

Components Allowed In The Measurements

A maximum of 32 HP 64700 Series emulators/analyzers can participate on the CMB side of an IMB/CMB measurement. However, if you plan to include more than 16 HP 64700s in a measurement, you must have your Hewlett-Packard Representative make the required changes. The HP 64306A IMB/CMB Interface is equivalent to one HP 64700 emulator/analyzer.

A maximum of 6 HP 64000-UX modules (emulators and analyzers combined) can participate on the IMB side of an IMB/CMB measurement. Only one module in the HP 64000-UX measurement system can be an HP 64306A. Refer to the *HP 64000-UX Measurement System Manual* for details about modules.

The limit on the number of HP 64306A IMB/CMB Interface Boards that you can install in a single HP 64120A cardcage is dictated only by the number of available slots in the cardcage. You can place as many HP 64306A IMB/CMB Interface Board in the cardcage as you want or need.

Note

To be sure that accurate IMB/CMB measurements are made:

1. All of the HP 64700 Series emulators/analyzers connected to the CMB must be powered up.

2. When using multiple HP 64120A cardcages, all cardcages connected to the IMB must be powered up.

How To Make Measurements

The remainder of this chapter shows you how to make measurements between modules on the InterModule Bus (IMB) and instruments on the Coordinated Measurement Bus (CMB).

Note

For additional details about measurement systems, modules, and making IMB measurements, refer to the *HP 64000-UX Measurement System Operating Manual.*

Make The Logical Connections

After you have made the physical connections (installed the HP 64306A boards and connected the cables), you must make the logical connections (allow the HP 64000-UX system to recognize the IMB/CMB Interface, emulators, and analyzers involved).

The logical connection is made by including all of the modules that are going to participate in the measurement(s), into one measurement system. This is described in the following steps.

Note

These examples include an HP 64224A 80186 emulator, HP 64155B Memory Controller and Memory, HP 64302A Internal Analyzer, and an HP 64306A IMB/CMB Interface in the HP 64120A Cardcage (in the HP 64000-UX environment), and an HP 64753 Z80 Emulator connected to another terminal or PC (in the HP 64700 Series environment).

For other configurations some steps may be slightly different, but the general process is the same.

1. Log on to the HP 9000 host computer.

2. Initialize all HP 64120A Instrumentation Cardcages involved in the measurement.

   TYPE: **msinit** and press **Return**

Use **msinit -s** if you just added cardcage(s) to the system.

3. Configure the measurement system(s).

   TYPE: **msconfig** and press **Return**

4. Make one measurement system consisting of the 80186 emulator, the HP 64302A Internal Analyzer, and the IMB/CMB Interface. The analyzer will automatically be combined with the emulator into a module. The IMB/CMB Interface will be a separate module.

   PRESS: **make_sys**

   TYPE: **imbtocmb** and press **Return**

Note If you are using multiple modules or measurement systems, you may want to customize the names given to the measurement systems and modules involved.

5. Add the 80186 emulator/analyzer module to the measurement system.

Note Module numbers are assigned by the measurement system software according to the relative position of the modules in the HP 64120A cardcage. The module numbers assigned to all of the modules are displayed on screen in the measurement system configuration specification. For this example, the following commands assume that the 80186 emulator/analyzer is assigned module number 1, and the IMB/CMB Interface module is assigned module number 0. This is consistent with the configuration shown in figure 5-4.

PRESS: **add 1 name_it**

TYPE: **trig186** and press **Return**

6. Add the IMB/CMB Interface to the measurement system.

PRESS: **add 0 Return**

7. End the measurement system configuration.

PRESS: **end Return**

Note ☝

Because we did not specify a name when adding the IMB/CMB Interface module (module 0), the default name (imb__cmb) is chosen for that module. You could assign the IMB/CMB Interface module a different name when adding that module to the measurement system by using the "name__it" softkey.

The logical connection between the 80186 emulator/analyzer and the HP 64306A IMB/CMB Interface is now complete. With this logical connection, whenever the 80186 emulator/analyzer requests to use IMB resources, the IMB/CMB Interface will automatically be programmed to provide the resource connection to the CMB.

Access The Measurement System

Before you can make any measurements, you must access the measurement system that contains the module to be included in the IMB/CMB measurement.

Access the measurement system you just created.

TYPE: **imbtocmb trig186** and press **Return**

Request IMB Resource Usage

After the logical connections are made between an HP 64000-UX emulator or analyzer and the HP 64306A IMB/CMB Interface, whenever the emulator or analyzer requests the use of an IMB resource, the measurement system software automatically includes the IMB/CMB Interface in the measurement. To understand how your emulator or analyzer requests IMB activity, refer to the *Operating Manual* supplied with that product.

For sake of consistency, the commands and examples that follow refer to the same components used previously in this chapter, namely the HP 80186 emulator and the HP 64302A internal analyzer.

For the HP 64302A analyzer to request IMB resources, you must modify the interactive measurements specification in the emulation configuration.

Note Refer to your *HP 64000-UX Emulator Operating Manual* for details about modifying your emulator's configuration.

When modifying the 80186 emulation configuration, you have two alternatives:

> You can modify the interactive measurement specification of the emulation configuration and save that configuration to a file. You can then load that configuration file to set up the trigger functions without having to manually answer all of the configuration questions again.
>
> You can modify the emulation configuration each time you want to change the interactive measurement specification.

In the examples in this chapter, you will modify the 80186 emulator configuration to drive external trigger, receive an external trigger, and to not drive any trigger. You will then save all of these configurations to separate files. This is the most convenient method for changing the interactive measurement specification.

An Example Configuration File

The following shows an example configuration file that is set up to drive trigger. Once you save the 80186 emulation configuration to a file, that file will resemble:

```
# This is an emulation configuration file
Micro-processor clock source?   internal
Restrict to real-time runs?   no
Number of significant address bits?   20
Break processor on write to ROM?   yes
Enable emulation monitor entry from reset ?   yes
Enable software breakpoint and illegal opcode vectors jams ?   yes
Enable user NMI ?   no
Allow user NMI to interrupt emulation monitor ?   no
Enable fast hold acknowledge mode ?   no
Enable output buffers (8087 present) ?   no
Interlock emulation memory READY with user READY ?   yes
BEGIN MEMORY MAP
default guarded
01000H thru 015FFH emulation rom
02000H thru 025FFH emulation ram
0F000H thru 0F0FFH emulation rom
0FFF00H thru 0FFFFFH emulation rom
END MEMORY MAP
PORT 1?   off
PORT 2?   off
Active edge?   rising
Trigger enable?   off
External trigger?   drive
Internal trigger?   on
Delay clock?   off
Enable polling for simulated I/O?   no
Simio control address 1?   SIMIO_CA_ONE
Simio control address 2?   SIMIO_CA_TWO
Simio control address 3?   SIMIO_CA_THREE
Simio control address 4?   SIMIO_CA_FOUR
Simio control address 5?   SIMIO_CA_FIVE
Simio control address 6?   SIMIO_CA_SIX
File used for standard input?   /dev/simio/keyboard
File used for standard output?   /dev/simio/display
File used for standard error?   /dev/simio/display
```

The relevant question for IMB interactive measurements is:

External trigger?

While the "External trigger?" question controls the HP 64302A analyzer's trigger communication to other modules, the "Internal trigger?" question controls the analyzer's internal trigger. To achieve proper behavior of the HP 64302A, you must answer both questions correctly. Any combination of internal and external triggers is permissible, but some are more useful that others. For example, if internal and external trigger are both set to "off", the HP 64302A never triggers. An "or" condition can be achieved by causing the external trigger to be "received", and setting the internal trigger "on". In this case, whichever event occurs first triggers the HP 64302A analyzer.

Modify the interactive measurement specification of the emulation configuration to create this file. When you get to the "External trigger?" question...

PRESS: **drive Return**

The next question you will see is "Internal trigger?". Change the current response to turn on the internal trigger since this configuration expects an internal trigger driver.

PRESS: **on Return**

If you defined other parts of your configuration file differently, you will have different answers for some of the questions.

Continue through the questions until you are prompted for a configuration file name.

TYPE: **drivetrig** and press **Return**

You will now have a file on your system named drivetrig.EA.

The emulation configuration file will automatically be saved with an extension of .EA. For example, if you save the emulation configuration to a file named "drivetrig" you will have a file on your host computer named "drivetrig.EA".

You can use the configuration file as it is now defined to drive trigger to an HP 64700.

Now create a configuration file to allow the HP 64000-UX system to receive trigger from the HP 64700. To do this, modify the interactive measurement specification in the emulation configuration again. When the question "External trigger?" appears...

PRESS: **receive Return**

The next question you will see is "Internal trigger?". Change the current response to turn off the internal trigger since this configuration expects an external trigger driver.

PRESS: **off Return**

Continue through the questions until you are prompted for a configuration file name.

TYPE: **recvtrig** and press **Return**

You will now have a file on your system named recvtrig.EA.

Create one more configuration file that does not drive or receive trigger. To do this, modify the interactive measurement specification in the emulation configuration again. When the question "External trigger?" appears...

PRESS: **off Return**

The next question you will see is "Internal trigger?". Change the current response to turn on the internal trigger since this configuration expects an internal trigger driver.

PRESS: **on Return**

Continue through the questions until you are prompted for a configuration file name.

TYPE: **notrig** and press **Return**

You will now have a file on your system named notrig.EA.

Set Up The HP 64700(s)

The setup for the HP 64700 emulators/analyzers follows the procedures described in chapters 1 and 3 of this manual. If you are not familiar with the content of those chapters, or the HP 64700 commands used for the CMB, read these chapters now, then return here.

When setting up the HP 64700, you have two alternatives:

1. You can define macros that contain all of the commands needed to perform a specific function (drive trigger, receive trigger, or start your HP 64700 running). You can then use these macros to execute functions without having to manually enter all of the commands.

2. You can execute multiple commands each time you want to perform a drive or receive function.

About The Example Macros

The following example macros are used to allow the HP 64700 Series emulator/analyzer to receive trigger or drive trigger. In these examples, we are using an HP 64753 Z80 Emulator. You may decide to use some or all of the commands in these example macros for your HP 64700.

Receive Macro

The command to define the example "receive" macro is:

```
mac receive={tgout none;cmbt -d trig1 -r none;tarm =trig1;tg arm}
```

The "receive" macro:

- does not drive trigger when the analyzer triggers (**tgout none**) (necessary if the "drive" macro was previously executed)
- CMB trigger drives internal trigger signal trig1 (**cmbt -d trig1 -r none**)
- trig1 arms the analyzer (**tarm =trig1**)
- specifies the trigger condition for the analyzer (**tg arm**)

Drive Macro

The command to define the example "drive" macro is:

```
mac drive={tarm always;cmbt -d none -r trig1;tgout trig1}
```

The "drive" macro:

- removes the arm condition established by the receive macro (**tarm always**)
- CMB trigger receives the internal trigger signal trig1 (**cmbt -d none -r trig1**)
- drives trig1 when the analyzer triggers (**tgout trig1**)

Refer to the *HP 64700 Terminal Interface User's Reference* for more information about these commands.

User Interfaces

HP 64000-UX User Interface

On the HP 64000-UX system, you may use the HP 64808 User Interface Software to access your emulation module (with the command **pmon**). Or you can access your emulation module directly, by specifying the measurement system and module name. The following examples DO NOT use the HP 64808 User Interface Software.

HP 64306A IMB/CMB User Interface

The HP 64306A does not have a dedicated user interface, as do the other HP 64000-UX analyzers and emulators. Thus you cannot select the imb_cmb module when entering the measurement system or ending out of another module in the measurement system. You can verify this by observing that no softkey options are available for the imb_cmb module when entering or leaving a measurement system.

Measurement System User Interface

Whenever a measurement system (such as imbtocmb) includes an HP 64306A IMB/CMB Interface, the HP 64306A will be added to the resource list in the Measurement System User Interface as other modules request resources that the HP 64306A uses to control the HP 64700 instruments.

The measurement system screen in Example #1 shows an example measurement system user interface, where an HP 64000-UX analyzer (trig186) has requested to drive trigger. Along with the analyzer shown as a trigger driver, the imb__cmb module is also shown as a driver. It is assumed that an HP 64700 analyzer is the trigger receiver.

The same is true when an HP 64000-UX module requests to receive trigger. That module is shown as a trigger receiver along with the imb__cmb module. It is assumed that an HP 64700 analyzer is the trigger driver.

Process Steps

Three classes of measurements that the HP 64306A IMB/CMB Interface deals with are 1) cross triggering, 2) coordinated starts, and 3) cross triggering with coordinated starts.

For these types of measurements, an overview of the appropriate process steps are listed below.

Cross Triggering

For cross triggering, the steps to follow are:

1. Clear any HP 64700 trigger. The HP 64700 trigger must be cleared before the HP 64000-UX analyzers are started, or the existing trigger will be interpreted as the new trigger event by any HP 64000-UX analyzers that are configured to receive trigger. (Use the **th** command to clear the HP 64700 trigger.)
2. Start the HP 64700 analyzers that are trigger receivers.
3. Start the HP 64000-UX analyzers.
4. Start the HP 64700 analyzers that are trigger drivers.

Coordinated Starts

For coordinated starts, the steps to follow are:

1. Prepare the HP 64000-UX modules to start.

2. Start the HP 64700 instruments.

3. Start the HP 64000-UX modules.

Cross Triggering with Coordinated Starts

For cross triggering with coordinated starts, the steps to follow are:

1. Prepare the HP 64000-UX modules to start.

2. Prepare the HP 64000-UX analyzers to trace.

3. Start the HP 64700 emulators and analyzers.

4. Start the HP 64000-UX modules.

Example Measurements

If you have been following the preliminary steps outlined in the previous sections of this manual, you will have already:

- installed the necessary boards
- connected all of the appropriate cables
- defined a measurement system (imbtocmb)
- included the HP 64224 80186 emulator/analyzer in the "imbtocmb" measurement system, naming it trig186
- included the HP 64306A IMB/CMB Interface with the name imb__cmb in the "imbtocmb" measurement system
- created 3 configuration files called drivetrig, recvtrig, and notrig
- accessed the "imbtocmb" measurement system you created

Four example measurements are described in the following pages:

- Example #1 shows how an HP 64000-UX analyzer drives trigger to an HP 64700 Series emulator/analyzer.
- Example #2 shows how an HP 64700 emulator/analyzer drives trigger to an HP 64000-UX analyzer.

- Example #3 shows you how to start the HP 64000-UX emulator and HP 64700 Series emulator running at the same time.
- Example #4 shows how both emulators start to run simultaneously, then HP 64000-UX drives trigger to HP 64700.

Example #1

An HP 64000-UX analyzer drives trigger to an HP 64700 Series emulator/analyzer.

In this example the 80186 emulator will be executing code, and upon finding a trigger event, will drive trigger to the HP 64753 Z80 emulation analyzer to determine what code was executing on the Z80. Figure 5-5 shows a block diagram of the measurement.

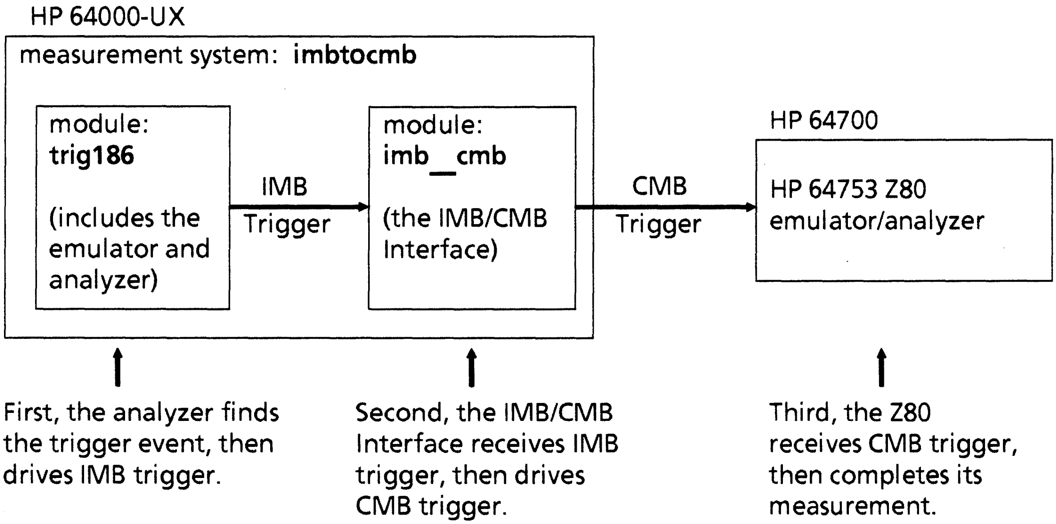

HP 64000-UX
measurement system: imbtocmb
module: trig186
(includes the emulator and analyzer)
IMB Trigger
module: imb_cmb
(the IMB/CMB Interface)
CMB Trigger
HP 64700
HP 64753 Z80 emulator/analyzer
First, the analyzer finds the trigger event, then drives IMB trigger.
Second, the IMB/CMB Interface receives IMB trigger, then drives CMB trigger.
Third, the Z80 receives CMB trigger, then completes its measurement.

Figure 5-5. HP 64000-UX Drives Trigger To HP 64700

1. Load the HP 64000-UX file configuration to drive trigger to the HP 64700.

   PRESS: **load config**

   TYPE: **drivetrig** and press **Return**

   You don't need to specify the extension of .EA when loading a configuration file. It is optional.

2. If you choose, you can now observe the measurement system specification.

   PRESS: **end select meas_sys Return**

   The measurement system screen will resemble:

```
HP 64000 Measurement System: imbtocmb
Module   Address    Status                     Description
trig186  07.18.2 initialized   I80186 emulation w/64302, 128K bytes emul mem
imb_cmb  07.18.9 initialized   IMB/CMB Interface Controller

                    Driver(s) Receiver(s)
master enable       exec/halt  trig186
trigger             imb_cmb
                    trig186
```

Notice that you are working in the imbtocmb measurement system, that both the trig186 and imb_cmb modules are initialized, as indicated by the Status, and that both modules are also drivers of trigger. The addresses on your screen may be different.

3. If you entered the measurement system display in step 2, reenter the trig186 module.

   PRESS: **trig186 Return**

Note

You must assemble and link the example 80186 program with the 80186 monitor, as described in the *HP 64000-UX 80186 Emulator Operating Manual,* before continuing.

4. Load an example program.

   For this example, we are using an example program named "demo186" that has already been assembled and linked with the 80186 monitor. The program resembles:

```
'80186'
            ASSUME CS:ORG
            GLB START, LOOP_1
            ORG 0FFFF0000H
            JMP FAR PTR START
             ASSUME CS:PROG
            PROG
START
            MOV AL,0EH
LOOP_1      DEC AL
            JNZ LOOP_1
            JMP START
            END START
```

   PRESS: **load**

   TYPE: **demo186** and press **Return**

5. Take the 80186 emulator out of the reset condition.

   PRESS: **run Return**

Your HP 64000-UX emulator/analyzer is now ready to participate in the measurements. Now you must prepare the HP 64700 Series emulator/analyzer to participate in the measurements. To do this, you must decide which commands you will execute on the HP 64700. You can define macros that include these commands. In this example, we will use the "receive" macro described earlier. You may want to use some different commands.

Refer to the *HP 64700 Emulators Terminal Interface User's Reference* for details about any of the commands executed on the HP 64700.

6. While your HP 64700 Series emulator/analyzer (HP 64753 Z80 in this example) is executing in the monitor (the prompt is "M>") define a macro to allow it to receive trigger from the HP 64000-UX analyzer. (To have the emulator begin executing in the monitor, type **b** and press **Return**.) To define the macro, on the terminal or PC connected to the Z80 emulator, type the following command:

```
mac receive={tgout none;cmbt -d trig1 -r none;tarm =trig1;tg arm}
```

7. Load a program into Z80 memory. For this example, we are using the following Z80 program:

```
START       LD A,4
OUTER       LD B,8
INNER       DEC B
            JRNZ    INNER
            INC A
            JP NZ   OUTER
            JP START
```

Because we have the Z80 program located at memory locations 0 through 0ch, those locations should resemble:

```
3e  04  06  08  05  20  fd  3c  20  f8  c3  00  00
```

Note If necessary, refer to your *HP 64700 Series Emulator User's Guide* for details about modifying memory. Then return here.

8. On the Z80 emulator/analyzer:

   TYPE: **receive** and press **Return**

The HP 64753 Z80 emulator/analyzer is prepared to participate in the measurement. Now let's make the actual measurement.

9. Start the Z80 emulator/analyzer.

   TYPE: **r** and press **Return**

   TYPE: **t** and press **Return**

The HP 64700 emulation trace has started, and the HP 64753 is waiting for HP 64000-UX analysis to complete its measurement. This will take place when we have issued a trace on the HP 64000-UX system, and the HP 64000-UX analyzer finds its trigger. The HP 64753 will wait to receive trigger for as long as it takes the HP 64000-UX system to complete the measurement.

10. You can program the Z80 analyzer to automatically display the trace data when the measurement completes.

    TYPE: **w -m;tl** and press **Return**

11. The trigger event for this example is the START symbol in the example 80186 program. Start a trace on the HP 64000-UX system.

    PRESS: **trace after**

    TYPE: **START** and press **Return**

12. Start the measurement from the HP 64000-UX system.

    PRESS: **run Return**

The result on the HP 64000-UX screen is:

```
Trace List     Break:none       Offset=0       Mode:execution
Label:  Address   Data                Opcode or Status                    time count
Base:     hex     hex                     mnemonic                         relative
after     0F000       B0    MOV AL,#0EH                                  ------------
+002      0F002     C8FE    DEC AL                                          0.64   uS
+003      0F004       75    JNZ F002H                                       0.64   uS
+005      0F002     C8FE    DEC AL                                          1.92   uS
+006      0F004       75    JNZ F002H                                       0.64   uS
+008      0F002     C8FE    DEC AL                                          1.28   uS
+009      0F004       75    JNZ F002H                                       0.64   uS
+011      0F002     C8FE    DEC AL                                          1.28   uS
+012      0F004       75    JNZ F002H                                       0.64   uS
+014      0F002     C8FE    DEC AL                                          1.92   uS
+015      0F004       75    JNZ F002H                                       0.64   uS
+017      0F002     C8FE    DEC AL                                          1.92   uS
+018      0F004       75    JNZ F002H                                       0.64   uS
+020      0F002     C8FE    DEC AL                                          1.28   uS
.
.
.
STATUS:   I80186--Running                Trace complete________________........
```

Once the HP 64000-UX analyzer has found its trigger event, the trigger will be driven to the HP 64753. Both systems will then display complete traces.

The HP 64753 trace resembles:

```
Line   addr,H  Z80 Mnemonic,H                         count,R  seq
 -----  ------  ------------------------------  ---------  ---
     0   0006        fd  operand                         ---    +
     1   0004     DEC B                            0.760 uS    .
     2   0052             refresh                  0.160 uS    .
     3   0005     JR NZ,0004                       0.240 uS    .
     4   0053             refresh                  0.160 uS    .
     5   0006        fd  operand                   0.280 uS    .
     6   0004     DEC B                            0.760 uS    .
     7   0054             refresh                  0.160 uS    .
     8   0005     JR NZ,0004                       0.240 uS    .
     9   0055             refresh                  0.160 uS    .
```

Because the Z80 analyzer was only cross triggering from an external trigger, and there was no further trigger qualification, data captured by the Z80 analyzer will likely be different on every retry of this measurement. In this example, the analyzer data is not significant. However, the method used to capture the data is important.

The HP 64000-UX and HP 64700 Series systems have now both completed their measurements.

To repeat the measurement, follow these steps:

13. Start another trace on the HP 64700.

    TYPE: **t** and press **Return**

14. Program the Z80 to automatically display the trace data when the measurement completes.

    TYPE: **w -m;tl** and press **Return**

15. Set up the HP 64000-UX trace.

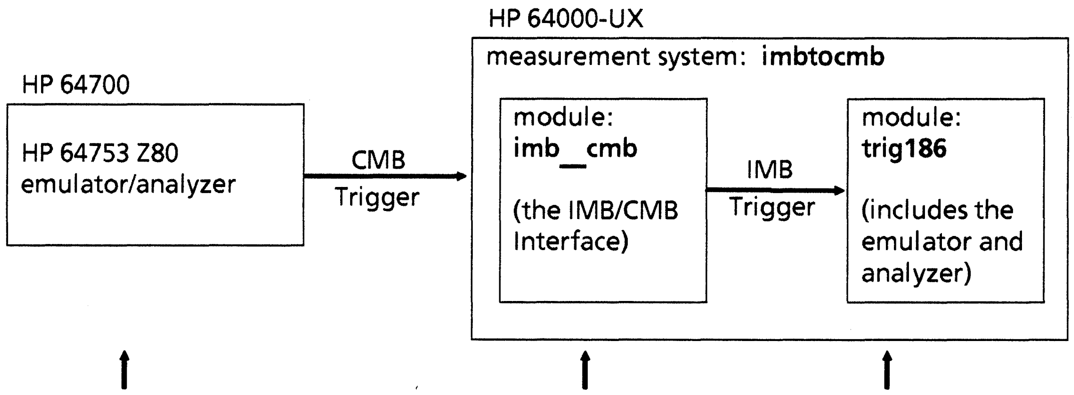
    PRESS: **trace again Return**

Both systems will complete and display their traces.

Example #2

An HP 64700 Series emulator/analyzer drives trigger to an HP 64000-UX analyzer.

In this example, the Z80 emulator will be executing code, and upon finding a trigger event, will drive trigger to the 80186 emulation analyzer to determine what code was executing on the 80186. Figure 5-6 shows a block diagram of the measurement.


HP 64700
HP 64753 Z80 emulator/analyzer
CMB Trigger
HP 64000-UX
measurement system: imbtocmb
module: imb_cmb
(the IMB/CMB Interface)
IMB Trigger
module: trig186
(includes the emulator and analyzer)

First, the HP 64700 emulation analyzer finds the trigger event, then drives CMB trigger.

Second, the IMB/CMB Interface receives CMB trigger, then drives IMB trigger.

Third, the HP 64000-UX analyzer receives IMB trigger, and completes its measurement.

Figure 5-6. HP 64000-UX Drives Trigger To HP 64700

In this example you will perform the same types of steps included in Example #1, except that you will:

- define and use the "drive" macro
- use the HP 64000-UX emulation configuration file to receive trigger

Before proceeding, make sure the Z80 emulator is running in the monitor (the prompt should be M>). If the prompt is not M>, type **b** and press **Return**.

1. On the HP 64700, define the macro to drive trigger to the HP 64000-UX system by typing:

```
mac drive={tarm always;cmbt -d none -r trig1;tgout trig1}
```

2. On the HP 64000-UX system load the configuration file you created earlier to allow HP 64000-UX to receive trigger from the HP 64700 (recvtrig).

   PRESS: **load config**

   TYPE: **recvtrig** and press **Return**

Note

If you did not perform Example 1, you must now load the example 80186 and Z80 programs (see Example 1, then return here).

---

3. If you choose, you can now observe the measurement system specification.

   PRESS: **end select meas_sys Return**

The HP 64000-UX screen will resemble:

```
HP 64000 Measurement System: imbtocmb
Module  Address    Status                       Description
trig186  07.18.2 complete    ! I80186 emulation w/64302, 128K bytes emul mem
imb_cmb  07.18.9 complete    ! IMB/CMB Interface Controller

                   Driver(s) Receiver(s)
master enable      exec/halt trig186
trigger                       imb_cmb
                              trig186
```

Notice that the imb__cmb and trig186 modules are both receivers of trigger. The status indicates that both modules have completed. If you did not perform Example #1, the status for the modules will indicate "initialized".

4. If you entered the measurement system specification in step 2, reenter the trig186 module.

PRESS: **trig186 Return**

5. Start the HP 64000-UX measurement system running. You could just simply start a trace, then run the emulator. However, if you are using the example 80186 program, run the emulator from START. For example:

   PRESS: **trace Return**

   PRESS: **run from**

   TYPE: **START** and press **Return**

The 80186 emulator is running, and the HP 64000-UX status line indicates "Trace in process". The imbtocmb measurement system is waiting for a trigger, and will not complete the measurement until it receives a trigger.

If the status line shows "Trace complete", the HP 64700 has previously driven trigger true. To clear the trigger on the HP 64700, type **th** and press **Return**, then repeat the 80186 trace command.

6. Prepare the HP 64700 to drive trigger using the "drive" macro.

   TYPE: **drive** and press **Return**

7. Set up the trigger event on the Z80 analyzer. For this example, use the "OUTER" loop of the Z80 program defined in Example 1. "OUTER" is located at address 02. The status of an opcode execution is 80h.

   TYPE: **tg addr=2 and stat=80** and press **Return**

8. Start the HP 64700 emulator and analyzer.

   TYPE: **t** and press **Return**

   TYPE: **r** and press **Return**

   To observe the HP 64700 trace...

TYPE: **tl** and press **Return**

Notice that the HP 64700 emulation analyzer displays its trace, along with the HP 64000-UX system. The HP 64700 trace resembles:

```
Line   addr,H  Z80 Mnemonic,H                        count,R  seq
   -----  ------  ------------------------------  ---------  ---
     -1    0001        04  operand                      ---   .
      0    0002   LD B,08                          0.240 uS   +
      1    001a            refresh                 0.160 uS   .
      2    0003        08  operand                 0.280 uS   .
      3    0004   DEC B                            0.240 uS   .
      4    001b            refresh                 0.160 uS   .
      5    0005   JR NZ,0004                       0.240 uS   .
      6    001c            refresh                 0.160 uS   .
      7    0006        fd  operand                 0.320 uS   .
      8    0004   DEC B                            0.720 uS   .
```

The HP 64000-UX trace resembles:

```
Trace List     Break:none       Offset=0        Mode:execution  Count:overflow
Label:  Address    Data               Opcode or Status                time count
Base:     hex     hex                     mnemonic                      relative
+001      0F002      C8FE    DEC AL                                    ------------
+002      0F004        75    JNZ F002H                                     0.64   uS
+004      0F002      C8FE    DEC AL                                        1.28   uS
+005      0F004        75    JNZ F002H                                     0.64   uS
+007      0F002      C8FE    DEC AL                                        1.92   uS
+008      0F004        75    JNZ F002H                                     0.64   uS
+010      0F002      C8FE    DEC AL                                        1.28   uS
+011      0F004        75    JNZ F002H                                     0.64   uS
+013      0F002      C8FE    DEC AL                                        1.28   uS
+014      0F004        75    JNZ F002H                                     0.64   uS
+016      0F006        EB    JMP F000H                                     0.64   uS
+018      0F000        B0    MOV AL,#0EH                                   1.92   uS
+020      0F002      C8FE    DEC AL                                        0.64   uS
.
.
.

STATUS:    I80186--Running                  Trace complete________________........
```

Notice that the status line indicates "Trace complete".

Because the 80186 analyzer was only cross triggering, data captured by the 80186 analyzer will likely be different for every measurement. In this case, the analyzer data is not significant. However, the method used to capture the data is important.

9. If you choose, you can now observe the measurement system specification.

   PRESS: **end select meas_sys Return**

The HP 64000-UX screen resembles:

```
HP 64000 Measurement System: imbtocmb

 Module  Address    Status                       Description
trig186  07.18.2 complete    ! I80186 emulation w/64302, 128K bytes emul mem
imb_cmb  07.18.9 complete    ! IMB/CMB Interface Controller

                 Driver(s) Receiver(s)
master enable    exec/halt  trig186
trigger                     imb_cmb
                            trig186
```

Notice that both modules have completed their measurement, as indicated in the Status column.

If you entered the measurement system specification in step 9, reenter the trig186 module.

   PRESS: **trig186 Return**

To repeat the measurement, only a few of the process steps need to be followed. They are:

10. Halt the HP 64700 trace.

    TYPE: **th** and press **Return**

11. Start the HP 64000-UX trace again.

    PRESS: **trace Return**

12. On the HP 64700, complete the measurement.

    TYPE: **t** and press **Return**

Both systems complete their traces.

Example #3

Start the HP 64000-UX emulator and HP 64700 Series emulator running at the same time.

In this example you will start the HP 64000-UX 80186 emulator and the HP 64700 Z80 emulator at the same time.

To start an HP 64000-UX emulator and an HP 64700 Series emulator running at the same time, you will:

- Prepare HP 64000-UX to run.
- Prepare the HP 64700 Series emulator to run.
- Execute the coordinated start on HP 64000-UX.

Follow the steps below to start both emulators running.

1. If you entered the measurement system specification in the last step in Example #2, reenter the trig186 module.

   PRESS: **trig186 Return**

2. Remove all cross trigger information from the analyzer.

   PRESS: **load config**

   TYPE: **notrig** and press **Return**

Note

If you did not perform Example 1 or 2, you must now load the example 80186 and Z80 programs (see Example 1, then return here).

---

3. On HP 64000-UX, set up the emulator to run.

   PRESS: **specify run from transfer_address Return**

4. If you choose, you can now observe the measurement system specification.

   PRESS: **end select meas_sys Return**

The measurement system screen resembles:

```
HP 64000 Measurement System: imbtocmb

 Module   Address    Status                      Description
trig186   07.18.2 complete    ! I80186 emulation w/64302, 128K bytes emul mem
imb_cmb   07.18.9 complete    ! IMB/CMB Interface Controller

                 Driver(s) Receiver(s)
emulation start             imb_cmb
                            trig186
```

Notice that the trig186 module is both a driver and receiver of emulation start, and that the imb_cmb module is a receiver of emulation start.

5. On the HP 64700, enable interaction with the CMB signals.

   TYPE: **cmb -e** and press **Return**

6. Set up the run address on the HP 64700.

   TYPE: **rx 0** and press **Return**

7. Start the HP 64753 Z80 emulator.

   TYPE: **x** and press **Return**

The following messages will appear:

```
!ASYNC_STAT  623! CMB execute break
!ASYNC_STAT  693! CMB execute; run started
```

The HP 64700 prompt changes to "W>", indicating that the Z80 emulator is waiting for CMB to become ready. To verify this, you can observe the status.

TYPE: **es** and press **Return**

You will see the message "Z80--Awaiting CMB ready". The HP 64700 will start only when it receives the READY signal on the CMB from the IMB/CMB Interface.

8. You may start the emulators from either the measurement system level or from the 80186 emulator (trig186) level. If you are at the measurement system level and want to start the measurement from the 80186 emulator level...

   PRESS: **trig186 Return**

The 80186 monitor is initialized, and the 80186 emulator is running in the monitor.

9. From either the measurement system level or the 80186 emulator level, start both emulators running.

   PRESS: **execute Return**

Both the HP 64700 and HP 64000-UX emulators start.

You can tell that the 80186 emulator has started running by observing the status line.

Now observe that the Z80 is running a user program.

TYPE: **es** and press **Return**

You will see the message "Z80--Running user program". The Z80 prompt changes to "U>", indicating that a user program is running.

To repeat the measurement, follow these steps:

10. Set up the 80186 run specification.

    PRESS: **specify run from transfer Return**

11. Send a CMB EXECUTE pulse from the Z80 emulator.

TYPE: **x** and press **Return**

12. Execute the measurement from the HP 64000-UX system.

    PRESS: **execute Return**

Example #4

Both emulators start to run simultaneously, then HP 64000-UX drives trigger to HP 64700.

1. On the HP 64000-UX system, load the configuration file to drive trigger.

   PRESS: **load config**

   TYPE: **drivetrig** and press **Return**

Note

If you did not perform Examples 1, 2, or 3, you must now load the example 80186 and Z80 programs (see Example 1, then return here).

---

2. Set up the 80186 run specification.

   PRESS: **specify run from**

   TYPE: **START** and press **Return**

3. Set up the 80186 trace specification.

   PRESS: **specify trace after**

   TYPE: **START** and press **Return**

The 80186 monitor is initialized, and the 80186 emulator is running in the monitor.

4. Set up the HP 64700 to receive trigger from the HP 64000-UX system by executing the receive macro you created in Example #1.

   TYPE: **receive** and press **Return**

5. Allow the HP 64700 to participate in the measurement.

   TYPE: **cmb -e** and press **Return**

6. Set up the HP 64700 to start the measurement upon receipt of CMB EXECUTE.

   TYPE: **tx -e** and press **Return**

7. Set up the HP 64700 to start running upon receipt of CMB EXECUTE.

   TYPE: **rx 0** and press **Return**

8. Start the HP 64700.

   TYPE: **x** and press **Return**

The following messages are displayed, then the HP 64700 prompt changes to "W>".

```
!ASYNC_STAT  623! CMB execute break
!ASYNC_STAT 1305! CMB execute; emulation trace started
!ASYNC_STAT  693! CMB execute; run started
```

9. On the HP 64000-UX system, execute the measurement.

   PRESS: **execute Return**

The 80186 emulator begins to run your user program, and the trace completes. The emulator continues to run.

The 80186 trace list resembles:

```
Trace List      Break:none        Offset=0       Mode:execution
Label:  Address    Data                 Opcode or Status                  time count
Base:     hex      hex                      mnemonic                        relative
after    0F000        B0    MOV AL,#0EH                                  ------------
+002     0F002      C8FE    DEC AL                                          0.64   uS
+003     0F004        75    JNZ F002H                                       0.64   uS
+005     0F002      C8FE    DEC AL                                          1.92   uS
+006     0F004        75    JNZ F002H                                       0.64   uS
+008     0F002      C8FE    DEC AL                                          1.28   uS
+009     0F004        75    JNZ F002H                                       0.64   uS
+011     0F002      C8FE    DEC AL                                          1.28   uS
+012     0F004        75    JNZ F002H                                       0.64   uS
+014     0F002      C8FE    DEC AL                                          1.92   uS
+015     0F004        75    JNZ F002H                                       0.64   uS
+017     0F002      C8FE    DEC AL                                          1.92   uS
+018     0F004        75    JNZ F002H                                       0.64   uS
+020     0F002      C8FE    DEC AL                                          1.28   uS
.
.
.
STATUS:    I80186--Running                 Trace complete________________........
```

10. Verify that the HP 64700 is running your program.

TYPE: **es** and press **Return**

11. Observe the HP 64700 trace.

TYPE: **tl** and press **Return**

The Z80 trace list resembles:

```
Line    addr,H  Z80 Mnemonic,H                          count,R   seq
   -----   ------  -----------------------------   ---------   ---
       0    0000    LD A,04                                 ---     +
       1    007f            refresh                   0.120 uS     .
       2    0001         04  operand                  0.320 uS     .
       3    0002    LD B,08                           0.240 uS     .
       4    0000            refresh                   0.160 uS     .
       5    0003         08  operand                  0.280 uS     .
       6    0004    DEC B                             0.280 uS     .
       7    0001            refresh                   0.120 uS     .
       8    0005    JR NZ,0004                        0.280 uS     .
       9    0002            refresh                   0.120 uS     .
```

To repeat the entire measurement, follow these steps:

12. Set up the 80186 run specification.

    PRESS: **specify run from**

    TYPE: **START** and press **Return**

13. Set up the 80186 trace specification.

    PRESS: **specify trace again Return**

14. Start the Z80 executing.

    TYPE: **x** and press **Return**

15. Start the measurement from the HP 64000-UX system.

    PRESS: **execute Return**

Both systems complete their measurements.

Notes

A

Status And Error Messages

This chapter contains a list of status and error messages that you may encounter while making CMB measurements, or while making measurements using the BNC.

Status Messages

BNC trigger break

A new positive edge was recognized at the BNC connector. The command **bc -e bnct** was executed, enabling the BNC Trigger as a valid break condition. This message will appear at the first request for status, following the event that occurred. This message can indicate that the condition caused the emulator to break into the monitor, or implies that a condition occurred following the break into the monitor.

CMB execute; emulation trace started

The emulation analyzer recognized that the CMB EXECUTE pulse was received, and started its trace.

CMB execute; external trace started

The emulation analyzer recognized that the CMB EXECUTE pulse was received, and started its external trace.

CMB execute; run started

The emulator recognized that the CMB EXECUTE pulse was received, and started running a user program.

CMB trigger break

The command **bc -e cmbt** was executed, enabling the CMB Trigger as a valid break condition. This message will appear at the first request for status, following the event that occurred. If this line is true when the break condition is enabled, the message will appear immediately, and if the emulator is not in the monitor, a break will occur.

This message can indicate that the condition caused the emulator to break into the monitor, or implies that a condition occurred following the break into the monitor. It differs from the BNC Trigger break, because during a break, monitor or run cycle, the message will be sent only once. If the CMB Trigger line is still active when a request to run occurs, and the status message has been sent while in the monitor, the emulator will immediately break and return to the monitor.

trig1 break

The trig1 signal caused the emulator to break into the monitor. The command **bc -e trig1** was executed, enabling trig1 as a valid break condition. This message will appear at the first request for status, following the event that occurred.

Before this line is enabled as a break condition, the system attempts to clear all previous true conditions. If trig1 is being driven true by its connection to a true CMB Trigger line, the message will appear immediately. If the emulator is not running in the monitor, a break will occur.

This message can indicate that the condition caused the emulator to break into the monitor, or implies that a condition occurred following the break into the monitor. If the cause for trig1 going true is a CMB Trigger that does not go away, it will behave like the CMB Trigger break, and only send its message once during a break cycle. If the CMB Trigger goes away, or trig1 is being driven from an internal source, the system will continue to send this message each time it is true when status is requested.

trig2 break

See **trig1 break** message above.

Error Messages

Run failed during CMB execute

The emulator was not able to run user code when CMB EXECUTE was driven. Reasons for this error may include:

a. an invalid stack exists

b. attempting to run in non-existent memory

c. no clock is present

To recover from this error, verify the program is loaded properly. Then verify that a clock input is present. Next, verify that the emulator memory map is set up properly.

Trace error during CMB execute

The emulation or external analyzer encountered an error while trying to load the trace specification upon receiving the CMB EXECUTE signal. One reason for this error may be that an invalid trigger setup exists.

To recover from this error, correct the trace setup. Refer to your *HP 64700 Emulators Terminal Interface Analyzer User's Guide*, or *HP 64700 Emulators Terminal Interface User's Reference*, if necessary, to verify correct usage of the analyzer commands.

Trig1 signal cannot be driven and received

An attempt was made to have either the emulation analyzer or external analyzer drive trig1 with **tgout** (or **xtgout**) and receive trig1 with **tarm** (or **xtarm**). Although the configuration appears valid, the current analyzers cannot both drive and receive the same resource.

To recover from this error, make sure that you have issued the proper commands for each emulator/analyzer involved in the CMB measurement. The commands for the emulation analyzer could include: **cmbt -r trig1**, **cmbt -d trig1**, **bnct -r trig1**, **bnct -d trig1**, **tgout trig1**, and **tarm = trig1**.

Trig2 signal cannot be driven and received

See message above for trig1.

Index

Notes

HEWLETT
PACKARD

Hewlett-Packard
Printed in the USA